Engineering 56 (2026) 173–185

Contents lists available at ScienceDirect

Engineering

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

Research
6G: From Theory to Practice—Article

Robot Subset Selection-Based Multi-User Edge Computing for Swarm Lifetime Maximization with Correlated Data Sources

Siqi Zhang, Yi Ma *, Rahim Tafazolli

Institute for Communication Systems, University of the Surrey, Guildford GU2 7XH, UK

ARTICLE INFO

Article history:
Received 23 December 2024
Revised 10 October 2025
Accepted 21 October 2025
Available online 27 October 2025

Keywords:
Edge computing
Resource allocation
Graph theory
Swarm network
Task offloading

ABSTRACT

In this paper, we investigate the problem of maximizing the lifetime of robot swarms in wireless networks utilizing a multi-user edge computing system. Robots offload their computational tasks to an edge server, and our objective is to efficiently exploit the correlation between distributed data sources to extend the operational lifetime of the swarm. The optimization problem is approached by selecting appropriate subsets of robots to transmit their sensed data to the edge server. Information theory principles are used to justify the grouping of robots in the swarm network, with data correlation among distributed robot subsets modeled as an undirected graph. We introduce a periodic subset selection problem, along with related and more relaxed formulations such as a graph partitioning problem and a subgraph-level vertex selection problem, to address the swarm lifetime maximization challenge. For additive white Gaussian noise channels, we analyze the theoretical upper bound of the swarm lifetime and propose several algorithms—including the least-degree iterative partitioning algorithm and final vertex search algorithm—to approach this bound. Additionally, we consider the impact of channel diversity on subset selection in flat-fading channels and adapt the algorithm to account for variations in the base station's channel estimation capabilities. Comprehensive simulation experiments are conducted to evaluate the effectiveness of the proposed methods. Results show that the algorithms achieve a swarm lifetime up to 650% longer than that of benchmark approaches.

© 2025 THE AUTHORS. Published by Elsevier LTD on behalf of Chinese Academy of Engineering and Higher Education Press Limited Company. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

1. Introduction

By robot swarm, we refer to the definition provided by the US Army Academy (Section 2 in Ref. [1]):

Definition 1 (robot swarm). A robot swarm is a group of (three or more) robots that perform jobs cooperatively while receiving limited or no control from human operators. The term "cooperatively" is defined in the way that involves mutual assistance in working toward a common goal. It does not necessarily imply or require communication or explicit coordination between entities [2A].

This advanced technology has wide-ranging applications in disaster recovery, defense, reconnaissance, infrastructure inspection, mapping, agriculture, food management, and space exploration [3]. As a promising use case for six-generation (6G) networks [4], robot swarms are expected to leverage emerging wireless technologies such as ultra-reliable low-latency communication (URLLC), massive machine-type communication (mMTC), and integrated sensing and communication (ISAC). The enhanced capabilities of 6G—including dense device connectivity, real-time data exchange, and artificial intelligence (AI)-powered edge computing—are crucial for enabling the precise coordination, adaptive behavior, and rapid decision-making required by swarm robotics. With these advancements, robot swarms could achieve unprecedented levels of scalability, intelligence, and operational efficiency, opening new opportunities across diverse industries.

However, realizing these capabilities in practice remains highly challenging, particularly in scenarios where swarm robots operate under tight energy and latency constraints while generating overlapping and redundant sensory data. These challenges are particularly critical in real-world deployments. For example, during post-disaster response operations, both aerial and ground robots may explore collapsed buildings, often sensing overlapping regions and producing redundant data. Similarly, in large-scale agricultural monitoring, multiple robots may observe similar crop health patterns, resulting in high spatial correlation among their sensory

* Corresponding author.
E-mail address: y.ma@surrey.ac.uk (Y. Ma).

https://doi.org/10.1016/j.eng.2025.10.015
2095-8099/© 2025 THE AUTHORS. Published by Elsevier LTD on behalf of Chinese Academy of Engineering and Higher Education Press Limited Company. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

inputs. Transmitting all collected data in such settings leads to excessive communication overhead and accelerated battery depletion, ultimately reducing the operational lifetime of the swarm.

In this work, we consider a robot swarm in which a set of mobile robots cooperatively performs a single job, as defined in Definition 1. Each robot can sense its environment and, when necessary, send the data to a computing server through a connected access point (AP). The server processes the received data and sends the computed results back to the robots, which then execute the job accordingly. The lifetime of the swarm is a vital metric for assessing its performance. It indicates the maximum number of tasks that the robot swarm can accomplish from initiation to completion [2]. Each computing outcome represents a job requiring collaborative execution by all robots. When the battery of a robot is depleted, rendering it inoperable, the entire swarm either halts or suspends operations until re-initiated. Consequently, the swarm lifetime is constrained by the operational lifetime of the robot with the shortest battery life, which acts as a bottleneck. Therefore, it becomes crucial to focus on optimization approaches that maximize the lifetime of the shortest-operating robot, given its heightened impact compared to other metrics [5-8].

This work falls within the scope of mobile edge computing (MEC) in the wireless communication domain. The motivation for offloading computing tasks to an edge server is twofold [9]: ① Data processing tasks are often energy-intensive and thus not ideal for battery-powered mobile robots; and ② edge servers typically contain powerful computing units (such as graphics processing units (GPUs), central processing units (CPUs), accelerated processing units (APUs), and neural processing units (NPUs)) that can provide parallel and low-latency computation. Most existing works on computation offloading focus on resource allocation in single-user equivalent computing scenarios (e.g., Refs. [6,8,10,11]). In these cases, even though multi-user communication is considered for reliability, data from different devices are processed individually at the edge server [2].

Recently, multi-user edge computing has gained growing attention [12], where data from multiple devices must be jointly processed to obtain a computing outcome. Researchers have explored resource allocation in such scenarios to ensure efficient data transmission to edge servers for collaborative processing [13–15]. Most of this work focuses on minimizing latency or energy consumption using centralized optimization, heuristic scheduling, or learning-based methods. Deep reinforcement learning (DRL) approaches such as actor–critic frameworks, including twin delayed deep deterministic (TD3) policy gradient strategy [16], have been applied to dynamic offloading and network resource allocation. These methods typically model individual user agents in stochastic environments and learn adaptive policies over time. However, most of these studies do not explicitly consider data correlation, which is central to collaborative robot swarms with shared information structures. Specifically, prior work either assumes data are independent and identically distributed (IID) or ignores opportunities to optimize performance by exploiting non-IID data correlations [17].

The development of goal- or task-oriented communication [18] offers a new perspective on multi-user edge computing by incorporating data correlation. This paradigm focuses on effectiveness by defining a clear communication objective. In goal-oriented communication, a transmitter sends only the data strictly relevant to achieving a specific goal, thereby optimizing performance under application constraints [19]. This aligns well with the requirements of multi-user edge computing.

The core theory underlying multi-user goal-oriented communication builds on distributed source coding [19] and multivariate information bottlenecks [20]. For example, Shao et al. [21] proposed a learning-based communication scheme for task-oriented, multi-device cooperative edge inference. By extending the distributed information bottleneck framework and developing a selective retransmission mechanism, they optimized feature extraction and encoding, effectively reducing redundancy and communication overhead. In this context, recent work has focused primarily on source-coding techniques, which aim to eliminate redundancy while retaining data relevant to the computational output [18,22]. While such approaches are applicable to multi-user edge computing, they typically require semantic encoders on the transmitter side [23], along with considerable storage, computational power, and stable energy supply—resources that may exceed the capacity of low-cost devices [24,25].

These limitations, particularly in handling data correlation and the high computation demands of source coding, motivated us to introduce the concept of a robot subset for robot swarms (defined in Section 2.2). The proposed subset comprises robots that possess sufficiently informative data, enabling accurate computation while reducing redundancy. By selectively choosing this subset, the number of robots involved in communication is significantly reduced. This approach not only mitigates existing limitations but also improves both computational and communication efficiency.

Consequently, this work falls under multi-user edge computing, with the goal of maximizing robot swarm lifetime by exploiting data source correlation among mobile robots. The optimization problem is addressed by selecting appropriate robot subsets to transmit their sensed data to the edge server. The main contributions of this study include:

(1) A new perspective on multi-user edge computing;

(2) A novel robot-subset model using an undirected graph to describe data source correlation between spatially distributed robots;

(3) A new cost function formulation for swarm lifetime optimization;

(4) Theoretical analysis of swarm lifetime bounds for various cases;

(5) Introduction of a periodic subset selection problem to address the swarm lifetime maximization challenge;

(6) Relaxation of the subset selection problem into graph partitioning and subgraph-level vertex selection problems, along with corresponding algorithms;

(7) Extension of swarm lifetime maximization under fading channels, accounting for diverse channel estimation requirements.

While the proposed approach builds on classical tools like graph partitioning and subset selection, its novelty lies in integrating these techniques into a unified swarm-level optimization strategy that explicitly models inter-robot data correlation. Prior work on energy-efficient task offloading often treats robots as independent agents, ignoring mutual information in their sensed data. In contrast, we model data correlation using an undirected graph and base the selection of informative robot subsets on information-theoretic sufficiency conditions.

