Coordination of drones at scale: Decentralized energy-aware swarm intelligence for spatio-temporal sensing

Chuhao Qin, Evangelos Pournaras *

School of Computing, University of Leeds, Leeds, UK

ARTICLE INFO

ABSTRACT

Smart City applications, such as traffic monitoring and disaster response, often use swarms of intelligent and cooperative drones to efficiently collect sensor data over different areas of interest and time spans. However, when the required sensing becomes spatio-temporally large and varying, a collective arrangement of sensing tasks to a large number of battery-restricted and distributed drones is challenging. To address this problem, this paper introduces a scalable and energy-aware model for planning and coordination of spatio-temporal sensing. The coordination model is built upon a decentralized multi-agent collective learning algorithm (EPOS) to ensure scalability, resilience, and flexibility that existing approaches lack of. Experimental results illustrate the outstanding performance of the proposed method compared to state-of-the-art methods. Analytical results contribute a deeper understanding of how coordinated mobility of drones influences sensing performance. This novel coordination solution is applied to traffic monitoring using real-world data to demonstrate a 46.45% more accurate and 2.88% more efficient detection of vehicles as the number of drones become a scarce resource.

1. Introduction

Unmanned Aerial Vehicles (UAVs), commonly known as *drones*, can form swarms for self-organization and collaboration in airborne mobile ad hoc networks. Nowadays, swarms of drones emerge in Smart Cites for spatio-temporal sensing (Wu et al., 2016; Inoue, 2020; Butilă and Boboc, 2022). They are assigned tasks to execute such as collecting sensor data in areas of interest. For instance, swarms can capture images/videos of traffic-related information on public roadways; measure air temperature and humidity to support sustainable crop production; transmit real-time reports of natural disasters such as fire and car accidents; or accurately deliver goods in densely populated areas. In transportation system, drones can be used for an accurate monitoring of traffic to detect traffic congestion at early stage. This allows traffic operators to apply mitigation actions that decrease the carbon footprint of a sector with one of the highest carbon emissions worldwide (Butilă and Boboc, 2022; Barmpounakis and Geroliminis, 2020).

To assist swarms of drones to complete sensing tasks efficiently, autonomous control of swarms and assignment of sensing tasks become a niche. Coordinated sensing involves the assignment of different sensing tasks to each drone while meeting the sensing requirements, drone capabilities and constraints (Poudel and Moh, 2022). Earlier work is proposed to address the task assignment problem for efficient and large-scale spatio-temporal sensing by swarms of drones (Fu et al., 2019; Yanmaz et al., 2018; Zhou et al., 2020). Considering the heterogeneity and number of tasks, the problem is formulated as an NP-hard combinatorial optimization problem to find the optimal assignment of sensing tasks. Task assignment algorithms designed to solve spatio-temporal sensing problems range from market-based methods (Alighanbari and How, 2005) to swarm intelligence (Mac et al., 2018).

However, mainstream approaches for UAV sensing task assignments do not achieve scalability, resilience and flexibility. This is because they rely on the centralized decision making while sacrificing a significant autonomy from drones (Poudel and Moh, 2022; Bupe et al., 2015). Therefore, the goal of this paper is to tackle instead a decentralized task assignment problem; drones collectively arrange and self-assign their sensing tasks, which is a highly complex research endeavor. On the one hand, coordinating the sensing tasks of drones is complex, i.e., large areas of interest, with varying sensing requirements and time-constrained missions. Certain areas with traffic jams or accidents may require for drones more fine-grained sensor measurements than areas with more regular traffic patterns. On the other hand, the inherent limitations in battery capacity influence spatio-temporal coverage. To tackle this complex task self-assignment problem, a decentralized and energy-aware coordination of drones at scale is introduced. Autonomous drones share information and allocate tasks cooperatively to meet complex sensing requirements while respecting battery constraints. Furthermore, the decentralized coordination method prevents single points of failure, it is more resilient and preserves the autonomy of drones to choose how they navigate and sense (Nik and Moazami, 2021).

A novel coordination model is designed using a multi-agent collective learning algorithm for multi-objective combinatorial optimization. It is selected because of its remarkable scalability (support of a large number of software agents/drones), efficiency (low communication and computational cost) and resilience, while preserving the privacy and autonomy of agents (Pournaras et al., 2018; Pournaras, 2020; Pournaras et al., 2020). The self-assignment problem of sensing tasks by a swarm of coordinated UAVs is also validated within a novel prototyping testbed deployed in indoor sensing environments (Qin et al., 2022; Fanitabasi et al., 2020). Furthermore, a novel plan generation strategy with three policies is designed to produce effective navigation and sensing alternatives for flexible drones. A power consumption model (Stolaroff et al., 2018) is used in the proposed strategy to estimate the energy consumption of these alternatives. This can make the coordination of drones energy-aware as each drone determines its sensing tasks based upon its energy consumption restrictions, while improving the sensing performance. Finally, extensive evaluations are conducted using analytical methods, simulated and real-world data including comparisons to the state-of-the-art baseline methods.

The contributions of this paper are outlined as follows:

1. A first study of the task self-assignment problem for spatio-temporal sensing by a swarm of interactive drones;
2. A decentralized coordination model by integrating multi-agent collective learning (Pournaras et al., 2018) to improve scalability, resilience, and flexibility of spatio-temporal sensing;
3. A plan generation strategy with three policies based on a power consumption model (Stolaroff et al., 2018) to make the coordination model energy-aware and achieve a highly efficient navigation and sensing of drones.
4. An open dataset (Qin and Pournaras, 2022) containing all the plans of the studied scenario. They can be used as benchmarks to encourage further research on this problem.
5. Analytical results to understand more rigorously how coordinated mobility of drones influences sensing performance.
6. A comprehensive empirical understanding of how a large spectrum of factors such as the number of dispatched drones, the spatial granularity of sensing, the number of base stations and the required amount of collected data influence the sensing performance.
7. A traffic monitoring model using coordinated drones validated as accurate and efficient using real-world data.

The rest of this paper is organized as follows: Section 2 positions this work in literature. Section 3 introduces the designed coordination model of spatio-temporal sensing. Section 4 illustrates our proposed plan generation method. Section 5 introduces the experimental settings, Section 6 illustrates the experimental evaluation and Section 7 concludes this paper. The appendices contain the power consumption model, the analytical results with proofs of theorems and further information on the experimental settings.

2. Related work

The assignment of sensing tasks within a swarm of intelligent and cooperative drones includes applications of traffic monitoring, disaster response, smart farming, last-mile delivery, etc. Poudel and Moh (2022). The UAV task assignment problem is earlier defined as a Traveling Salesmen Problem for combinatorial optimization (Gao et al., 2018; Wu et al., 2021; Chen et al., 2018). An optimal task assignment for drones is found at specified places, subject to constraints including task emergency, time scheduling and flying costs. The digraph-based methods are introduced to formulate the problem of reaching optimal sensing efficiency in terms of coverage, inspection delay, events detection rate and the cost of flying trajectories. Nevertheless, existing models do not address the energy impact of task assignment since small-scale spatio-temporal scenarios do not drain the batteries of drones significantly. Although earlier work focuses on solving the energy-aware task assignment problem for large-scale and efficient sensing, it relies on a centralized system and does not consider the time of sensor data collection (Motlagh et al., 2019; Zhou et al., 2018; Bartolini et al., 2020). This paper models the cost of plans calculated by the total power consumption of flying and hovering (Stolaroff et al., 2018), which is used to optimize the decentralized coordination of swarms, while accounting for the battery constraint of each drone.

In the view of the optimization approaches shown in Table 1, scholars introduce algorithms that require a centralized computation, such as particle swarm optimization (Gao et al., 2018), genetic programming (Wu et al., 2021) and wolf pack search (Chen et al., 2018). Chen et al. (2018) introduce a deadlock-free algorithm to prevent two or more drones from waiting for each other in a simultaneous task, and leverage the classical interior point method and wolf pack search algorithm to solve the uncertainty problem, including the uncertainty of the flying velocity, task effectiveness and communication. Bartolini et al. (2019) leverage a greedy approach by removing redundant target points from trajectories to maximize the weighted coverage while respecting energy constraints (with 5 drones to cover 225 target points). However, these centralized methods face the risk of single

Table 1

Comparison to related work: (✓) indicates criteria covered, (✗) indicates criteria not covered.

Criteria Ref.:	Gao et al. (2018)	Wu et al. (2021)	Chen et al. (2018)	Bartolini et al. (2019)	Alighanbari and How (2006)	Chen et al. (2022)	Elloumi et al. (2018)	This paper
Sensing-efficiency	✓	✓	✓	✓	✓	✓	✓	✓
Energy-awareness	✗	✗	✗	✓	✗	✗	✗	✓
Scalability	✗	✓	✓	✓	✓	✓	✗	✓
Resilience	✗	✗	✗	✗	✗	✗	✗	✓
Flexibility	✗	✗	✗	✗	✓	✓	✗	✓
Coordination	✗	✗	✗	✗	✓	✓	✗	✓

Table 2

Notations.

Notation	Explanation
$\mathcal{U}, \mathcal{N}$	The set of dispatched drones; the set of grid cells
u, U	The index of a dispatched drone; the total number of dispatches
n, N	The index of a grid cell; the total number of grid cells
T_n	The target value (or sensing requirements) at cell n
$S_{u,n}$	The sensing values collected by drone u at cell n
τ_u	The total flying time of drone u without hovering (seconds)
P_u^f, P_u^h	Flying and hovering power consumption of drone u (Watt)
E_u^f, E_u^h	Flying and hovering energy consumption of drone u (Joule)
C_u, e	The battery capacity of drone u (Joule); energy utilization ratio
J_u	The set of visited cells
f	The frequency with which drones collect sensing data (unit of sensing value per second)
p, P	The index of a plan; the total number of plans
B	The index of a base station for departure/return
K_B	The travel range or cells close to base station B
m, M	The index of a time unit in a plan; the total number of time units in a plan
$h_{m,n}$	The element of the plan to represent whether a drone occupies the cell n at time unit m
V_m, V_m^*	The total number of vehicles at time unit m; the number of vehicles observed by drones at time unit m

point of failure (Omidshafiei et al., 2017). If the central control station fails, the task assignment process cannot recover or mitigate such failure.