Importantly, our work is not a general-purpose graph partitioning or offloading framework. It is specifically designed to solve the lifetime maximization problem in energy-constrained robot swarms. The optimization goal—maximizing the number of cooperative tasks completed before any robot depletes its energy—leads to a unique formulation that jointly considers communication constraints, energy consumption, and task accuracy. All algorithmic relaxations and graph-based models in this study are driven by this lifetime objective, distinguishing our method from conventional approaches that prioritize computational efficiency or load balancing alone.

The proposed subset selection framework addresses several technical challenges essential to the realization of 6G-enabled swarm intelligence. Specifically, it supports URLLC and mMTC by

reducing uplink congestion and eliminating redundant transmissions. Its ability to model data correlation using undirected graphs and enable distributed, resource-aware decision-making aligns with ISAC principles. These features make our approach a lightweight and scalable solution for real-world 6G applications such as autonomous surveillance, disaster response, and smart logistics.

To further contextualize our contributions, we highlight practical 6G-enabled scenarios where the proposed framework can be readily applied. For instance, in post-disaster recovery missions involving aerial and ground robot swarms, selectively offloading data from the most informative robots under tight energy and latency constraints is critical. Similarly, in smart agriculture or environmental monitoring, the method supports long-term, energy-aware sensing across large deployments with spatially correlated data. These scenarios benefit from reduced communication overhead, improved energy balance, and faster coordination, which are key goals of 6G edge intelligence.

2. System description, modeling, and problem formulation

2.1. System description

Let us consider a robotic swarm network consisting of M robots. These robots work collaboratively to accomplish a single job, assisted by an edge server that provides computational support and coordination. Each robot can perform environmental sensing and, if required, transmit its sensed data to the edge server via a connected AP. The server processes the received data and returns the computed outcomes to the robots, which then use these results to execute the task. This scenario falls within the domain of multi-user edge computing, where multiple devices rely on an edge server for computational support.

The swarm is described as a set $\mathscr{A} = \{A_m | m = 1, ..., M\}$, where A_m denotes the mth robot. Mathematically, a robot can be represented as a tuple, $A \triangleq (S, \mathscr{E}, \varepsilon)$. S denotes the sensed data, which can be considered as a random variable with its realization defined as s; $\mathscr{E}$ represents the remaining energy of the robot; ε stands for the energy consumption per job (for the sake of concise presentation, ε is assumed to be a constant). The sensed data of A_m is denoted as s_m, which can be assumed as a realization of random variable S_m.

Provided that all robots send their data to the edge server, the multi-user edge computing process can be described by

$$\theta = f(s_m | m = 1, ..., M) \tag{1}$$

where $f(\cdot)$ is the edge computing function; θ denotes the computing outcome, which can be considered the realization of the random variable Θ. Eq. (1) implies that when all the sensed data from the robots are available, the uncertainty associated with the computing outcome, as quantified by the conditional entropy (H), is reduced to zero.

$$H(\Theta | \mathscr{S}) = 0 \tag{2}$$

where $\mathscr{S} \triangleq \{S_1, ..., S_M\}$. This shows that the computing outcome Θ is completely determined by the knowledge of all the sensed data $S_1, S_2, ..., S_M$ from the M robots and that there is no remaining uncertainty about the outcome once all the data are known. Additionally, given the practicality, we assume that the data sources in the swarm are not independent [21]; hence, we derive the subsequent equation as follows:

$$\sum_{m=1}^{M} H(S_m) > H(\mathscr{S}) \geq H(\Theta) \tag{3}$$

This aligns with general assumptions [21]. From an information-theoretic perspective, the principles of distributed source coding [19,26] and goal-oriented semantic coding [27] can be employed to eliminate redundancies. However, in practical applications, the robot's battery, computational resources, and storage capacity constraints (for coding) make it challenging to effectively remove all redundancies. This motivated us to consider subset-level multi-user edge computing.

2.2. Data correlation and subset model

Eq. (3) implies that when the correlation (corresponding to Θ) between different data sources is sufficiently large, the edge server may not require the entire data source to form the computing outcome, that is,

$$H(\Theta | \mathscr{S} \setminus S_m) = 0, \exists A_m \in \mathscr{S} \tag{4}$$

This insight motivated us to propose the concept of a robot subset for multi-user edge computing.

Definition 2 (robot subset). Consider a subset of robots within the swarm, denoted by $\Omega_w \subset \mathscr{A}$, where w is the index of the subset. The edge computing outcome corresponding to the data of subset Ω_w is $\widehat{\theta}_{\Omega_w}$ and described by

$$\widehat{\theta}_{\Omega_w} = f(s_m | \forall A_m \in \Omega_w) \tag{5}$$

and Ω_w satisfies

$$H(\Theta | \mathscr{S}_{\Omega_w}) = 0 \tag{6}$$

$$I(\mathscr{S}_{\Omega_w} \setminus S_m; \Theta) < H(\Theta), \forall A_m \in \Omega_w \tag{7}$$

where $I(\cdot;\cdot)$ denotes the mutual information and $\mathscr{S}_{\Omega_w}$ denotes the random variables set which corresponds to the sensed data of Ω_w. This statement implies that even if the edge server computes $\widehat{\theta}_{\Omega_w}$ using only the data corresponding to the subset of robots Ω_w, it can still accurately obtain the computing outcome θ. In other words, the edge server can achieve the same level of accuracy in estimating θ using $\widehat{\theta}_{\Omega_w}$ as it would have achieved if it had access to the sensed data from all the robots. Mathematically, this can be represented as $\theta = \widehat{\theta}_{\Omega_w}$.

Definition 2 implies that the entire set can be divided into several robot subsets, each having the same information value. Then, for each computing task, only one of these subsets is required to send its sensed data to the edge server. This results in significant communication energy savings and an extended swarm lifetime [2].

In systems where robots move randomly or irregularly, selecting subsets that consistently provide sufficient information for edge computing often introduces redundancy. This is because such subsets must account for rare or worst-case scenarios to ensure reliability. To address this, we adopt a simplified assumption that each robot operates within a fixed activity area. This reduces redundancy without relying on complex techniques such as semantic encoders, which may be impractical for resource-constrained robots.

Although this assumption does not fully reflect real-world scenarios—where robots may move dynamically because of task changes or environmental conditions—it allows us to focus on the impact of data correlation in subset selection. In many practical applications, such as agricultural monitoring, warehouse logistics, or environmental sensing, robots are often assigned predefined or constrained regions to improve efficiency and avoid collisions. Therefore, the fixed-area assumption remains a reasonable approximation.

Nevertheless, in highly dynamic settings where correlation patterns change over time, the subset selection process may need periodic updates. Future work can explore mobility-aware or adaptive correlation estimation methods to improve robustness under such conditions.

2.3. Communication model

For every task, the edge server chooses one of the robot subsets (denoted by Ω), requesting all robots in this subset to send their data to the edge through orthogonal subchannels. For $A_m \in \Omega$, the channel capacity between the robot and the AP is denoted as $C(h_m, p_m)$, where h_m is the channel and p_m is the transmit power. The set of selected robots must then fulfill the following criteria:

Criterion 1: $\forall A_m \in \Omega$, their required transmission rate (denoted by R_m) must fulfill

$$R_m \leq C(h_m, p_m), R_m = \frac{D_m}{T_c} \tag{8}$$

where D_m is the input number of bits for the task of the mth robot, and T_c denotes the maximum tolerable communication latency for a task. If the criteria listed in Eq. (8) are not satisfied, the robot (A_m) or robot set (Ω) must not be chosen because of a capacity outage.

Criterion 2: All robots must possess sufficient energy to execute an allocated job.

After computing Eq. (5), the computing outcome is broadcast to all robots. The broadcast is assumed to be reliable and efficient. This is a commonly used assumption in the literature (e.g., Refs. [8,28,29]) that allows us to focus on the key problem of interest.

2.4. Problem formulation

In this study, the objective is to maximize the number of tasks that the swarm can successfully complete, referred to as the swarm lifetime, before any robot depletes its battery and halts its entire operation. A key challenge lies in minimizing energy consumption while maintaining computational correctness. Because the data collected by nearby robots are often highly correlated, it is possible to reconstruct the task outcome using only a subset of the robot data. Therefore, our design freedom lies in selecting a subset of robots to transmit data to the edge server at each task iteration.

Considering the life cycle of the ith task conducted by the swarm, the robots have the following energy consumption model [2]:

$$\mathscr{E}_m(i) = \mathscr{E}_m(i-1) - p_m(i)T_c - \epsilon - \varepsilon, \forall A_m \in \Omega \tag{9}$$

$$\mathscr{E}_m(i) = \mathscr{E}_m(i-1) - \varepsilon, \forall A_m \notin \Omega \tag{10}$$

where $p_m(i)$ denotes the transmission power of the mth robot for the ith task and $\mathscr{E}_m(i)$ is the remaining energy of mth robot after the ith task is conducted. ϵ is the energy consumption for communication-related computations, such as source and channel coding. Note that we have the following conditions.

$$\mathscr{E}_m(i) \geq 0 \tag{11}$$

Otherwise, the ith task does not occur and the $(i-1)$th task is the final task assigned to the swarm network.