Distributed task assignment algorithms include the robust decentralized task assignment (RDTA) (Alighanbari and How, 2006) and the consensus-based bundle algorithm (CBBA) (Chen et al., 2022). RDTA improves robustness and reduces communication cost using the decentralized planning, wherein the drones plan independently and communicate a set of candidate plans. CBBA combines the distributed structure of market-based mechanisms and situation awareness with a consensus strategy that converges and can avoid conflicts between the tasks executed by drones. Both algorithms coordinate autonomous drones, and can ensure scalability and flexibility of multi-UAV systems to adapt to complex sensing scenarios. Nevertheless, they do not address the constraints of energy consumption. Moreover, the resilience of the algorithms is not verified, in contrast to earlier work on EPOS shown to preserve its learning capacity in dynamic and unstable networked environments (Pournaras et al., 2020). The proposed method in this paper overcomes scalability and resilience barriers, while leaving drones with a significant level of autonomy to make a flexible cooperative selection of tasks to execute. Furthermore, the prototyping problem of moving from simulation, to live deployment of decentralized socio-technical systems, and ultimately to a robust operation of a high technology readiness level (TRL) as well as online iterative traffic optimization is addressed in earlier work, with EPOS as a case study (Fanitabasi et al., 2020; Gerostathopoulos and Pournaras, 2019).

Although drones have shown a potential to advance traffic monitoring in accuracy, safety and cost savings (Outay et al., 2020), there is a very limited related work that focus on the task assignment in this scenario. Elloumi et al. (2018) propose a road traffic monitoring system that generates adaptive UAVs trajectories by extracting information about vehicles. Although this approach shows high performance in collecting traffic vehicle data, it does not address energy constraints and varying tasks over different time spans.

3. System model

This section introduces the coordination model to solve the decentralized task assignment problem for spatio-temporal sensing by a swarm of drones. Table 2 illustrates the list of mathematical notations used in this paper.

3.1. Scenario and architecture

Consider a swarm of drones that perform sensing over a grid — a 2D map over the spatial illustrative model shown in Fig. 1. A dispatch u is defined as a sensing task between the departure and return of a drone. In this scenario, a finite number of base/charging stations, from where drones depart and return, are set with fixed coordinates in the area. Drone are equally distributed over these
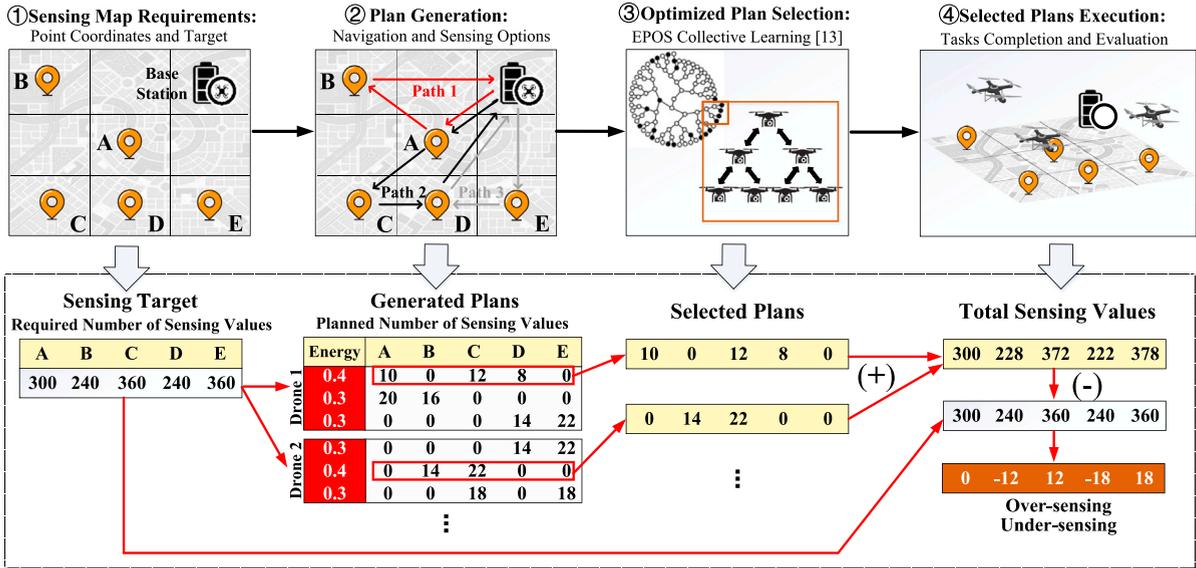
Fig. 1. Overview of the decentralized energy-aware coordination model for spatio-temporal sensing by a swarm of drones.

base stations, and have to return to the original base stations from which they depart. In addition, let $\mathcal{N}$ be the set of N points of interest, each is regarded as a grid cell that covers an area in the map. The main goal of drones is to coordinate their visits to cells to collect the required data.

Sensing map requirements. In the context of a sensing task, each cell at a time period has specific sensing requirements that determine the hovering duration and data acquisition of drones. Such sensing requirements can be determined by a continuous kernel density estimation, for instance, monitoring cycling risk based on requirements calculated by past bike accident data and other information (Castells-Graells et al., 2020; Qin et al., 2022). A high risk level in a cell represents higher sensing requirements over this area (e.g., the accidents or crucial intersection of traffic flow). Thus, a higher number of sensing values is set at this cell, which requires drones to hover a longer time over the cell to measure accurately. For example, see Fig. 1. It is assumed that the cell A requires the total sensing values of 300 at a time period. The sensing requirements of cells at a time period are set as the *targets* encoded by a vector of size N, e.g., $\{300, 240, 360, 240, 360\}$.

Plan generation. To coordinate the assignment of sensing tasks, the drones use the decentralized multi-agent collective learning method of EPOS, the *Economic Planning and Optimized Selections* (Pournaras et al., 2018; Pournaras, 2020). Given the sensing map, each drone, controlled by a local software agent, autonomously generates a finite number of discrete navigation and sensing alternatives, which provide flexibility for the drones to choose in a coordinated way. Each alternative has sensing details as well as an estimated energy consumption; the alternatives are referred to as the *possible plans* and each comes with a (normalized) *cost* respectively. For instance, a plan encoding that a drone travels and hovers over the cells A, C and D to collect respectively 12, 10 and 10 sensing values over a time period is encoded with: $\{12, 0, 10, 10, 0\}$ with $N = 5$. The energy consumed by a drone that carries out its planned tasks is calculated via a power consumption model (Stolaroff et al., 2018) with input the specification of the drone (weight, propeller, and battery parameters). This model can estimate the cost of navigation and sensing plans, and emulates the outdoor environments (Qin et al., 2022).

Plan selection and execution. To make coordinated plans selection, the agents of the drones connect into a tree communication structure within which each interacts with its children and parent in a bottom-up and top-down fashion to improve plan selections iteratively (Pournaras et al., 2018). The objective of this method is to select the optimal plan for each agent such that all choices together add up to maximize the sensing quality: the overall sensing data collected by all agents matches well the required data (target, see Fig. 1). In contrast, the mismatch is a result of *over-sensing* and *under-sensing*. For instance, in Fig. 1, the sensing value of 372 for which drones hover over the cell C exceeds the requirements of 360 (over-sensing), whereas drones hover over the cell E to collect 222 sensor values that is lower than the requirements of 240 (under-sensing). Over-sensing causes excessive data that needs further processing, waste of energy consumption, high storage and privacy cost, while under-sensing fails to satisfy sensing requirements. As a consequence, the generation and selection of high-quality plans are required to eliminate over-sensing and under-sensing.

3.2. Problem formulation

The goal of the proposed system is to find the optimal plan for each drone $u \in \mathcal{U}$, i.e., determining which cells a drone visits and how many sensor values it collects over a time period. Apart from preventing over-sensing and under-sensing, the formulated

problem aims to maximize the accomplishment of sensing tasks under the energy constraints for drones. Therefore, the problem is formulated as follows:

$$\min_{S_{u,n}, u \in \mathcal{U}, n \in \mathcal{N}} \sum_{n \in \mathcal{N}} (T_n - \sum_{u \in \mathcal{U}} S_{u,n})^2, \tag{1}$$

$$\min_{S_{u,n}, u \in \mathcal{U}, n \in \mathcal{N}} (1 - \frac{\sum_{u \in \mathcal{U}} \sum_{n \in \mathcal{N}} S_{u,n}}{\sum_{n \in \mathcal{N}} T_n}), \tag{2}$$

$$s.t. \quad \sum_{u \in \mathcal{U}} \sum_{n \in \mathcal{N}} S_{u,n} \leq \sum_{n \in \mathcal{N}} T_n, \tag{3}$$

$$P_u^f \cdot \tau_u + P_u^h \cdot \frac{\sum_{u \in \mathcal{U}} S_{u,n}}{f} \leq C_u. \tag{4}$$

where $S_{u,n}$ is the sensor values collected by the drone u at cell n, $u \in \mathcal{U}$, $n \in \mathcal{N}$; T_n is the target value at cell n; Objective (1) measures the over-sensing and under-sensing using a quadratic function (Allen-Zhu and Hazan, 2016). Objective (2) measures the accomplishment of sensing tasks. Eq. (3) limits the total sensor value collected by drones to the required one at maximum. Constraint (4) models the energy constraint of the drone u, where P_u^f and P_u^h are the flying and hovering power consumption respectively (Stolaroff et al., 2018); τ_u is the total flying time that drone u travels without hovering; f is the frequency with which drones collect sensor data as they hover over a cell.

EPOS optimizes sensing quality via a coordinated selection among alternative plans generated locally by the drones (i.e., navigation and sensing options). However, plans generation also results in a series of new problems, including *how to select the cells*, *how to determine collected sensing values*, *how to calculate the energy cost of a plan*, and *how to prevent more than two drones to occupy the same cell at the same time*. Therefore, this paper proposes a novel plan generation strategy to solve these problems such that drones coordinate efficiently to achieve high-quality sensing.