Define $\Phi(i) \triangleq (\Omega(1),\ \Omega(2), \ldots,\ \Omega(i))$. We further define $\Gamma(\Phi(i))$ as the robot selection strategy which yields

$$\Gamma(\Phi(i)) = \begin{cases} i, & (11) \text{ is true} \\ 0, & (11) \text{ is false} \end{cases} \tag{12}$$

According to the definition of the swarm lifetime, the swarm lifetime maximization problem has the following objective function:

$$\Phi^{\star}(i) = \text{argmax}_{\Phi(i)} \Gamma(\Phi(i)) \tag{13}$$

where $\Phi^{\star}$ denotes the optimal robot selection strategy. The implication of Eq. (13) is that swarm lifetime maximization is equivalent to the convergence problem of an iterative algorithm. To derive the universal energy consumption model in Eq. (14) rigorously, we begin with the task-level energy update expressions defined in Eqs. (9) and (10). For each robot A_m, the cumulative communication energy consumed after ith task is denoted as $\zeta_m(i)$, where $\zeta_m(i)$ accumulates all per-task communication energy terms $(p_m T_c + \epsilon)$ whenever A_m is selected. Therefore, subtracting this cumulative consumption from the initial energy $\mathscr{E}_m(0)$ yields Eq. (14), which provides the residual energy:

$$\mathscr{E}_m(i) = \mathscr{E}_m(0) - \zeta_m(i) - i\varepsilon \tag{14}$$

and $\zeta_m(i)$ is defined by

$$\zeta_m(i) \triangleq \sum_{i'=1}^{i} \sum_{j \in \Psi_m} \left(p_m(i')T_c + \epsilon\right)\delta(i'-j) \tag{15}$$

where $i' = 1, \ldots, i$; Ψ_m is the set that collects all iteration indexes when the mth robot is chosen to send its data; j denotes the task index belonging to the set Ψ_m assigned to the mth robot; and $\delta(\cdot)$ is the Dirac Delta function [30]. This formulation explicitly captures the cumulative energy usage, which is essential for swarm lifetime optimization.

Define $e_m(i) \triangleq \mathscr{E}_m(0) - \zeta_m(i)$. A robot with a smaller $e_m(i)$ has a shorter lifetime and the swarm lifetime is determined by the robot with the shortest lifetime. Therefore, the problem of swarm lifetime maximization is equivalent to maximizing the minimum of $e_m(i), \forall m$, that is,

$$\Phi^{\star}(i) = \underset{\Phi(i)}{\text{argmax}} \left(\min(e_1(i),\ e_2(i), \ldots,\ e_M(i))\right) \tag{16}$$

Thus, the swarm lifetime maximization problem can be formulated as a subset selection problem.

Moreover, the maximization–minimization (MM) optimization problem in Eq. (16) is known to be non-deterministic polynomial-time hard (NP-hard) (Refs. [4,31]). However, it is possible to find good suboptimal solutions based on the relaxed conditions. The objective function in Eq. (16) is grounded in the fundamental operational model of swarm robotics, in which the swarm is considered inactive when an individual robot fails because of energy depletion. Thus, maximizing the minimum residual energy directly translates into prolonging the time before the first robot fails, thereby maximizing the swarm lifetime. This MM formulation is widely used in energy-constrained distributed systems and aligns naturally with the mission-critical nature of swarm tasks.

3. Graph model for swarm lifetime maximization in additive white Gaussian noise channels

3.1. Analysis of the theoretical swarm lifetime upper bound

In the context of the additive white Gaussian noise (AWGN) channel, the channel quality measure (α_m) is not robot dependent. Instead, it can simply be set as $\alpha_m = 1, \forall m$, and the channel capacity is not a criterion for selecting a robot subset. In this study, we assume that all the robots in the swarm have the same type and hardware configuration. As such, they are equipped with batteries of equal capacity, and thus start with the same initial energy. $\mathscr{E}(0)$ and p have their indexes omitted because they are identical for all robots. In practice, the energy cost of executing a job may vary depending on its type, for example sensing or physical actuation. To reflect this, we denoted the energy consumption of the job

corresponding to the ith task as ε_i. However, in this study, we adopted a unified average energy consumption ε for all tasks and robots. This is justified by our assumption of a centralized MEC-based task assignment; the MEC server schedules heterogeneous tasks in a manner that balances the overall workload and energy consumption across the swarm. As a result, each robot tends to receive a statistically similar task profile over time, allowing us to approximate the $\varepsilon_i \approx \varepsilon$ for all i. This abstraction enables a tractable analysis while preserving its practical relevance.

We defined $\varphi_m(i)$ to represent the total number of times the mth robot was selected after the completion of the ith task, that is,

$$\varphi_m(i) \triangleq \sum_{i'=1}^{i} \sum_{j \in \Psi_m} \delta(i' - j) \tag{17}$$

Thus, the term of $\zeta_m(i)$ in Eq. (15) can be rewritten as

$$\zeta_m(i) = \varphi_m(i)(pT_c + \epsilon) \tag{18}$$

Owing to the constraints listed in Eq. (11), we substitute $\zeta_m(i)$ into Eq. (14) and rewrite it as a linear function of the selection frequency under a uniform transmission power (only in AWGN cases) and coding energy:

$$\mathscr{E}(0) - \varphi_m(i)(pT_c + \epsilon) - i\varepsilon \geq 0, \forall m \tag{19}$$

In the problem formulation, we define the swarm lifetime as the number of tasks that the swarm can successfully complete. Here, we denote the swarm lifetime by $i^\star$, which also represents the maximum number of tasks the swarm can perform. Therefore, the relationship between $i^\star$ and $\varphi_{m,\forall m}$ can be formulated as

$$i^\star \leq \frac{\mathscr{E}(0) - \min\limits_{\varphi_m(i)} \left(\max\left\{ \varphi_m\left(i^\star\right), \forall m \right\} \right)(pT_c + \epsilon)}{\varepsilon} \tag{20}$$

However, it is challenging to determine the theoretical upper bound of the swarm lifetime based on Eq. (20), because it is difficult to ascertain the optimum $\varphi_{m,\forall m}$.

The optimization problem outlined in Eq. (16) was conceptualized as a subset-selection issue. It is assumed that the subset selection process, which spans the entire swarm lifetime, operates periodically. Here, the period length is denoted by γ (≥ 1), whereas the maximum selection frequency for any robot within a single period is represented by μ ($\leq \gamma$). Given these parameters, Eq. (20) can be reformulated as follows:

$$i^\star \leq \frac{\mathscr{E}(0) - \min\limits_{\mu,\gamma} \left(\mu\lceil \frac{i^\star}{\gamma} \rceil \right)(pT_c + \epsilon)}{\varepsilon} \tag{21}$$

where $\lceil \cdot \rceil$ denotes integer ceiling. This indicates the importance of $\min\limits_{\mu,\gamma} \left(\mu\lceil \frac{i^\star}{\gamma} \rceil \right)$ for the upper bound of swarm lifetime. For the term $\mu\lceil \frac{i^\star}{\gamma} \rceil$, we have

$$\mu\lceil \frac{i^\star}{\gamma} \rceil \geq \lceil \frac{i^\star \mu}{\gamma} \rceil \tag{22}$$

Considering the case that swarm lifetime is large enough, we can assume that

$$\min_{\mu,\gamma} \left(\mu\lceil \frac{i^\star}{\gamma} \rceil \right) \approx \min_{\mu,\gamma} \left(\lceil \frac{i^\star \mu}{\gamma} \rceil \right) \tag{23}$$

Thus, the swarm lifetime upper bound in Eq. (21) can be transformed into

$$i^\star \lesssim \frac{\mathscr{E}(0) - \min\limits_{\mu,\gamma} \left(\lceil \frac{i^\star \mu}{\gamma} \rceil \right)(pT_c + \epsilon)}{\varepsilon} \tag{24}$$

This provides the foundation for addressing lifetime maximization in a generic case, for which we model the subset selection problem as a period subset selection problem.

Definition 3 (period subset selection problem). Our period subset selection problem is stated by selecting several subsets (one vertex can be selected more than once) to form period that maximize the ratio of the period length ($\bar{\mu}$) to the selection frequency ($\bar{\gamma}$) of the robot that is chosen most often within a single period.

By applying the period vertex selection concept to the analysis of the swarm lifetime upper bound, the following result can be obtained:

Theorem 1. Given $\mathscr{E}_m(0), \forall m$ are identical (condition (c1)), the swarm lifetime is upper-bounded by

$$i^\star \lesssim \frac{\mathscr{E}(0) - \lceil i^\star / \left(\frac{\bar{\mu}}{\bar{\gamma}} \right) \rceil (pT_c + \epsilon)}{\varepsilon} \tag{25}$$

Proof. The validation of this theorem is directly based on Eq. (24) and Definition 3. To maintain brevity and conserve space, a detailed explanation is not provided here.