Algorithm 1: The local plan generation strategy for each drone.

Input: Power consumption of drone u for flying P_u^f and hovering P_u^h, the battery capacity C_u, the targets $T = (T_1, ..., T_N)$ and their coordinates, the base station B for departure/return and its travel range K_B, the total number of visited cells $|J_u|$, the number of generated plans P.

1 **Initialization**: Initialize a set of plans $\hat{\mathbb{P}} = \emptyset$;
2 **for** *each plan index* $p := 1, ..., P$ **do**
3   **Path calculation**: Find J_u via the K-nearest search within the range K_B;
4   Find the shortest path among visited cells and base station B via the greedy algorithm;
5   **Energy calculation**: Calculate the flying energy $E^f(J_u)$ based upon the path; Determine the energy utilization ratio e based on p via Eq. (5);
6   Calculate the maximum hovering energy $E^h(J_u)$ via Eq. (6);
7   Calculate the total sensing values $S(J_u)$ to collect via Eq. (7) and the total targets $T(J_u)$ via Eq. (9);
8   **Sensing allocation**: Allocate the sensing values $S_{u,n}$ proportionally to the visited cells via Eq. (8);
9   **Plan generation**: Calculate the cost of the plan $E(J_u)$ via Eq. (10);
10   Generate the plan $\mathbb{P}$ of size $N \times M$, and add it to the set $\hat{\mathbb{P}}$ of sensing plans with $E(J_u)$;
11 **end**

Output: Set of plans $\hat{\mathbb{P}}$ for drone u.

---

4. Proposed method

Algorithm 1 outlines the following steps of the proposed plan generation strategy for a swarm of drones.

Initialization. Given a drone u, the inputs of the algorithm are listed: The flying and hovering power consumption are calculated based on the power consumption model (Stolaroff et al., 2018). The parameters of drone u $\{m_b, m_e, d, r, v, F_d, \epsilon\}$ and environmental parameters such as air density and air speed are determined (see Table 3 and Appendix A). Each drone u also needs the information of the map including the coordinates and sensing requirements of cells, as well as the base stations of departure and return.

Next, according to the objective functions of Eq. (1) and (2), the total number of visited cells $|J_u|$ is determined empirically (see Section 6.1) using one of the three policies: (1) *min sensing mismatch*, (2) *min mission inefficiency*, and (3) *balance*. Min sensing mismatch focuses on avoiding over-sensing and under-sensing, while min mission inefficiency minimizes the uncollected sensing data. The policy for balance is a trade off between the first two policies. The policy of min sensing mismatch has a larger number of visited cells, while min mission inefficiency has a lower one. The two theorems in Appendix B provide the theoretical foundations behind the design of these policies. The algorithm also initializes the set $\hat{\mathbb{P}}$ for the plans of drone u.

Path calculation. At the beginning of each round, the algorithm finds the set of visited cells indices J_u via the K-nearest search. It selects the first cell randomly from K_B (a range of cells close to base station B). This range is calculated based on the relative distance between base stations. Then, the algorithm searches the nearest cell (within K_B) to the previous selected one until finding other $|J_u| - 1$ cells. After this, the algorithm finds the shortest possible path among the cells of J_u and the base station B via the greedy algorithm for traveling salesmen problem (Kizilateş and Nuriyeva, 2013), and calculates the traveling time τ_u excluding

Table 3
Notations for drones (Stolaroff et al., 2018).

Notation	Description	Value
m_b	mass of drone body	1.07 kg
m_e	mass of battery	0.31 kg
d	diameter of propellers	0.35 m
r	number of propellers	4
υ	ground speed	6.94 m/s
F_d	drag force	4.1134 N
ϵ	power efficiency	0.8
C	battery capacity	275 kJ
f	sensing frequency	60

hovering. Note that the drone returns to B at the end of period, and thus the path starts and ends at B. Taking an example in Fig. 1, the first plan of Drone 1 has the set of visited cells $\{A, C, D\}$.

Energy calculation. The algorithm determines the maximum energy consumption E_u^{max} of the drone before calculating the collected sensor values and the corresponding energy cost. The algorithm uses the energy utilization ratio e to compute the maximum energy constraint $C_u \cdot e$. The ratio can be expressed as:

$$e = 1 - \frac{p}{\delta \cdot P}, \tag{5}$$

where δ is a constant to determine energy utilization. The ratio limits the energy consumption of drones, and gives them flexibility to select plans with varying cost, i.e. energy consumption. Next, the algorithm calculates the hovering energy consumption $E^h(J_u)$ using the flying power consumption P_u^f:

$$\begin{aligned} E^h(J_u) &= C_u \cdot e - E^f(J_u), \\ &= C_u \cdot e - P_u^f \cdot \tau(J_u). \end{aligned} \tag{6}$$

The total sensor values to collect among J_u visited cells $S(J_u)$ is then calculated as follows:

$$S(J_u) = \sum_{n \in \mathcal{N}} S_{u,n} = \frac{E^h(J_u)}{P_u^h} \cdot f, \tag{7}$$

where P_u^h is the power consumption for hovering. Both P_u^f and P_u^h are computed by the power consumption model (Stolaroff et al., 2018) (see Appendix A).

Sensing allocation. To determine $S_{u,n}$ in each cell n, the algorithm allocates the collected sensor values among J_u visited cells proportionally to the target; a higher number of sensor values is collected from the cell with a higher target value. The equation is shown as follows:

$$S_{u,n} = \begin{cases} S(J_u) \cdot \frac{T_n}{T(J_u)}, & n \in J_u \\ 0, & otherwise \end{cases}, \tag{8}$$

$$where \quad T(J_u) = \sum_{n \in J_u} T_n. \tag{9}$$

The proportional sensing allocation aims to improve the matching between the total sensor values collected and the required ones. We also prove its high performance by comparing it to the equal allocation (mean) $S_{u,n} = \frac{S(J_u)}{|J_u|}$ (see Section 6.1).

Plan generation. With the energy consumption of hovering and flying, the algorithm computes the cost of the plan $E(J_u)$:

$$E(J_u) = C_u \cdot e. \tag{10}$$

Furthermore, the algorithm avoids more than two drones occupying the same cell at the same time. It generates a plan encoded by a matrix of size $N \times M$, where M denotes the total number of time units. Each element in the matrix can be represented by $h_{m,n}^u \in \{0,1\}$, $m = 1, 2, \ldots, M$, $n = 1, 2, \ldots, N$, $\sum_u h_{m,n}^u \leq 1$ $\sum_m h_{m,n}^u \leq M$, $\sum_n h_{m,n}^u \leq N$, where $h_{m,n}^u = 1$ denotes the drone occupies the cell n at time unit h, whereas $h_{m,n}^u = 0$ denotes the drone does not occupy the cell. The targets in EPOS are also set as a matrix of size $N \times M$, whose elements are set as 1 to prevent more than two drones occupying the same cell at same time. Finally, the algorithm adds the plan and its energy cost to the set $\hat{\mathbb{P}}$.

5. Experimental methodology

This section introduces the experimental settings including the sensing map, the drones and the learning algorithm at first. The metrics and baselines used for the performance evaluation are also introduced.

Table 4
Parameters of the EPOS algorithm (Pournaras et al., 2018).

Parameters	Value
Number of agents/drones	1000
Number of plans per agent	64
Network communication topology	balanced binary tree
Number of repetitions	40
Number of iterations	40
Non-linear cost function	Min RSS (Unit-Length)
Energy utilization parameter	$\delta = 8$
Behavior of agent	$\beta = 0$
Number of tested maps	200

5.1. Experimental settings

Sensing map. A square area of size 1600×1600 m split into a finite number of cells is studied. Each cell is defined as a rectangular square that can be captured by the cameras of drones (see Fig. 1). The target values, or the sensing requirements of hovering time are distributed according to a Beta distribution. The base (charging) stations are uniformly distributed in the map. There are three types of scenarios for the experimental evaluation of this paper:

- *Basic synthetic scenario.* It has 4 base stations, 64 cells and the total target values of 20000 to collect over 48 time periods, which correspond to one day. Each period lasts 30 min and is divided into 12 time units, each of equal length. We dispatch 1000 drones over all periods, during which approximate 20 drones sense the area (camera recording) in parallel. The purpose to use this map is to compare the performance of different plan generation policies in the proposed method.
- *Complex synthetic scenario.* It varies the parameter settings such as the number of dispatches, the number of base stations and cells as well as the total target values. The goal is to assess the scalability of the proposed method in different experimental conditions.
- *Transportation scenario.* It originates from the central business district of Athens, where a swarm of drones use cameras to record traffic flows. The goal is to assess the accuracy and efficiency of the proposed method in the real-world traffic monitoring. More details are given in Part 6.4 of this section.

Drones. We assume that drones are of the same type (DJI Phantom 4 Pro), equipped with the same type of battery (6000 mAh LiPo 2S) and camera (4K) to capture images/videos (Company DJI, 2022), and thus they have the same power consumption (Stolaroff et al., 2018) and battery capacity. To ensure the camera of a drone covers the whole area of a cell (see Fig. 1), the minimum hovering height of drones is determined at which the field of view of the camera and the cell overlap. Based on the distance between any two cells D (approximately 200 m in the basic synthetic scenario), the hovering height H is computed using the pixels PX, focal length derived from the camera calibration c_k and ground sampling distance GSD, with the formula: $H = \text{GSD} \cdot c_k / \text{PX}$ (Wierzbicki, 2018). Thus, in this paper, each drone is equipped with a $4K$ camera, sensing from a minimum height of 82.4 meter based on the pixels and field of view that a $4K$ camera has. The drone parameters and their description are summarized in Table 3.

Learning algorithm. We generate plans for each of the 1000 agents. All generated plans are made openly available to encourage further research on coordinated spatio-temporal sensing of drones (Qin and Pournaras, 2022). Each agent in EPOS is mapped to a drone. During the coordinated plan selection via EPOS,[1] agents self-organize into a balanced binary tree as a way of structuring their learning interactions (Pournaras et al., 2018). The shared goal of the agents is to minimize the residual sum of squares (RSS) between the total sensor values collected and the total required target values, both in unit-length scaled. The RSS is a proxy to optimize the two objectives defined in Eq. (1) and (2), and can be formulated as: $\log_{10}(\sum_{n \in \mathcal{N}} (\sum_{u \in \mathcal{U}} S_{u,n} - T_n)^2)$. The algorithm repeats 40 times by changing the random position of the agents on the tree.[2] At each repetition, the agents perform 40 bottom-up and top-down learning iterations during which RSS converges to the minimum optimized value. For the validation of the proposed algorithm, a number of 1000 basic synthetic scenarios are generated, each with a new distribution of cells and target values. The algorithm requires at least 200 sensing maps as input to output stable results (see Fig. 7(b) in Appendix C). Therefore, a number of 200 random tested maps are set for the following experiments. The computational cost of EPOS is $O(P \cdot I)$ for each agent and $O(P \cdot I \cdot \log U)$ for the system, where I is the total number of iterations. The communication cost is $O(I)$ for each agent and $O(I \cdot \log U)$ for the system. Earlier comparisons of this complexity with state-of-the-art approaches demonstrate the cost-effectiveness of EPOS (Pournaras et al., 2018). Table 4 summarizes the optimization parameters of EPOS.

5.2. Metrics and baselines

To evaluate the sensing quality, i.e., the accomplishment of sensing targets, we introduce two performance metrics:

---

[1] EPOS is open-source and available at: https://github.com/epournaras/EPOS.
[2] More information about the influence of the tree topology and agents' positioning on the tree is illustrated in earlier work (Nikolic and Pournaras, 2019; Pournaras et al., 2020).

Table 5
Performance of two implementations in *Greedy-sensing*.

Implementation	Global view	Local view
Sensing Mismatch	2.79	3.21
Mission Inefficiency (%)	19.80	26.11

Table 6
Results for the number of visited cells in *Round-robin*.

Number of visited cells:	5	6	7	8	9	10
Sensing Mismatch	2.24	2.40	2.34	1.86	3.01	3.79
Mission Inefficiency (%)	25.93	40.42	62.59	75.51	85.69	96.68

- *Sensing mismatch.* It denotes the approximation error[3] between the total sensing values collected and the target values. A low sensing mismatch prevents the cases of over-sensing and under-sensing (Qin et al., 2022). It is formulated as log_{10} of Eq. (1).
- *Mission inefficiency.* It denotes the ratio of sensing values in all cells that are not collected by the drones during their mission over the total target values in all cells. It is formulated as Eq. (2).

In simple words, the sensing mismatch measures the data sampling quality, while the mission inefficiency measures the completeness of the required collected data.

A fair comparison of the proposed method with related work is not straightforward as there is a very limited number of relevant decentralized algorithms. These algorithms (Alighanbari and How, 2006; Chen et al., 2022) cannot be directly applied to this large-scale task assignment problem while respecting the energy constraints of drones. For this reason, we use as baselines for comparison three state-of-the-art centralized sensing methods capable of performing multi-drone task optimization: (1) *Greedy-sensing* (Bartolini et al., 2019), (2) *Round-robin* (Alwateer and Loke, 2019) and (3) *Min-energy* (Pournaras et al., 2018).

- *Greedy-sensing.* It requires a drone to complete the required sensing tasks of the cells one by one without violating the battery constraint. This method reduces the number of visited cells and traveling distance compared to the proposed method; drones spend more energy on sensing than traveling. We implement the method on a centralized coordinator that has a global view of the remaining sensor values required such that over-sensing and under-sensing are prevented. Table 5 illustrates the higher performance of the method with a global view vs. a version with a local view, i.e., no knowledge of the remaining sensor values required.
- *Round-robin.* It comes in sharp contrast to *Greedy-sensing*. Drones visit the same number of cells and spend more energy on traveling than *Greedy-sensing*. According to the results shown in Table 6, the number of visited cells is divided equally into 8 for each drone as it has the minimum sensing mismatch.
- *Min-energy.* It minimizes the total energy consumption and does not sacrifice energy for improving sensing quality. This method is implemented by changing the behavior of EPOS agents to $\beta = 1$ such that the agents select the plans with the lowest energy consumption cost. No coordination is performed in this case.

6. Performance evaluation

This section assesses the different system parameters and illustrates the results of the three evaluated scenarios: basic synthetic, complex synthetic and transportation scenario. Analytical results about how coordinated mobility influences the sensing performance are illustrated in Appendix B.

6.1. Effect of different parameters

The purpose of the evaluation in this section is to understand how different parameters influence the system performance. We use the results of this section to make an empirical choice of these parameters for the rest of the evaluation scenarios. The calculations presented here can be automated for any scenario within a hyper-parameter optimization, which is though not the focus of this paper.

We use *EPOS-balance* and the basic synthetic scenario to test the mobility range, the number of generated plans, energy utilization, the proportional sensing allocation and the behavior of agents in EPOS.

Mobility range. Three policies of plan generation are compared in EPOS: *balance*, *min sensing mismatch* and *min sensing inefficiency*. A simple test is implemented in the basic synthetic scenario to compare different numbers of visited cells set in the input, see Algorithm 1. As shown in Fig. 2(a), the higher the number of cells a drone visits, the higher the flying energy is, and the lower the allocated hovering/sensing energy is, which increases the mission inefficiency. To balance the sensing mismatch and mission

---

[3] Error and correlation metrics such as the root mean squared error, cross-correlation or residuals of summed squares can estimate this matching, which is shown earlier to be an NP-hard combinatorial optimization problem in this context (Pournaras et al., 2018; Pournaras, 2020).
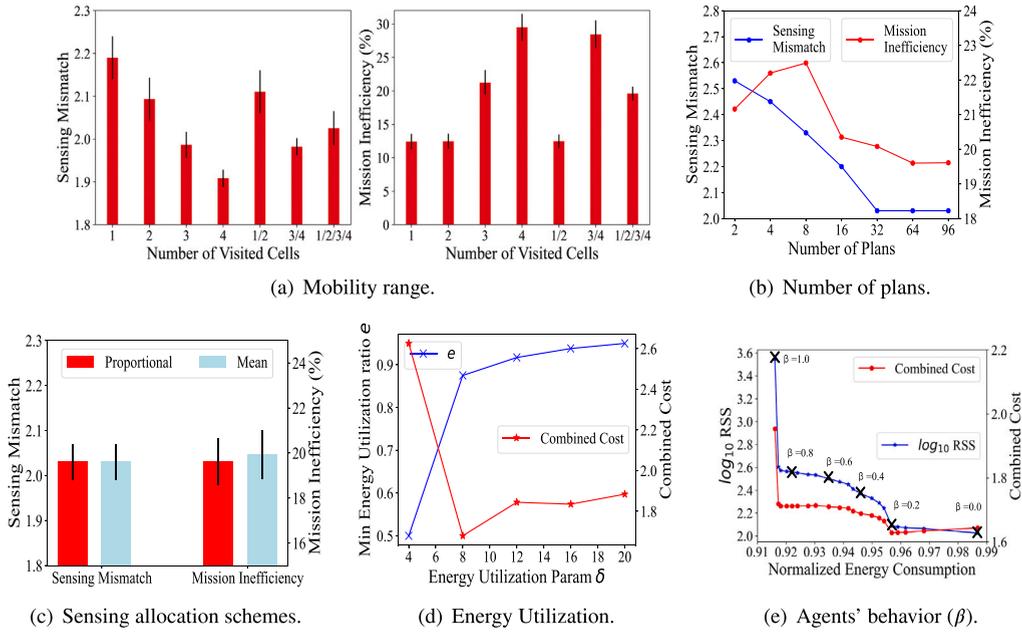
Fig. 2. Performance comparison for different parameters of the proposed method.

inefficiency, the policies of min sensing mismatch ($|J_u| = 1|2$, the strategy chooses 1 or 2 cells randomly), min mission inefficiency ($|J_u| = 3|4$), and balance ($|J_u| = 1|2|3|4$) are empirically designed. Thus, three policy-based methods for EPOS are proposed that are referred to as *EPOS-mismatch*, *EPOS-inefficiency* and *EPOS-balance* respectively.

Number of plans. See Fig. 2(b), as the number of generated plans increases, both sensing mismatch and mission inefficiency decrease and converge to 64. Thus, 64 plans for each agent are generated in EPOS.

Sensing allocation: proportional vs. mean. Fig. 2(c) illustrates the performance comparison between proportional and mean sensing allocation defined in Eq. (8). The mean sensing allocation has higher inefficiency than proportional one. Using the Mann–Whitney U test, the sensing mismatch distribution of the two methods comes with $p = 0.08$, the one of mission inefficiency has $p = 0.003$. The results demonstrate a statistically higher performance when sensor values collected are proportional, i.e. a similar mission mismatch but a 1.56% higher mission efficiency.

Energy utilization. Fig. 2(d) illustrates the changes of minimum energy utilization ratio e (when $p = P$ in Eq. (5)) and the combined cost as the parameter δ increases. The combined cost is the sum of normalized total energy consumption, sensing mismatch and mission inefficiency. According to the results, a $\delta = 8$ is selected for the experimental settings.

Agents' behavior and Pareto optimality. Fig. 2(e) illustrates the effect of agents' behavior by varying the parameter β (Pournaras et al., 2018). As β increases from 0 to 1, agents reduce the energy cost of their selected plans, while increasing the sensing mismatch. This is because drones with higher β choose a plan with lower energy cost, i.e., the plan with lower energy utilization ratio e, and the total sensing values collected is reduced according to Eq. (6) and (7). To minimize the log_{10} RSS, the value of $\beta = 0$ is selected in the proposed methods, while $\beta = 0.2$ is the calculated Pareto optimal point referred to as *EPOS-Pareto*, minimizing the combined cost.