Theorem 1 shows the theoretical swarm lifetime upper bound for a generic robot swarm system under the assumption that the corresponding $\bar{\mu}$ and $\bar{\gamma}$ are fully accessed. Moreover, the period subset selection problem is well known as NP-hard (Ref. [32]), and an optimum solution cannot be claimed. However, it is possible to find good suboptimal solutions based on relaxed conditions. We tackle this problem step-by-step, from a special case to a generic case.

3.2. Graph theory-based methods for swarm lifetime maximization

In the analysis, we define a total of W subsets. First, the theoretical swarm lifetime upper bounds in the two special cases and their corresponding subset selection methods are considered.

Lemma 1. Suppose: $\mathscr{E}_m(0), \forall m$ are identical (condition (c1)); the common part $\Omega_c \triangleq \Omega_1 \cap \Omega_2 \cap \ldots \cap \Omega_W \neq \varnothing$ (condition (c2)). The swarm lifetime is given by

$$i^\star = \left\lfloor \frac{\mathscr{E}(0)}{pT_c + \epsilon + \varepsilon} \right\rfloor \tag{26}$$

where $\lfloor \cdot \rfloor$ stands for the integer floor.

Proof. Given condition (c1), robot subsets are the only factors influencing lifetime maximization. Given condition (c2), regardless of which subset was chosen, the common part (Ω_c) is always chosen. Therefore, robots within Ω_c have the shortest lifetimes. The term of $\zeta(i)$ [2]

$$\zeta(i) = i(pT_c + \epsilon) \tag{27}$$

and their energy consumption model Eq. (14) becomes

$$\mathscr{E}(i) = \mathscr{E}(0) - i(pT_c + \epsilon + \varepsilon) \tag{28}$$

Given $\mathscr{E}\left(i^\star\right) \geq 0$ solving this inequality leads to Eq. (26).

Lemma 2. Given the condition (c1) and $\Omega_{w_1} \cap \Omega_{w_2} = \varnothing$, $\forall w_1 \neq w_2$, where w_1 and w_2 are two different subset index (condition (c3)), the swarm lifetime is upper-bounded by

$$i^\star \leq \frac{\mathscr{E}(0) - \lceil i^\star / W \rceil (pT_c + \epsilon)}{\varepsilon} \tag{29}$$

Proof. Under conditions (c1) and (c3), the best strategy for subset selection is round robin because all the subsets are equal. In this case, the energy consumption model generally remains the same as in Eq. (14), but the term $\zeta(i)$ becomes a robot index-independent term

$$\zeta(i) \leq \lceil \frac{i}{W} \rceil (pT_c + \epsilon) \tag{30}$$

This expression describes the subsets chosen most frequently. In other words, the robots within these subsets have the shortest lifetimes. Due to $\mathscr{E}\left(i^\star\right) \geq 0$, that is,

$$\mathscr{E}(0) - \lceil i^\star / W \rceil (pT_c + \epsilon) - i^\star \varepsilon \geq 0 \tag{31}$$

We immediately have inequality Eq. (29). The upper bound in Eq. (29) does not provide a closed-form solution for i. Nevertheless, it is easy to determine the i through a line search [2].

Eqs. (26) and (29) provide the theoretical upper bounds of the swarm lifetime, based on the underlying structure of the robot subset overlap. Specifically, Eq. (26) corresponds to the optimistic case, where task loads are perfectly balanced across all robots through maximal reuse of correlated subsets—that is, a complete graph scenario. Conversely, Eq. (29) represents a pessimistic configuration, where no subsets share any robots, resulting in a null graph and requiring each robot to operate independently. In real-world swarm systems, actual performance typically lies between these two extremes. Highly correlated sensing environments, such as structured indoor factories or aerial swarms with overlapping fields of view, tend to approximate the complete graph condition. Meanwhile, systems with widely dispersed or isolated agents approach the null graph case. These two bounds, therefore, define the operational envelope for swarm lifetime under specific communication and energy constraints.

The theoretical upper bounds of the swarm lifetime and the corresponding subset selection strategies, as presented in Lemmas 1 and 2, apply only to two specific instances of robot subsets. Nevertheless, these cases provide a foundation for analyzing the theoretical upper bound of swarm lifetime in more general scenarios, where graphs are used to represent robot subsets.

Definition 4 (graphical model of subsets). Define an undirected graph $G(V, E)$, where V denotes the set of vertices and E represents the set of edges. Each vertex corresponds to a subset. The edge between two vertices reflects the presence of common part between two corresponding subsets (i.e., the edge between the vertices (subset) w_1 and w_2 exists when $\Omega_{w_1} \cap \Omega_{w_2} \neq \varnothing$). A degree matrix $\boldsymbol{D}$ is formed to record the number of edges linked to vertices (see the definition of $\boldsymbol{D}$ in the graph theory [33]). A single graph can be partitioned into numerous subgraphs (if the vertex number is no smaller than two). These subgraphs can be either connected or disconnected, it means that each subgraph can either share nodes with other subgraphs or they can be entirely separate. Each graph can be conceptualized as an aggregation of several complete graphs (a graph with a solitary vertex qualifies as a complete graph. Similarly, a graph with two interconnected vertices also constitutes a complete graph), whether connected or disconnected [34].

According to Definition 4, condition (c2) can be interpreted as an extreme case within a complete graph $G\left(W, \frac{W(W-1)}{2}\right)$, a concept referred to as an ex-complete graph.

Definition 5 (ex-complete graph). Consider a complete graph (shown in Fig. 1) in which all the vertices (subsets) share at least one identical robot.

Condition (c3) can be embodied by a null graph $G(W, 0)$ (shown in Fig. 1).

Theorem 2. $\bar{W}_e$ is defined as the maximum number of ex-complete subgraphs that the graph can be partitioned into. Given condition (c1), the graph can be divided into $\bar{W}_e$ ex-complete graphs (condition (c4)); a specific vertex can surely be identified in each graph, and these vertices exhibit pairwise orthogonality (condition (c5)). We call this vertex representative vertex (R-vertex). The swarm lifetime is upper-bounded by

$$i^\star \leq \frac{\mathscr{E}(0) - \lceil i^\star / \bar{W}_e \rceil (pT_c + \epsilon)}{\varepsilon} \tag{32}$$

Proof. Subset selection is applied at the subgraph level in a round-robin manner. For each subgraph, only one of the R-vertices is chosen for data transmission. Since the R-vertices from different subgraphs are not directly connected, this case is equivalent to that in Lemma 2. Following the discussion in the proof of Lemma 2, the swarm lifetime is determined as Ref. [32]. The reason for selecting R-vertices is to ensure that there is no overlap between the selected subsets; otherwise, some robots would be chosen multiple times within a single round, which would shorten the swarm lifetime—as previously discussed in Lemma 2. Given $\bar{W}$ as the number of subgraphs (i.e., ex-complete graphs), the right-hand side of Eq. (32) is maximized. Theorem 2 is thus proved [2].

Although Theorem 2 corresponds only to a case where the graph is composed of multiple ex-complete graphs, it highlights the importance of identifying the maximum number of orthogonal vertices for solving the swarm lifetime maximization problem. In Theorem 2, each robot is selected at most once in a single period. This technical insight motivates the relaxation of the period subset selection problem into a graph partitioning problem. Specifically, the original subset selection formulation (as defined in Eq. (16)) explicitly searches for all feasible combinations of robots, which become computationally intractable as the swarm size increases.

To address this challenge, we leverage the insight that subsets of robots naturally correspond to the vertices of a correlation graph, with edges indicating shared robots between subsets. By reformulating the subset selection problem into a graph partitioning problem, we shift the computational burden from exhaustive subset enumeration to efficient vertex partitioning into subgraphs with minimal overlap. This graph-based formulation significantly simplifies the search space and enables the use of practical heuristics for swarm management. Therefore, transitioning from subset selection to graph partitioning is not arbitrary—it is well justified by both the need for computational tractability and deployment feasibility.

Definition 6 (graph-partitioning problem). Our graph partitioning problem is stated by partitioning an undirected graph into a maximal number of subgraphs ($\bar{W}$) [2].

Once the period subset selection problem is relaxed into a graph partitioning problem and an optimal partitioning algorithm is applied, the maximum achievable swarm lifetime—as indicated in Eq. (32)—can be obtained. However, graph partitioning is generally an NP-hard problem, and obtaining the optimal solution is challenging [32]. To address this, we propose a least-degree iterative partitioning (LDIP) algorithm, which offers a good suboptimal solution. Fig. 2 illustrates the concept of the LDIP algorithm. Fundamentally, the algorithm consists of three steps [2]:

Step 1 (least degree finding): Sort the diagonal of the degree matrix $\boldsymbol{D}$ in ascending order; the first diagonal entry of $\boldsymbol{D}$ (denoted by $D_{(0,0)}$) has the lowest degree.