6.2. Evaluation on the basic synthetic scenario

As shown in Fig. 3, the three optimization methods based on EPOS coordinate drones to select the plans with the minimum energy utilization ratio e and result in the lowest total energy consumption except *Min-energy* ($\beta = 1.0$) and *EPOS-Pareto* ($\beta = 0.2$). The coordination of drones achieves the lowest sensing mismatch compared to baselines. Therein, *EPOS-balance* (with a total energy consumption of 271467 kJ and sensing mismatch of 2.02) has the mission inefficiency of 19.6% that is close to *Greedy-sensing*; *EPOS-mismatch* sacrifices mission efficiency to obtain a very low sensing mismatch among three policies, and just 0.12 higher than *Round-robin*; *EPOS-inefficiency* has the lowest mission inefficiency of 12.49% among all methods. In contrast, without coordination, the *Min-energy* method lowers energy consumption to 233,484 kJ at a cost of higher sensing mismatch (3.74) and mission inefficiency (47.5%). *Greedy-sensing* and *Round-robin* also come with higher sensing mismatch (2.79) and mission inefficiency (75.51%) respectively.

Fig. 3(d) illustrates the combined cost calculated from Figs. 3(a)–3(c). The proposed method is superior to baseline methods especially when combining all performance metrics.
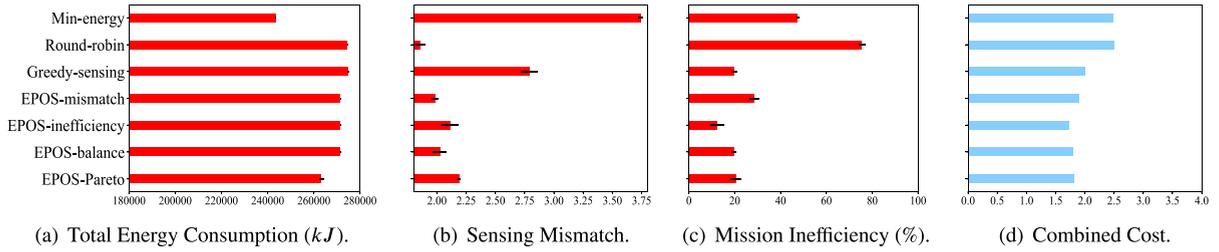
(a) Total Energy Consumption (kJ).   (b) Sensing Mismatch.   (c) Mission Inefficiency (%).   (d) Combined Cost.

Fig. 3. Performance comparison of the six methods on the basic synthetic scenario: 4 base stations, 64 cells and 20 000 total target values.
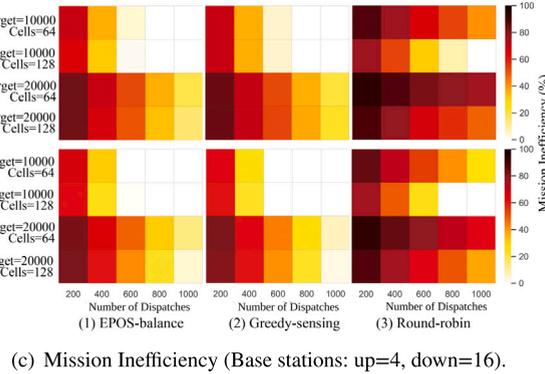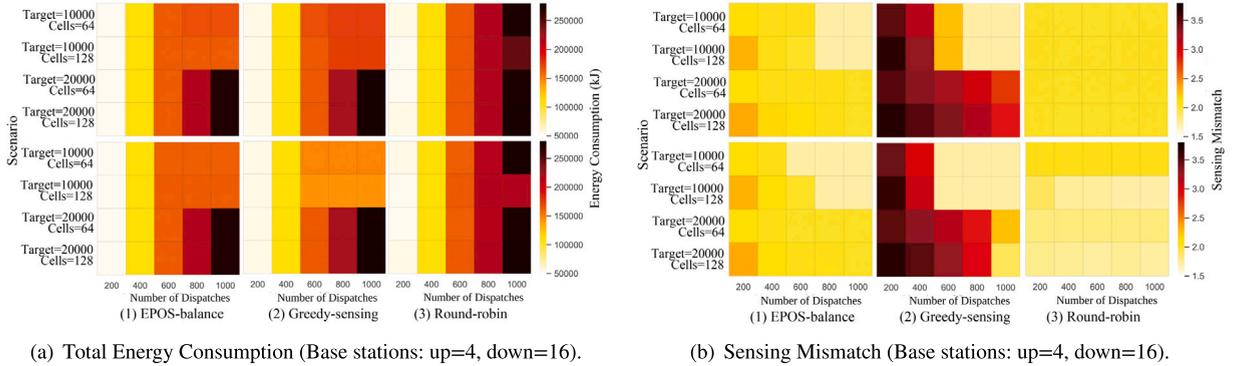

(a) Total Energy Consumption (Base stations: up=4, down=16).   (b) Sensing Mismatch (Base stations: up=4, down=16).

(c) Mission Inefficiency (Base stations: up=4, down=16).

Fig. 4. Performance comparison under varying parameters: total target values (10 000 and 20 000), the number of cells (64 and 128), the number of base stations (heatmap triplets: up = 4, down = 16), and the number of drone dispatches (from 200 to 1000).

6.3. Evaluation on the complex synthetic scenario

There are four dimensions we study here: (1) the number of dispatched drones used in the sensing mission, (2) the total target values to collect, (3) the number of cells into which the same map is split, and (4) the number of base stations. Fig. 4 shows the performance comparison between *EPOS-balance*, *Greedy-sensing* and *Round-robin* methods.

Number of dispatches. Take an example of the complex synthetic scenario (target = 20000, cells = 64, base stations = 4). As the number of dispatches increases, the total energy consumption of *EPOS-balance* rises, while mission inefficiency decreases proportionally (from 271467 kJ, 83% to 54268 kJ, 19% in Figs. 4(a) and 4(c)). This shows that drones take full advantage of energy resources to collect sensor values efficiently. Compared to *Round-robin*, *EPOS-balance* collects more data (exceeds the average of 38.5%) with a lower number of dispatches and energy. Furthermore, as the number of dispatches decreases, *EPOS-balance* shows a lower increase of sensing mismatch (increased by 4.21%) than *Greedy-sensing* (increased by 26.17%). If some drones fail to be dispatched due to attacks or other factors, *EPOS-balance* is proved to effectively mitigate the penalties of over-sensing and under-sensing, which validates the resilience of the proposed method.

Total target values. If the total target values to collect decrease from 20000 to 10000, the mission inefficiency of *EPOS-balance* reduces to 0 and the energy consumption keeps constant when the number of dispatches exceeds 500. This is because the total

Fig. 5. A swarm of 10 drones hovering over the central business district of Athens over five days to record traffic flows in a congested area of a 1.3 km² with more than 100 km lanes of road network, around 100 busy intersections, more than 50 bus stops and close to half a million trajectories. There are 5 drones that depart from cell $H1$ and travel to the cell 1, 2, 3, 5 and 10 (shown in blue), whereas the other 5 drones set off from $H2$ and hover over the rest of cells (shown in green) (Barmpounakis and Geroliminis, 2020).

target values represent the total hovering time of drones, and *EPOS-balance* does not waste energy resources when drones complete their sensing tasks. In Fig. 4(b), the sensing mismatch of *Greedy-sensing* is highly sensitive to the change of target values. It increases by 1.09, while *EPOS-balance* increases only by 0.34.

Number of cells. If the number of cells increases from 64 to 128, the size of each cell decreases, and the distance between any two cells decreases, which reduces the travel distance of drones. As a consequence, drones spend more energy on collecting sensor values according to Eqs. (6) and (7), which reduces the mission inefficiency. In Fig. 4(c), *EPOS-balance* shows a higher decrease in mission inefficiency (decreased by 4%) than *Greedy-sensing* (decreased by 1%) as the number of cells increases. Furthermore, *EPOS-balance* collects higher quality data with a lower number of cells and drones as shown in Fig. 4(b).

Number of base stations. If the number of base stations increases from 4 to 16, the travel distances of drones reduce and the total sensor values collected rise. As a result, mismatch and inefficiency of *EPOS-balance* decrease. *EPOS-balance* shows an overall superior performance as targets and base stations vary.

In summary, *EPOS-balance* has the lowest sensing mismatch (close to *Round-robin*) and mission inefficiency (close to *Greedy-sensing*) as the variables in the environment change. This confirms that *EPOS-balance* has the highest performance in overall under the same battery constraints and different sensing scenarios.

6.4. Traffic monitoring by coordinated drones

This paper shows the significant and broad impact of the proposed coordinated sensing method on a transportation scenario, where a swarm of drones perform traffic monitoring. The evaluation uses real-world data from vehicle trajectories collected by a swarm of drones in the congested downtown area of Athens, Greece (Barmpounakis and Geroliminis, 2020). This application scenario envisions a large-scale and long-term monitoring of traffic congestion using coordinated drones to measure and predict traffic patterns. As shown in Fig. 5, a swarm of 10 drones hover over 10 cells for 20 time periods to record traffic streams of 6 types of vehicles: *car, taxi, bus, medium vehicle, heavy vehicle* and *motorcycle*. Each drone departs from and returns to one of the two base stations and hovers up to 25 min at a single cell.

Sensing model. We model the application scenario of coordinated drones that perform traffic sensing. To improve the sensing quality, we calculate the required sensing tasks (or targets) to be proportional to the spatio-temporal normalized distribution of vehicles shown in Fig. 6(a). We assume that typical distributions can be derived from historical data (Barmpounakis and Geroliminis, 2022). The higher the likelihood a vehicle type drives through a cell, the higher the target value is, and $T_n \leq (25 \times 20)$, $n = 1, 2, \ldots, 10$. Over 20 time periods (30 min), a certain number of drones in each period is dispatched.

Metrics and Baselines. The performance metrics are adjusted to evaluate more directly the observability of vehicles in this traffic monitoring scenario: *accuracy* and *efficiency*. Accuracy A denotes the correlation between the distributions of observed vehicles and actual existing ones. Efficiency E denotes the ratio of actual vehicles captured by the drones over the total ones. They are formulated as follows:
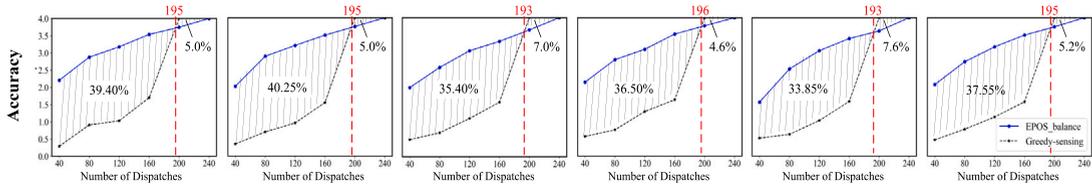
$$A = \log_{10}\left(\frac{1}{\sum_{m=1}^{M}(V_m^* - V_m)^2}\right), \tag{11}$$

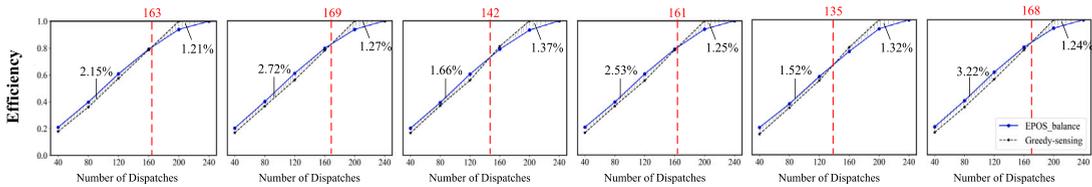$$E = \frac{\sum_{m=1}^{M} V_m^*}{\sum_{m=1}^{M} V_m}, \tag{12}$$

where V_m indicates the total number of vehicles at time unit m acquired from pNEUMA (Barmpounakis and Geroliminis, 2020) (we set one minute as a time unit), and V_m^* denotes the number of vehicles monitored by drones at time unit m using the proposed
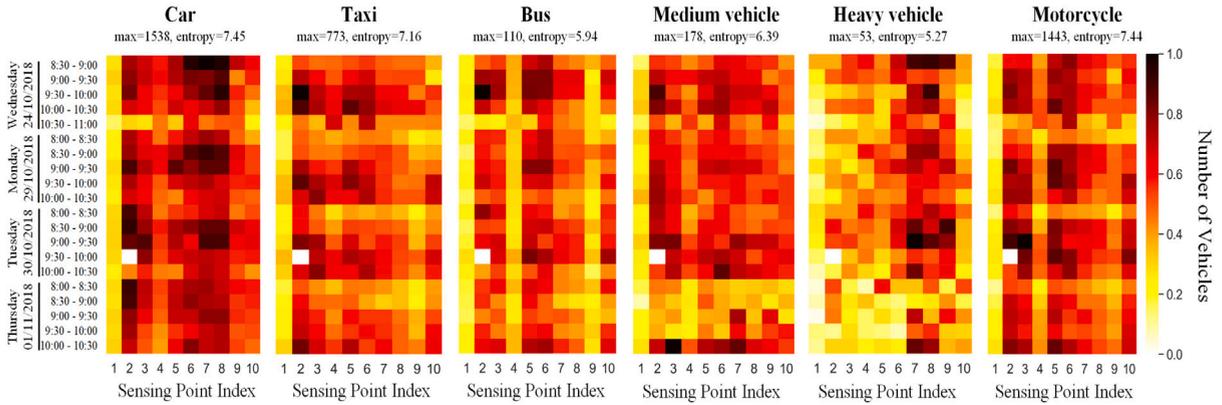
(a) Spatio-temporal distribution of vehicles.

(b) Accuracy (correlation) of vehicles detection.

(c) Efficiency of vehicles detection.

Fig. 6. The comparison results among six types of vehicles (car, taxi, bus, medium vehicle, heavy vehicle, and motorcycle) in a downtown area of Athens based on the open traffic monitoring data collected by a drone swarm.

method, $V_m^* = \sum_{n \in \mathcal{N}} h_{m,n}^u \cdot V_m$. Moreover, the *Greedy-sensing* method is used as the baseline. It requires a low energy consumption and emulates the vehicles data collection in pNEUMA (Barmpounakis and Geroliminis, 2020).

Experimental results. Fig. 6 illustrates the performance comparison between the proposed *EPOS-balance* and the baseline for varying numbers of dispatched drones used in traffic monitoring. The results in Fig. 6(b) show that the accuracy of *EPOS-balance* among six types of vehicles is significantly higher and more stable than that of *Greedy-sensing* when the number of dispatches is lower than 200. This is because with the total of 200 drones' dispatches under *Greedy-sensing*, each cell within each period is monitored by exactly one drone. With a lower number of drones' dispatches (i.e., scarce resources), however, *EPOS-balance* coordinates drones to monitor all cells over all time periods, preventing over-sensing and under-sensing. Fig. 6(c) illustrates that *EPOS-balance* is relatively more efficient than *Greedy-sensing* with no more than 160 dispatches. This is because *EPOS-balance* coordinates drones to collect sensor values that are proportional to the distributions of vehicles, which increases the number of vehicles observed by drones. Note that there is a strong linear relationship between the threshold for the number of dispatches (see Fig. 6(c)) and the entropy of vehicles distribution (see Fig. 6(a)), with a Pearson coefficient correlation of 0.93 and corresponding *p*-value of 0.007.

In summary, for lower than 160 dispatches, *EPOS-balance* is approximately 46.45% more accurate and 2.88% more efficient than *Greedy-sensing* among six vehicle types monitored. This verifies the remarkable performance of the proposed method under a scarce number of drone resources, requiring less than 80% of the drones to achieve equivalent or higher performance than *Greedy-sensing*.

7. Conclusion and future work

In conclusion, this paper illustrates a decentralized coordination model to solve the task self-assignment problem for scalable, resilient and flexible spatio-temporal sensing by a swarm of drones. To ensure energy-aware and efficient coordination of sensing,

this paper introduces a novel plan generation strategy with three policies. Extensive experiments demonstrate that the proposed method is adaptive to complex sensing scenarios and has better sensing performance than existing methods. The evaluation on real-world data shows that the designed model has a strong potential in optimizing traffic monitoring in the Smart Cities using a limited number of drones.

Nevertheless, the proposed method can be further improved towards several research avenues: (1) Use of real-world datasets in other applications of Smart Cities, including last-mile delivery, disaster response, and smart farming. (2) Use of other learning method, such as multi-agent reinforcement learning algorithm, to extend the study of coordinated drones with charging capabilities and obstacle/collision avoidance. (3) Use of efficient wireless communication technology and special hardware to implement the coordination capability on board and online.

CRediT authorship contribution statement

Chuhao Qin: Conceptualization, Methodology, Software, Validation, Investigation, Formal analysis, Data Curation, Writing – original draft. **Evangelos Pournaras:** Conceptualization, Funding acquisition, Resources, Supervision, Writing – review & editing.

Data availability

The open dataset contains all the plans of the studied scenario: https://figshare.com/articles/dataset/EPOS-based_Plans_for_Drones/21432804.

Acknowledgments

Appendix A. Power consumption model

Drones spend energy to surpass gravity force and counter drag forces due to wind and forward motions. A drone controls the speed of each rotor to achieve the thrust T and pitch θ necessary to stay aloft and travel forward at the desired velocity while balancing the weight and drag forces. For a drone with mass m_b and its battery with mass m_c, we define the total required thrust as follows:

$$\mathcal{T} = (m_b + m_c) \cdot g + F_d, \tag{13}$$

where g is the gravitational constant, and F_d is the drag force that depends on air speed and air density. For steady flying, the drag force can be calculated by the pitch angle θ as:

$$F_d = (m_b + m_c) \cdot g \cdot tan(\theta). \tag{14}$$

Building on the model in Stolaroff et al. (2018), the power consumption with forward velocity and forward pitch is given by:

$$P^f = (v \cdot sin\theta + v_i) \cdot \frac{\mathcal{T}}{\epsilon}, \tag{15}$$

where v is the average ground speed; ϵ is the overall power efficiency of the drone; v_i is the induced velocity required for given T and can be found by solving the nonlinear equation:

$$v_i = \frac{2 \cdot \mathcal{T}}{\pi \cdot d^2 \cdot r \cdot \rho \cdot \sqrt{(v \cdot cos\theta)^2 + (v \cdot sin\theta + v_i)^2}}, \tag{16}$$
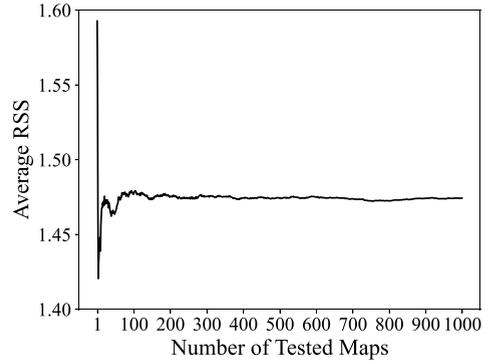
where d and r are the diameter and number of drone rotors; ρ is the density of air.

Moreover, the power consumption for hovering of a drone is calculated by:

$$P^h = \frac{\mathcal{T}^{3/2}}{\epsilon \cdot \sqrt{\frac{1}{2}\pi \cdot d^2 \cdot n \cdot \rho}}. \tag{17}$$
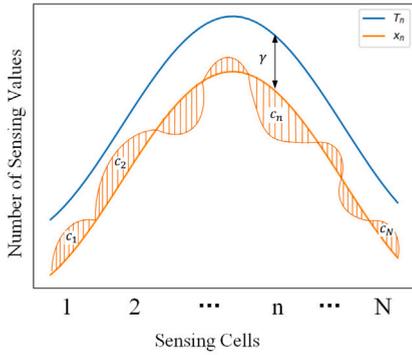
(a) An illustration to assist the proof of Theorem B.2: The distribution of targets and total sensing values collected by drones.

(b) Average residual sum of squares as the number of basic scenarios increases.

Fig. 7. Figures for Appendices B and C.

Appendix B. Mobility and sensing quality: Analytical results

Two theorems are introduced that link the performance metrics of mission inefficiency and sensing mismatch with the mobility of the coordinated drones, in particular with their flying coverage modeled by the number of visiting cells.