Step 2 (subgraph forming): The vertex (subset) corresponding to $D_{(0,0)}$ is chosen as the R-vertex of the subgraph under construction. All the vertices that are directly connected to the R-vertex are included in this subgraph.

Step 3 (degree matrix updating): Update the degree matrix $\boldsymbol{D}$ by eliminating all vertices selected in step 2. Repeat step 1 until $\boldsymbol{D}$ reduces to a scalar equal to zero, that is, no vertices remain.

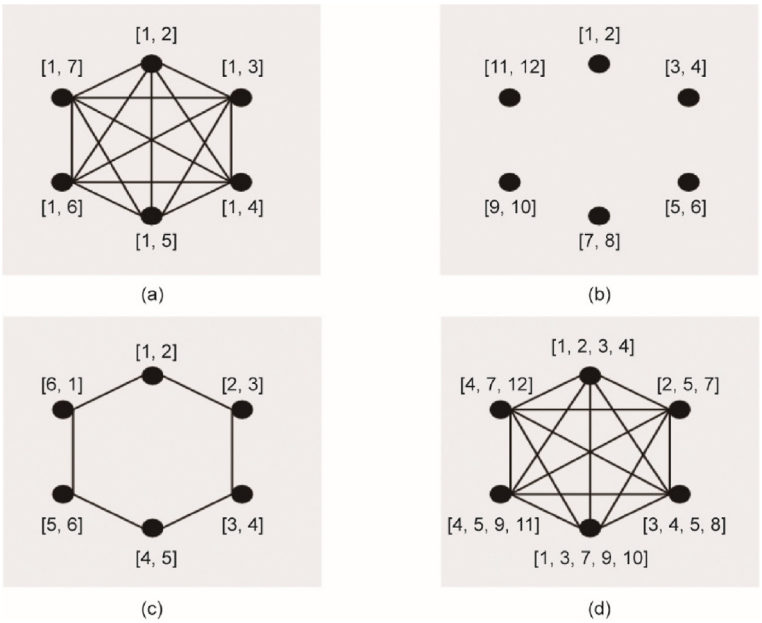

Fig. 1. Examples of (a) ex-complete graph, (b) null graph, (c) cycle graph, and (d) general complete graph. [·] denotes the robot indexes of the corresponding subset.

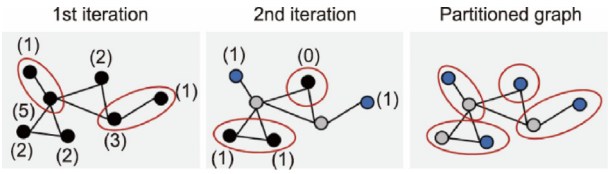

Fig. 2. Concept of the LDIP algorithm for graph partitioning. Red ellipses are subgraphs; blue-faced circles are R-vertices of their subgraphs.

The suboptimality of LDIP lies in the fact that multiple vertices may share the least degree at each iteration. Finding the optimal solution would require examining all such cases in every iteration, leading to exponential search complexity. LDIP instead selects one of the least-degree vertices at random in each iteration, offering linear search complexity at the cost of optimality [2].

Now consider a generic correlation graph. Besides the performance degradation introduced by the LDIP algorithm itself, relaxing the period subset selection problem to a graph partitioning problem also causes degradation because conditions (c4) and (c5) are not generally satisfied. The core idea of the graph partitioning problem is to maximize the length of each period while ensuring that no robot appears more than once per period. This strategy aims to increase the ratio between the period length and the selection frequency of the most frequently chosen robot. However, this does not always guarantee the maximum possible ratio.

Lemma 3. Suppose condition (c1); $(\Omega_{w_1} \cap \Omega_{w_2}) \cap (\Omega_{w_1} \cap \Omega_{w_3}) = \varnothing, \forall w_1 \neq w_2 \neq w_3$ (condition (c6)), where w_3 denotes a robot subset index; for any given robot subset, there exist two other robot subsets each sharing at least one robot with the given subset (condition (c7)). The swarm lifetime is given by:

$$i^\star \leq \frac{\mathscr{E}(0) - 2\lceil i^\star / W \rceil (pT_c + \epsilon)}{\varepsilon} \tag{33}$$

Proof. Based on conditions (c6) and (c7), it can be determined that each robot will participate in a maximum of two subsets. Some robots will be part of two subsets, whereas others will be part of only one subset. The optimal strategy for subset selection embodies a round-robin approach. This is primarily due to the round-robin scheduling policy, which ensures that each robot participates in at most two of the Z tasks, where $Z = W$. Employing any alternative vertex selection strategy, the robot that has been chosen the greatest number of times within W selections is chosen no fewer than twice. Therefore, the term $\zeta_m(i)$

$$\zeta(i) \leq 2\lceil \frac{i}{W} \rceil (pT_c + \epsilon) \tag{34}$$

Following the concept that shown in Lemma 1, we can easily obtain

$$\mathscr{E}(0) - 2\lceil i^\star / W \rceil (pT_c + \epsilon) - i^\star \varepsilon \geq 0 \tag{35}$$

Lemma 3 is therefore proved.

According to Definition 3, conditions (c6) and (c7) can be described as a cycle graph $G(W, W)$ (Fig. 1).

Consider a cycle graph with W subsets (vertices) that can be partitioned into maximum $\lfloor \frac{W}{2} \rfloor$ complete graphs. Thus, by following the graph-partitioning problem, we obtain the corresponding swarm lifetime:

$$i^\star \leq \frac{\mathscr{E}(0) - \lceil i^\star / (\lfloor \frac{W}{2} \rfloor) \rceil (pT_c + \epsilon)}{\varepsilon} \tag{36}$$

Comparing Eq. (36) with Eq. (33), we can easily determine that the performance loss is caused by the graph-partitioning problem for the cycle graph. In Eq. (36), only the $\lfloor \frac{W}{2} \rfloor$ subsets are involved in subset selection, but the number corresponding to Eq. (33) is W. This implies that Eq. (33) may involve more robots, while guaranteeing the same maximum selected number (with Eq. (36)) for a single robot. This insight can also be applied to a more generic complete graph.

Theorem 3. Under the stipulations of conditions (c1); M robots can be depicted as a complete graph comprising W subset (condition (c8)); $(\Omega_{w_1} \cap \Omega_{w_2}) \cap (\Omega_{w_1} \cap \Omega_{w_3}) = \varnothing, \forall w_1 \neq w_2 \neq w_3$ robots (condition (c9)). The swarm lifetime is subjected to the subsequent inequality:

$$i^\star \lesssim \frac{\mathscr{E}(0) - \lceil i^\star / (\frac{W}{2}) \rceil (pT_c + \epsilon)}{\varepsilon} \tag{37}$$

Proof. Considering conditions (c8) and (c9), similar to the reasoning applied in Lemma 3, it is determined that a standalone robot will be utilized no more than twice in the offloading of Z tasks. The validation of this lemma is directly based on Lemma 3. To maintain brevity and conserve space, a detailed explanation is not provided here.

Consider a complete graph with conditions (c8) and (c9), such a graph can be partitioned into a maximum of only one subgraph. Again, this exemplifies the disadvantage that the graph-partitioning problem involves only a small number of vertices. This motivated us to further update the graph-partitioning problem to a subgraph-level vertex-searching problem (Definition 7). Although the graph partitioning formulation significantly reduces complexity compared to direct subset selection, its practical implementation still poses challenges due to potential redundancy and limited utilization of robot subsets within each subgraph. To further refine this approach, we introduce the subgraph-level vertex selection problem, specifically designed to efficiently select a small number of representative vertices (subsets) within each subgraph. This additional relaxation step effectively balances subset diversity and computational simplicity, ensuring that each selected vertex represents a minimally overlapping configuration. Thus, this second relaxation, from graph partitioning to subgraph-level vertex selection, further enhances computational feasibility and practical deployment without significantly compromising performance.

Definition 7 (subgraph-level vertex-searching problem). Our subgraph-level vertices searching problem is stated by: select no less one vertex from each subgraph (obtained from solving graph-partitioning problem) to maximize $\sum_{\bar{w}=1}^{\bar{W}} \frac{\widetilde{W}_{\bar{w}}}{\widetilde{M}_{\bar{w}}}$ ($\widetilde{W}_{\bar{w}}$ denote the selected vertices number in the $\bar{w}$th subgraph ($\bar{w} = 1, 2, ..., \bar{W}$), and $\widetilde{M}_{\bar{w}}$ denotes the number of occurrences of the robot with the highest number of occurrences within the selected vertices) with any two selected vertices from different subgraph are orthogonal (condition (c10)). We call this vertex as final vertex (F-vertex).

Acquiring $\bar{\gamma}$ and $\bar{\delta}$ for an entire graph, as previously described, is inherently complex. Nonetheless, through graph partitioning, the intricacies of the subgraph structure are somewhat attenuated, thereby enhancing the feasibility of pinpointing vertices in the subgraph that resonate with the optimum solution of the period subset selection problem.