Theorem B.1. *In a mission of a swarm of drones $\mathcal{U}$, starting from their base stations, flying with a constant ground speed over an area consisting of N cells to collect sensor data, and returning back consuming all energy of their battery, the mission inefficiency is proportional to the number of random visited cells $|J_u|$:*

$$\alpha_1 \cdot |J_u| \rightarrow f_1(J_u) = 1 - \frac{\sum_{u \in \mathcal{U}} S(J_u)}{\sum_{n \in \mathcal{N}} T_n}, \tag{18}$$

if the drones collect sensor values over the visited cells proportionally to required target values, where α_1 is a positive constant and $f_1(J_u)$ is the function of mission inefficiency according to the optimization objective of Eq. (2).

Proof. Based on Eq. (6) and (7) and since each drone uses all its battery capacity for flying and hovering, the number of collected sensor values over the cells J_u is:

$$S(J_u) = \sum_{n \in \mathcal{N}} S_{u,n} = \frac{C_u \cdot e - P_u^{\mathrm{f}} \cdot \tau(J_u)}{P_u^{\mathrm{h}}} \cdot f. \tag{19}$$

Eq. (2) can be reformulated as:

$$1 - \frac{\sum_{u \in \mathcal{U}} \sum_{n \in \mathcal{N}} S_{u,n}}{\sum_{n \in \mathcal{N}} T_n} = 1 - \frac{\sum_{u \in \mathcal{U}} S(J_u)}{\sum_{n \in \mathcal{N}} T_n} := f_1(J_u), \tag{20}$$

given that $\sum_{u \in \mathcal{U}} \sum_{n \in \mathcal{N}} S_{u,n} = \sum_{u \in \mathcal{U}} S(J_u)$. The total flying time $\tau(J_u)$ of a drone u can be modeled as:

$$\tau(J_u) \approx (|J_u| - 1) \cdot \tilde{\tau} + 2 \cdot \tilde{\tau}_B, \tag{21}$$

where $\tilde{\tau}$ is the mean expected traveling time between any two random cells and $\tilde{\tau}_B$ is the mean expected traveling time between the base station of the drone and a random cell. Assume the number of random visited cells increases from $|J_u|$ to $|J_u'|$, where $|J_u| < |J_u'|$. Then, the total flying time without hovering is also likely to increase as $\tau(J_u) < \tau(J_u')$ given that each drone u flies with a constant ground speed. The influence $f_1(J_u') - f_1(J_u)$ on the mission inefficiency based on Eq. (20) is calculated as follows:

$$
\begin{aligned}
f_1(J_u') - f_1(J_u) &= \frac{\sum_{u \in \mathcal{U}} (\overbrace{S(J_u)}^{Eq.(19)} - \overbrace{S(J_u')}^{Eq.(19)})}{\sum_{n \in \mathcal{N}} T_n} \\
&= \frac{\sum_{u \in \mathcal{U}} \frac{P_u^{\mathrm{f}} \cdot f}{P_u^{\mathrm{h}}}}{\sum_{n \in \mathcal{N}} T_n} \cdot (\overbrace{\tau(J_u')}^{Eq.(21)} - \overbrace{\tau(J_u)}^{Eq.(21)}) \\
&\approx \frac{\sum_{u \in \mathcal{U}} \frac{P_u^{\mathrm{f}} \cdot f}{P_u^{\mathrm{h}}}}{\sum_{n \in \mathcal{N}} T_n} \cdot \tilde{\tau} \cdot (|J_u'| - |J_u|) \\
&\leftarrow \alpha_1 \cdot (|J_u'| - |J_u|).
\end{aligned}
\tag{22}
$$

Since all parameters in $\frac{\sum_{u\in U}\frac{P_u^f\cdot f}{P_u^h}}{\sum_{n\in\mathcal{N}}T_n}\cdot\tilde{\tau}$ are positive, $\alpha_1>0$, and therefore the mission inefficiency is proportional to the number of random visited cells. ∎

Theorem B.2. *In a mission of a swarm of drones U, starting from their base stations, flying with a constant ground speed over an area consisting of N cells to collect sensor data, and returning back consuming all energy of their battery, the sensing mismatch is inverse proportional to the number of random visited cells $|J_u|$:*

$$\alpha_2\cdot|J_u|\to f_2(J_u)=\sum_{n\in\mathcal{N}}\left(T_n-\sum_{u\in U}S_{u,n}\right)^2, \tag{23}$$

if and only if $|J_u'|+|J_u|<N$ when increasing the number of visited cells from $|J_u|$ to $|J_u'|$ and if the drones collect sensor values over the visited cells proportionally to required target values, where α_2 is a negative constant and $f_2(J_u)$ is a function of sensing mismatch according to the optimization objective of Eq. (1).

Proof. Fig. 7(a) assists this proof. Since sensor values over the different cells are collected proportionally to the required target values T_n, the collected sensor values are modeled by $x_n+v_n=\sum_{u\in U}S_{u,n}$, $\forall n\in\{1,\ldots,N\}$, and $x_n\leq T_n$. Each x_n corresponds to the collected sensor values with an optimal matching to the required target values (min RSS), while $v_n\in\{+c_n,-c_n,0\}$ models mismatches such that $\sum_{n\in\mathcal{N}}(x_n+v_n)\approx\sum_{n\in\mathcal{N}}x_n$, and thus $\sum_{n\in\mathcal{N}}v_n\approx0$. Moreover, the mission inefficiency in the optimal collected sensor data x_n is given by γ such that $\gamma:=T_n-x_n$. The optimal matching between T_n and x_n denotes that $\gamma\geq0$ is constant $\forall n\in\{1,\ldots,N\}$. Therefore, it holds that:

$$\begin{aligned}\sum_{n\in\mathcal{N}}\left(T_n-\sum_{u\in U}S_{u,n}\right)&=\sum_{n\in\mathcal{N}}(T_n-(x_n+v_n))\\&\approx\sum_{n\in\mathcal{N}}(T_n-x_n)\\&\approx N\cdot\gamma>0\end{aligned} \tag{24}$$

Eq. (24) can be squared to calculate the sensing mismatch $f_2(J_u)$ as follows:

$$\begin{aligned}f_2(J_u)&=\sum_{n\in\mathcal{N}}(T_n-(x_n+v_n))^2\\&=\sum_{n\in\mathcal{N}}(\gamma-v_n)^2\\&=N\cdot\gamma^2-2\gamma\cdot\sum_{n\in\mathcal{N}}v_n+\sum_{n\in\mathcal{N}}v_n^2\\&=N\cdot\gamma^2+\sum_{n\in\mathcal{N}}c_n^2.\end{aligned} \tag{25}$$

The higher the c_n is, the higher the $f_2(J_u)$.

The distribution of the mission inefficiency values $T_n-(x_n+v_n)$ among N cells is determined by the selection of the cells by each drone. By assuming that each of the U drones chooses the visiting cells randomly (with replacement), the distribution among N cells can be explained by a Binomial distribution:

$$P(X=k,J_u)=\binom{U}{k}\cdot p(J_u)^k\cdot(1-p(J_u))^{U-k}, \tag{26}$$

where $p(J_u)$ is the probability that a drone u chooses $|J_u|$ number of cells from the total of N cells that do not contain the cell n. This results in mismatch at cell n that either originates from (i) an under-sensing $v_n=-c_n$, if a drone u has a high probability $p(J_u)$ to miss cell n from J_u, or (ii) an over-sensing $T_n-(x_n+v_n)=C+c_n$ if this probability is low (see Fig. 7(a)). The probability $p(J_u)$ can be formulated as:

$$p(J_u)=\binom{N-1}{|J_u|}\Big/\binom{N}{|J_u|}=1-\frac{|J_u|}{N}, \tag{27}$$

which expresses that the higher the number of visiting cells is, the lower the probability of drone u to miss a cell n.

The mismatch c_n at cell n for a drone visiting J_u points can be modeled by a Binomial distribution:

$$c_n(J_u):=k_n(J_u)\cdot\overline{S}_n(J_u), \tag{28}$$

with the expected values of $k_n(J_u)$ denoting the average number of drones that miss cell n and $\overline{S}_n(J_u)$ denoting the average number of collected values by each drone u. The expressions of these values are formulated as follows:

$$k_n(J_u)=U\cdot p(J_u), \tag{29}$$

$$\overline{S}_n(J_u)=\sum_{u\in U}S(J_u)\cdot\frac{T_n}{U\cdot\sum_{n\in\mathcal{N}}T_n}. \tag{30}$$

By increasing the number of sensing cells from $|J_u|$ to $|J'_u|$, the influence $f_2(J'_u) - f_2(J_u)$ on the sensing mismatch can be formulated using Eq. (25) as follows:

$$
\begin{aligned}
f_2(J'_u) - f_2(J_u) &= \sum_{n \in \mathcal{N}} [\ \overbrace{c_n(J'_u)^2}^{Eq.((28),(29),(30))} - \overbrace{c_n(J_u)^2}^{Eq.((28),(29),(30))} \ ] \\
&= \sum_{n \in \mathcal{N}} \frac{T_n}{\sum_{n \in \mathcal{N}} T_n} \cdot \overbrace{p(J'_u)^2}^{Eq.(27)} \cdot \overbrace{\sum_{u \in \mathcal{U}} S(J'_u)^2}^{Eq.(22)} - \sum_{n \in \mathcal{N}} \frac{T_n}{\sum_{n \in \mathcal{N}} T_n} \cdot \overbrace{p(J_u)^2}^{Eq.(27)} \cdot \overbrace{\sum_{u \in \mathcal{U}} S(J_u)^2}^{Eq.(22)} \\
&\leftarrow \sum_{n \in \mathcal{N}} T_n \cdot [\frac{\alpha_1}{N} \cdot |J'_u|^2 - \alpha_1 \cdot |J'_u|] - \sum_{n \in \mathcal{N}} T_n \cdot [\frac{\alpha_1}{N} \cdot |J_u|^2 - \alpha_1 \cdot |J_u|] \\
&\leftarrow [\frac{|J'_u| + |J_u|}{N} - 1] \cdot \alpha_1 \cdot \sum_{n \in \mathcal{N}} T_n \cdot (|J'_u| - |J_u|) \\
&\leftarrow \alpha_2 \cdot (|J'_u| - |J_u|)
\end{aligned}
$$

(31)

Thus, it holds that $\alpha_2 = [\frac{|J'_u| + |J_u|}{N} - 1] \cdot \alpha_1 \cdot \sum_{n \in \mathcal{N}} T_n < 0$ if and only if $|J'_u| + |J_u| < N$, where in this case the sensing mismatch is proven to be reverse proportional to the number of random visited cells J_u. ∎

Appendix C. Results of experimental settings

Fig. 7(b) illustrates the required number of tested maps to achieve a stable mean global cost, measured by the residual of sum squares (RSS).