Theorem 4. Given conditions (c1); the corresponding graph can be divided into $\bar{W}$ subgraphs through solving graph partitioning problem (condition (c10)). Through solving the subgraph-level vertices searching problem with the optimum solution, the swarm lifetime is given by

$$i^{\star} \lesssim \frac{\mathscr{E}(0) - \left\lceil i^{\star} \Big/ \left(\max\left\{ \sum_{\bar{w}=1}^{\bar{W}} \frac{\widetilde{W}_{\bar{w}}}{\widetilde{M}_{\bar{w}}} \right\} \right) \right\rceil (pT_c + \epsilon)}{\varepsilon} \tag{38}$$

Proof. In Eq. (38), $\max\left\{ \sum_{\bar{w}=1}^{\bar{W}} \frac{\widetilde{W}_{\bar{w}}}{\widetilde{M}_{\bar{w}}} \right\}$ denotes the maximum frequency of the robot chosen most often within a single period by solving the subgraph-level vertex-searching problem. Therefore, adhering to the principles of Theorem 1, Theorem 2, and Theorem 3, the proof of Theorem 4 is established.

To approach the swarm lifetime shown in Theorem 4, we propose the following F-vertex search algorithm that consists of three steps:

Step 1 (initialization): Mark every vertex from the set of R-vertices as F-vertices, and initialize a counter $\bar{w}$ to 1.

Step 2 (orthogonal vertex searching): Start with the $\bar{w}$th subgraph. We look for vertices that do not share robots with the F-vertices from other subgraphs, forming a set of these vertices. If this set has more than two vertices, then proceed to step 3. If not, increment w by 1 and repeat this step.

Step 3 (F-vertex selection):

- A vertex is randomly selected from the formed set (in the $\bar{w}$th subgraph) and marked as a temporary F-vertex.
- If another vertex exists that shares robots with the temporary F-vertex and existing F-vertices, but the shared robots are different, the temporary F-vertex is upgraded to the F-vertex, and this vertex is removed from the formed vertex set.
- Select a vertex from the remaining vertex set that shares different robots with all the existing F-vertices. If such a vertex is found, it is marked as an F-vertex and removed from the formed vertex set. This process continues until no more vertices are selected from the set.
- If there were two F-vertices in $\bar{w}$th subgraph, one was deleted.
- Increment $\bar{w}$ by 1 and return to step 2, repeating the process until $\bar{w} = \bar{W}$.

The potential sub-optimality of the algorithm for subgraph-level vertex searching may arise from the fact that multiple vertices can satisfy the selection criteria at each iteration. The algorithm processes each subgraph individually and sequentially, selecting F-vertices independently within each subgraph. Although this approach is computationally efficient, it may not yield a globally optimal solution due to the lack of coordination across subgraphs. Consequently, some vertex selections may be locally valid but suboptimal from a global perspective. Achieving a truly optimal solution would require evaluating all possible vertex combinations across subgraphs, which incurs exponential complexity due to the combinatorial nature of the problem.

While our heuristic algorithm avoids this intractability by selecting one vertex per iteration, it is important to assess the computational complexity more formally to understand the algorithm's scalability in large-scale swarm systems. We therefore provide the computational complexities of both the LDIP and F-vertex selection algorithms below. The LDIP algorithm iteratively selects the vertex with the lowest degree and forms a subgraph by grouping it with its neighbors. In each iteration, identifying the vertex with the smallest degree and updating the degree matrix requires at most time complexity $O(W)$ operations. Because each vertex is selected only once throughout the process, the overall time complexity is $O\left(W^2\right)$. This quadratic complexity is acceptable even in large-scale systems and offers a practical alternative to exact graph partitioning, which is known to be computationally intractable.

The F-vertex selection algorithm further refines the process by checking orthogonality among candidate vertices across subgraphs. Let K represent the maximum number of robots in any subset. In the worst case, the algorithm performs pairwise comparisons across all subgraphs, resulting in a time complexity of $O\left(W^2 \cdot K\right)$. As K is typically much smaller than W in practice, the overall complexity remains tractable.

We acknowledge that several classical subset selection and graph partitioning methods (e.g., greedy heuristics, k-means clustering, integer linear programming (ILP) formulations, or genetic algorithms) have been widely studied in related domains. However, our problem formulation incorporates not only data correlation but also energy depletion dynamics and communication reuse—constraints that these methods are not directly designed to handle. Moreover, methods like ILP suffer from poor scalability, and evolutionary algorithms require significant tuning and long convergence times, making them unsuitable for dynamic swarm operations. In contrast, the LDIP and final-vertex selection

algorithms are tailored for fast, structure-aware partitioning in real-time swarm control. That said, we consider it a valuable direction to adapt and benchmark these classical methods against our approach in future work.

4. Swarm lifetime maximization in independent flat-fading channels with limited channel estimation

4.1. Analysis of the optimization in flat-fading channels

Consider the channel quality measure (α_m) as a random variable that is independent (but not necessarily identically distributed) with respect to the robot index and the task index. Consequently, the transmit power (p_m) also varies independently. In this case, the energy consumption model for the mth robot reverts to the original form in Eq. (14), and the term $\zeta_m(i)$ can be rewritten as [2]:

$$\begin{aligned}\zeta_m(i) &= T_c \sum_{i'=0}^{i} \sum_{j\in\Psi_m} \left(p_m\left(i'\right)\right) + |\Psi_m|\epsilon \\ &= \omega_m(i) T_c \left(\frac{1}{\omega_m(i)} \sum_{i'=0}^{i} \sum_{j\in\Psi_m} \left(p_m\left(i'\right)\right)\right) + \omega_m(i)\epsilon \end{aligned} \tag{40}$$

where $\omega_m(i)$ signifies the number of times the mth robot is selected until the ith task is completed, and it is easy to obtain the following swarm lifetime:

$$i^{\star} \lesssim \frac{\mathscr{E}(0) - \lceil i^{\star} \Big/ \left(\frac{\bar{p}}{\bar{\gamma}}\right) \rceil (\max\{\bar{p}_m, \forall A_m\} T_c + \epsilon)}{\varepsilon} \tag{41}$$

where $\bar{p}_m$ denotes the average transmission power of the mth robot when it is selected for data transmission, and $\bar{p}_m$ is generally robot-dependent. Therefore, involving channel diversity in the subset selection for each computation offloading process in the fading channel cases, the swarm lifetime is upper bounded by

$$i^{\star} \lesssim \max_{p_m(i')} \frac{\min\left\{\mathscr{E}(0) - \lceil i^{\star} \Big/ \left(\frac{\bar{p}}{\bar{\gamma}}\right) \rceil (\bar{p}_m T_c + \epsilon), \forall A_m\right\}}{\varepsilon} \tag{42}$$

Based on Eq. (42), it is evident that instantaneous channel quality significantly impacts the subset selection strategy in fading channels. The key challenge is identifying a subset for each task transmission that maximizes the overall swarm lifetime. Achieving a locally optimal subset (for a single transmission) does not necessarily guarantee a globally optimal result across the swarm's operational lifetime.

To address this complexity, the concept of energy balance [35,36] is often employed, particularly using the MM principle, which aims to maximize the minimum residual energy after each offloading operation [4,6,37]. By applying this principle, the original optimization in Eq. (16) is reformulated into a short-term objective function:

$$\Omega_s^{\star}(i) = \underset{\Omega_s(i)}{\operatorname{argmax}} \left(\min(e_1(i),\ e_2(i), \ldots,\ e_M(i))\right), \forall \Omega_s(i) \tag{43}$$

where $\Omega_s^{\star}(i)$ denotes the optimal short-term subset selection strategy for the ith task, and $\Omega_s(i)$ represents the feasible robot subset selection strategy for the ith task.

This optimization problem can be solved using a brute-force search that searches all subsets, and the subset selection strategy obtained from Eq. (43) can offer significant improvements in the swarm lifetime performance [2]. This enhanced performance was primarily attributed to substantial channel diversity. Therefore, this method has a fatal issue: The base station (edge server side) must perform channel estimation for all robots. In scenarios with a vast number of robots, executing this is virtually infeasible for the base station.

4.2. Optimization in flat-fading channels with limited channel estimation

Such issues caused by huge requirements for channel estimation can be effectively addressed by integrating the MM concept with the method we proposed for the AWGN channel. The subset associated with an F-vertex is referred to as an optional candidate. Denote $\mathscr{W}$ to be the set of all optional candidates. Involving the concept of MM into the period vertices selection problem, the optimization function can be further transferred into Eq. (44):

$$\Omega_s^{\star}(i) = \underset{\Omega_s(i)}{\operatorname{argmax}} \left(\min(e_1(i),\ e_2(i), \ldots, e_M(i))\right), \forall \Omega_s(i) \in \mathscr{W} \tag{44}$$

For the ith task, we selected one subset from all optional subset candidates that can maximize the minimum remaining energy of the swarm network after the ith task. Employing this method typically results in a substantial reduction in the number of robots required for channel estimation. However, this reduction in channel diversity tends to decrease swarm lifetime.

Furthermore, when delving into an alternative scenario where the quantity of robots within an optional candidate set surpasses the maximal capacity of the base station, affiliated with the edge server, for channel estimation, the MM principle remains applicable. This concept can guide the selection of multiple subsets from the optional candidate set, contingent on the current remaining energy, to execute the channel estimation. Let V_C denote the number of robots for which the base station can provide channel estimation. The robots, corresponding to the subset which has a higher lowest remaining, have higher priority to be given the channel estimation service. Let $Q_{V_C}(\cdot)$ represent the optional candidate reorganization function. We adhere to the principle of continually choosing the subset with the highest remaining minimum power (avoiding repetitions) until the number of associated robots reaches a maximum value not surpassing V_C. Therefore, the reformed optional candidate set ($\bar{\mathscr{W}}$) for ith task can be described by

$$\bar{\mathscr{W}}(i) = Q_{V_C}(\mathscr{W}), \bar{\mathscr{W}}(i) \subset \mathscr{W} \tag{45}$$

Thus, the optimization problem can be further reformulated as

$$\Omega_s^{\star}(i) = \underset{\Omega_s(i)}{\operatorname{argmax}} \left(\min(e_1(i),\ e_2(i), \ldots,\ e_M(i))\right), \forall \Omega_s(i) \in \bar{\mathscr{W}}(i) \tag{46}$$

Both Eqs. (44) and (46) can be solved by brute-force search over the reduced subset set, yielding low computational overhead.

In this work, we focus on AWGN and IID. Rayleigh fading channels to validate the core behavior of our subset selection framework. However, the method does not critically rely on these assumptions. Optimization is performed over $\bar{\mathscr{W}}$, which includes only subsets that satisfy current latency and channel constraints. Thus, the algorithm does not require static channels or global channel state information (CSI).

This design ensures robustness against moderate channel variation, interference, and mobility-induced fading, provided that viable subsets remain. In dynamic environments, the data correlation graph can be periodically updated, and channel feasibility re-evaluated with minimal overhead. Since subset selection is performed per task, the algorithm adapts naturally to changing channel states without relying on long-term predictions or full CSI. These properties collectively ensure practical applicability under real-world, time-varying wireless conditions.

5. Experiment design and simulation results

5.1. Model for generating robot subsets

Computer simulations were conducted to evaluate the proposed subset selection approach in terms of robot swarm lifetime. A major challenge in these simulations lies in forming robot subsets during Monte Carlo trials, as no well-established stochastic or deterministic subset models currently exist in the literature. In our previous work [2], we proposed a location-based model for simulation. However, this model assumes complete orthogonality between different areas (i.e., they are entirely independent and non-overlapping). In real-world scenarios, such assumptions may be overly idealized, as overlaps or interdependencies between regions are often present.

For instance, the model assumes that if robots A and B operate in the same area, they possess identical information. Consequently, any subset formed with robot A can also be formed with robot B. This simplification may not capture the true complexity of real environments. Thus, we sought to develop a more flexible, yet simple, stochastic method to generate robot subsets.

The objective is to randomly divide M robots into W subsets (each with up to K robots). The subset-forming algorithm includes the following three steps:

- **Step 1 (initialization):** Randomly select K robots. If any robot appears more than once in the group, remove duplicates so that each robot is unique within the group.
- **Step 2 (verification):** Check whether this group has already been created—either as a subset of a previous group or vice versa. If it has, discard it and generate a new group. Otherwise, add it to the collection of unique groups.
- **Step 3 (group creation):** Repeat the above steps until W unique robot groups are formed.

5.2. Parameter setting for simulations

In addition to the formation of robot subsets, several other parameters require appropriate settings. These parameters included $\mathscr{E}_m$, ε, ϵ, p_m. One of the challenging issues is determining the relationship between the energy costs for signal transmission, modulation, encoding, and robot task execution. This provided a unified energy consumption model for computer simulations. Unfortunately, we are not aware of any explicit descriptions of these relationships in the literature, despite an extensive survey. Therefore, in our simulations, we assumed that a robot's energy was divided into two fixed and separated parts: one for communication and the other for task execution. This division reflects a practical scenario in which robots consume different amounts of energy for data transmission and task execution, allowing us to model and optimize each component independently. To focus our study on the communication domain, it is assumed that the bottleneck of the robot swarm is communication energy. In other words, when a robot uses its communication energy, it counts as "dead" [2].

To simplify the computer simulations, we set $\mathscr{E}_m$ and p_m to be identical for all robots and omitted the subscript m. Moreover, we ignore the energy cost for modulation and coding (i.e., ϵ) because they are often negligibly small compared with the radio transmission energy. In our experiments, this was set to allow the robot to conduct 200 data transmissions in an AWGN channel with a signal-to-noise ratio (SNR) of 10 dB. This essentially sets the lifetime of the robot in the AWGN channel (i.e., $i = 200$) [2].

5.3. Simulation results and discussion

Our computer simulations primarily included two experiments concerning communication channels to be AWGN and IID Rayleigh. We consider the following two baselines:

- In conventional computation offloading, all the robots send their data to an edge server [8]. In the AWGN channel, this baseline approach straightforwardly provided a swarm lifetime of $i = 200$ according to our setting.
- MM algorithm for subset selection when only V_C robots are provided with a channel estimation service. This algorithm is an extension of the MM user selection approach in sensor networks [6], which is used for the network lifetime maximization problem; however, we apply it at the subset level. This MM principle is used to maintain the energy balance and enhance the sensor network lifetime with good performance. Strictly speaking, this is a novel algorithm, which we used as a baseline to demonstrate its advantages of the proposed algorithm.

The one with only R-vertices involved in the subset selection is called LDIP, and the one with all F-vertices involved in the subset selection is called LDIP-plus.

Experiment 1: The objective of this experiment is to evaluate the proposed approach in an AWGN channel environment. We simulated a robot swarm consisting of $M = 50$ robots, where each subset contained at most $K = 5, 10, 15, 20$ robots. Given the AWGN channel, we assume $V_C = M$. Figs. 3(a)–(c) illustrate the swarm lifetime as a function of the number of subsets (W) for $K = 5$, 10, and 15. The results demonstrated substantial gains (up to a 650% improvement) by leveraging data correlations. As W increases and K remains constant, the swarm lifetime tends to increase. However, this growth rate gradually decreased as W continued to increase. Beyond a certain threshold, further increments in W yield negligible improvements in the swarm lifetime. This trend is intuitive; larger W values enable a broader range of robot combinations, thereby enhancing robot diversity. Nevertheless, when most robots are maximally utilized, the benefits of the increased diversity plateau limit further gains. Additionally, as K increased, the swarm lifetime decreased across all subset selection approaches. This behavior was expected because larger K values require more robots for each offloading operation, resulting in higher resource demand.

Fig. 3(d) illustrates the swarm lifetime as a function of the number of subsets (W) when $K = 20$. The observed trend indicates that, except for the LDIP-plus algorithm (which achieves a gain of 5%–24%), the performance of the other algorithms is notably weak, closely resembling that of the traditional method. This poor performance is attributed to the high incidence of duplicate robots across different subsets when K is large, resulting in characteristics similar to those of a complete graph. From our analysis of the complete graphs, it is evident that the LDIP algorithm consistently selected the R-vertex for each data transmission. However, this approach is suboptimal for complete graphs and contributes to the observed performance degradation. In contrast, our enhanced LDIP-plus algorithm selects one of several F-vertices for each transmission. This strategy effectively adopts a cycle-graph-like selection pattern, which substantially improves swarm lifetime. It also ensures broader robot participation in data transmission and distributes the workload more evenly throughout the swarm lifetime. Fig. 4 presents the average swarm lifetime as a function of K in the AWGN channels with varying W values. This empirically shows that increasing the subset size K leads to a reduction in the swarm lifetime. This observation is consistent with the theoretical upper bound derived in Eq. (26), which indicates that the swarm lifetime is inversely proportional to the total number of robots involved across all subsets. As K increases, more robots are involved in each subset, leading to faster depletion of energy across the swarm. This in turn reduces the time until the first robot reaches its energy threshold, which defines the swarm lifetime. Therefore, both our theoretical analysis and simulation results confirm that K plays a critical role in lifetime optimization.