References

Alighanbari, Mehdi, How, Jonathan P., 2005. Decentralized task assignment for unmanned aerial vehicles. In: Proceedings of the 44th IEEE Conference on Decision and Control. IEEE, pp. 5668–5673.

Alighanbari, Mehdi, How, Jonathan, 2006. Robust decentralized task assignment for cooperative UAVs. In: AIAA Guidance, Navigation, and Control Conference and Exhibit. p. 6454.

Allen-Zhu, Zeyuan, Hazan, Elad, 2016. Variance reduction for faster non-convex optimization. In: International Conference on Machine Learning. PMLR, pp. 699–707.

Alwateer, Majed, Loke, Seng W., 2019. A two-layered task servicing model for drone services: Overview and preliminary results. In: 2019 IEEE International Conference on Pervasive Computing and Communications Workshops. PerCom Workshops, IEEE, pp. 387–390.

Barmpounakis, Emmanouil, Geroliminis, Nikolas, 2020. On the new era of urban traffic monitoring with massive drone data: The pNEUMA large-scale field experiment. Transp. Res. C Emerg. Technol. 111, 50–71.

Barmpounakis, E., Geroliminis, N., 2022. pNEUMA dataset. https://open-traffic.epfl.ch/. (Accessed 16 November 2022).

Bartolini, Novella, Coletta, Andrea, Maselli, Gaia, 2019. On task assignment for early target inspection in squads of aerial drones. In: 2019 IEEE 39th International Conference on Distributed Computing Systems. ICDCS, IEEE, pp. 2123–2133.

Bartolini, Novella, Coletta, Andrea, Maselli, Gaia, et al., 2020. A multi-trip task assignment for early target inspection in squads of aerial drones. IEEE Trans. Mob. Comput. 20 (11), 3099–3116.

Bupe, Paul, Haddad, Rami, Rios-Gutierrez, Fernando, 2015. Relief and emergency communication network based on an autonomous decentralized UAV clustering network. In: SoutheastCon 2015. IEEE, pp. 1–8.

Butilă, Eugen Valentin, Boboc, Răzvan Gabriel, 2022. Urban traffic monitoring and analysis using unmanned aerial vehicles (UAVs): A systematic literature review. Remote Sens. 14 (3), 620.

Castells-Graells, David, Salahub, Christopher, Pournaras, Evangelos, 2020. On cycling risk and discomfort: urban safety mapping and bike route recommendations. Computing 102 (5), 1259–1274.

Chen, Jie, Qing, Xianguo, Ye, Fang, Xiao, Kai, You, Kai, Sun, Qian, 2022. Consensus-based bundle algorithm with local replanning for heterogeneous multi-UAV system in the time-sensitive and dynamic environment. J. Supercomput. 78 (2), 1712–1740.

Chen, Yongbo, Yang, Di, Yu, Jianqiao, 2018. Multi-UAV task assignment with parameter and time-sensitive uncertainties using modified two-part wolf pack search algorithm. IEEE Trans. Aerosp. Electron. Syst. 54 (6), 2853–2872.

Company DJI, 2022. DJI Phantom 4 Pro. https://www.dji.com/uk/phantom-4-pro/info#specs. (Accessed 16 November 2022).

Elloumi, Mouna, Dhaou, Riadh, Escrig, Benoit, Idoudi, Hanen, Saidane, Leila Azouz, 2018. Monitoring road traffic with a UAV-based system. In: 2018 IEEE Wireless Communications and Networking Conference. WCNC, IEEE, pp. 1–6.

Fanitabasi, Farzam, Gaere, Edward, Pournaras, Evangelos, 2020. A self-integration testbed for decentralized socio-technical systems. Future Gener. Comput. Syst. 113, 541–555.

Fu, Zhangjie, Mao, Yuanhang, He, Daojing, Yu, Jingnan, Xie, Guowu, 2019. Secure multi-UAV collaborative task allocation. IEEE Access 7, 35579–35587.

Gao, Yang, Zhang, Yingzhou, Zhu, Shurong, Sun, Yi, 2018. Multi-UAV task allocation based on improved algorithm of multi-objective particle swarm optimization. In: 2018 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery. CyberC, IEEE, pp. 443–4437.

Gerostathopoulos, Ilias, Pournaras, Evangelos, 2019. TRAPPed in traffic? A self-adaptive framework for decentralized traffic optimization. In: 2019 IEEE/ACM 14th International Symposium on Software Engineering for Adaptive and Self-Managing Systems. SEAMS, IEEE, pp. 32–38.

Inoue, Yoshio, 2020. Satellite-and drone-based remote sensing of crops and soils for smart farming–a review. Soil Sci. Plant Nutr. 66 (6), 798–810.

Kizilateş, Gözde, Nuriyeva, Fidan, 2013. On the nearest neighbor algorithms for the traveling salesman problem. In: Advances in Computational Science, Engineering and Information Technology. Springer, pp. 111–118.

Mac, Thi Thoa, Copot, Cosmin, De Keyser, Robin, Ionescu, Clara M, 2018. The development of an autonomous navigation system with optimal control of an UAV in partly unknown indoor environment. Mechatronics 49, 187–196.

Motlagh, Naser Hossein, Bagaa, Miloud, Taleb, Tarik, 2019. Energy and delay aware task assignment mechanism for UAV-based IoT platform. IEEE Internet Things J. 6 (4), 6523–6536.

Nik, Vahid M., Moazami, Amin, 2021. Using collective intelligence to enhance demand flexibility and climate resilience in urban areas. Appl. Energy 281, 116106.

Nikolic, Jovan, Pournaras, Evangelos, 2019. Structural self-adaptation for decentralized pervasive intelligence. In: 2019 22nd Euromicro Conference on Digital System Design. DSD, IEEE, pp. 562–571.

Omidshafiei, Shayegan, Agha-Mohammadi, Ali-Akbar, Amato, Christopher, Liu, Shih-Yuan, How, Jonathan P, Vian, John, 2017. Decentralized control of multi-robot partially observable Markov decision processes using belief space macro-actions. Int. J. Robot. Res. 36 (2), 231–258.

Outay, Fatma, Mengash, Hanan Abdullah, Adnan, Muhammad, 2020. Applications of unmanned aerial vehicle (UAV) in road safety, traffic and highway infrastructure management: Recent advances and challenges. Transp. Res. Part a Policy Pract. 141, 116–129.

Poudel, Sabitri, Moh, Sangman, 2022. Task assignment algorithms for unmanned aerial vehicle networks: A comprehensive survey. Veh. Commun. 100469.

Pournaras, Evangelos, 2020. Collective learning: A 10-year odyssey to human-centered distributed intelligence. In: 2020 IEEE International Conference on Autonomic Computing and Self-Organizing Systems. ACSOS, IEEE, pp. 205–214.

Pournaras, Evangelos, Pilgerstorfer, Peter, Asikis, Thomas, 2018. Decentralized collective learning for self-managed sharing economies. ACM Trans. Auton. Adapt. Syst. (TAAS) 13 (2), 1–33.

Pournaras, Evangelos, Yadhunathan, Srivatsan, Diaconescu, Ada, 2020. Holarchic structures for decentralized deep learning: a performance analysis. Cluster Comput. 23 (1), 219–240.

Qin, Chuhao, Candan, Fethi, Mihaylova, Lyudmila, Pournaras, Evangelos, 2022. 3, 2, 1, Drones go! A testbed to take off UAV swarm intelligence for distributed sensing. In: Proceedings of the 2022 UK Workshop on Computational Intelligence. Springer Nature.

Qin, Chuhao, Pournaras, Evangelos, 2022. EPOS-based plans for drones. http://dx.doi.org/10.6084/m9.figshare.21432804.v10.

Stolaroff, Joshuah K., Samaras, Constantine, O'Neill, Emma R., Lubers, Alia, Mitchell, Alexandra S., Ceperley, Daniel, 2018. Energy use and life cycle greenhouse gas emissions of drones for commercial package delivery. Nature Commun. 9 (1).

Wierzbicki, Damian, 2018. Multi-camera imaging system for UAV photogrammetry. Sensors 18 (8), 2433.

Wu, Di, Arkhipov, Dmitri I., Kim, Minyoung, Talcott, Carolyn L., Regan, Amelia C., McCann, Julie A., Venkatasubramanian, Nalini, 2016. ADDSEN: Adaptive data processing and dissemination for drone swarms in urban sensing. IEEE Trans. Comput. 66 (2), 183–198.

Wu, Xueli, Yin, Yanan, Xu, Lei, Wu, Xiaojing, Meng, Fanhua, Zhen, Ran, 2021. Multi-UAV task allocation based on improved genetic algorithm. IEEE Access 9, 100369–100379.

Yanmaz, Evşen, Yahyanejad, Saeed, Rinner, Bernhard, Hellwagner, Hermann, Bettstetter, Christian, 2018. Drone networks: Communications, coordination, and sensing. Ad Hoc Netw. 68, 1–15.

Zhou, Zhenyu, Feng, Junhao, Gu, Bo, Ai, Bo, Mumtaz, Shahid, Rodriguez, Jonathan, Guizani, Mohsen, 2018. When mobile crowd sensing meets UAV: Energy-efficient task assignment and route planning. IEEE Trans. Commun. 66 (11), 5526–5538.

Zhou, Yongkun, Rao, Bin, Wang, Wei, 2020. UAV swarm intelligence: Recent advances and future trends. IEEE Access 8, 183856–183878.