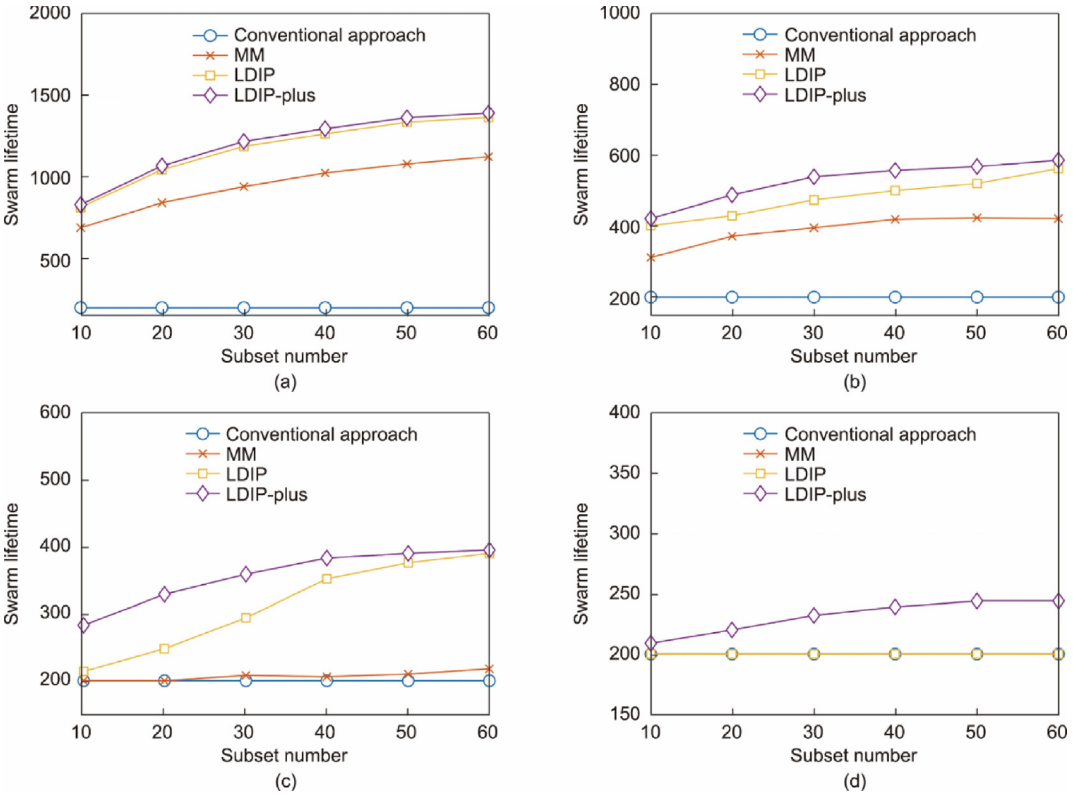

Fig. 3. Average swarm lifetime as a function of W in AWGN channels: (a) K = 5, (b) K = 10, (c) K = 15, and (d) K = 20.

Experiment 2: With the same system setup as in experiment 1, this experiment focused on the performance of the IID Rayleigh channels. The main difference between experiments 2 and 1 are ① power adaptation is needed in the fading channels (see Criterion 1); ② we simulate a robot swarm consisting of (M = 100) robots. This larger size was used in the Rayleigh fading experiments to evaluate performance under more dynamic channel conditions. For the AWGN scenarios in experiment 1, we used only M = 50 to focus on the energy trends under stable channels. Forty subsets of M = 40 subsets. Each robot subset had a maximum of (K = 5, 10, 15, and 20) robots. The number of robots for which the base station could provide channel estimation services was V_C = 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100.

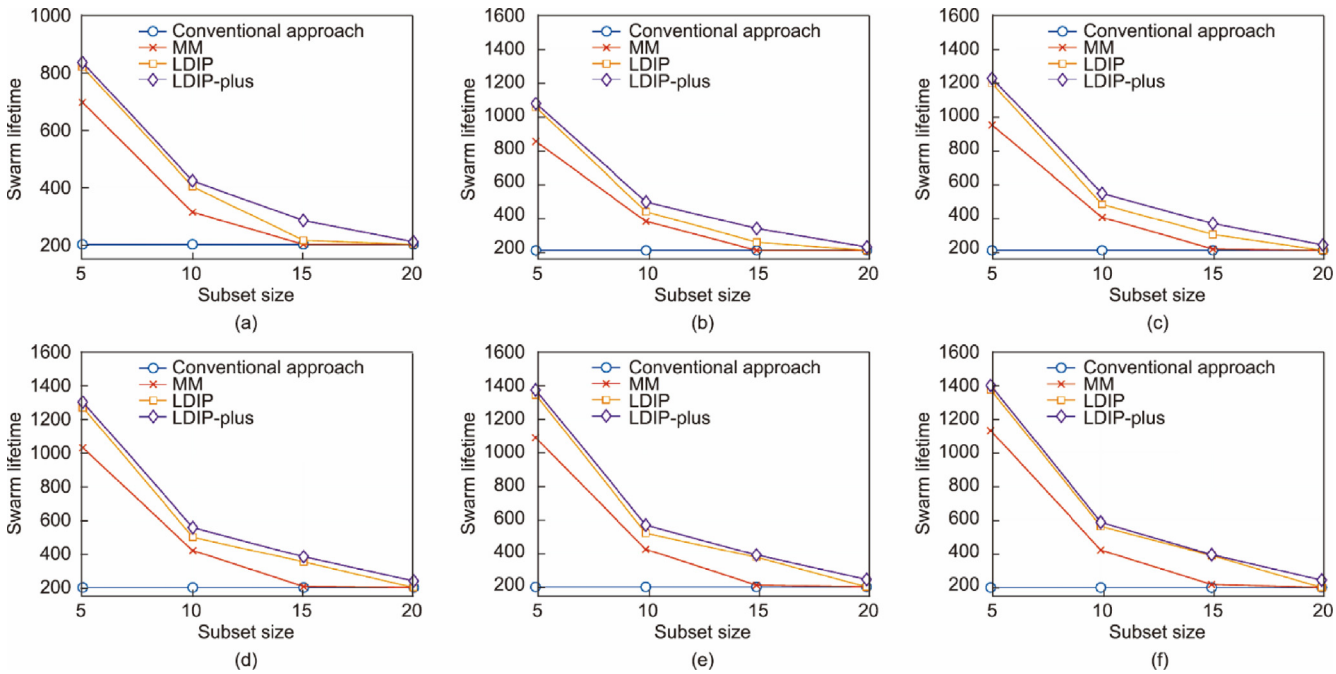

Fig. 4. Average swarm lifetime as a function of K in AWGN channels: (a) W =10, (b) W = 20, (c) W = 30, (d) W = 40, (e) W = 50, and (f) W = 60.

Fig. 5 illustrates the swarm lifetime when the base station provides channel estimation services to all robots. The MM algorithm achieves the longest swarm lifetime by fully leveraging channel diversity. Its subset selection strategy utilizes comprehensive CSI, enabling optimal resource allocation. In contrast, the LDIP and LDIP-plus algorithms, even with full access to CSI, utilize only partial data during subset selection. Although computationally simpler, these algorithms exhibit noticeably lower performance in this scenario.

To ensure a fair comparison, we evaluate all subset selection algorithms using the same amount of CSI, as shown in Fig. 6. The performance of the conventional method is not included in these figures, as it fails when the number of robots with channel estimation V_C differs from the total number of robots M. These results clearly highlight the advantages of the proposed LDIP and LDIP-plus algorithms, especially when V_C is small.

As V_C increases, enabling more diverse subsets and better exploitation of channel diversity, the swarm lifetime for all three algorithms (LDIP, LDIP-plus, and MM) increases, with each reaching a different performance plateau. Notably, when the number of robots with channel estimation is small, LDIP and LDIP-plus outperform the MM algorithm. However, as V_C continues to grow, this trend reverses: the MM algorithm eventually surpasses LDIP and LDIP-plus in performance. This is because LDIP and LDIP-plus reach their performance ceilings sooner, due to their limited subset diversity compared to MM.

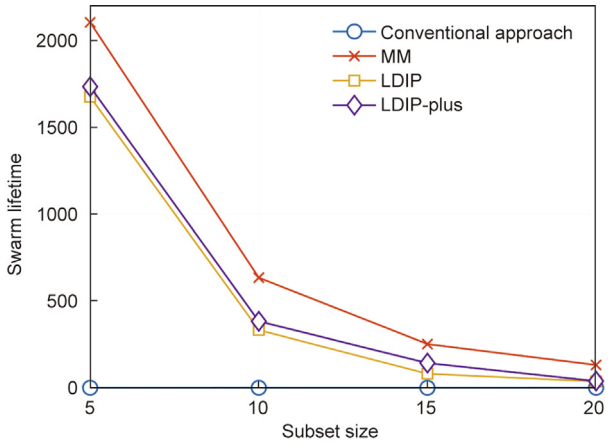

Fig. 5. Average swarm lifetime as a function of K in IID. Rayleigh fading channels when V_C =100.

6. Conclusions

This paper investigated the problem of maximizing the lifetime of robot swarms in multi-user edge computing systems by effectively leveraging the correlation among distributed data sources. We proposed a graph-based model for subset selection and developed corresponding optimization algorithms. Theoretical analysis and simulation results demonstrated that the proposed approach can significantly extend swarm lifetime, by up to 650%, in both AWGN and Rayleigh fading channels. The findings of this study offer new insights and practical solutions for resource optimization in multi-user edge computing, laying a strong foundation for future deployments of robot swarms in 6G-enabled environments.

CRediT authorship contribution statement

Siqi Zhang: Writing – review & editing, Writing – original draft, Software, Methodology, Investigation, Data curation, Conceptualization. **Yi Ma:** Supervision, Funding acquisition. **Rahim Tafazolli:** Funding acquisition.

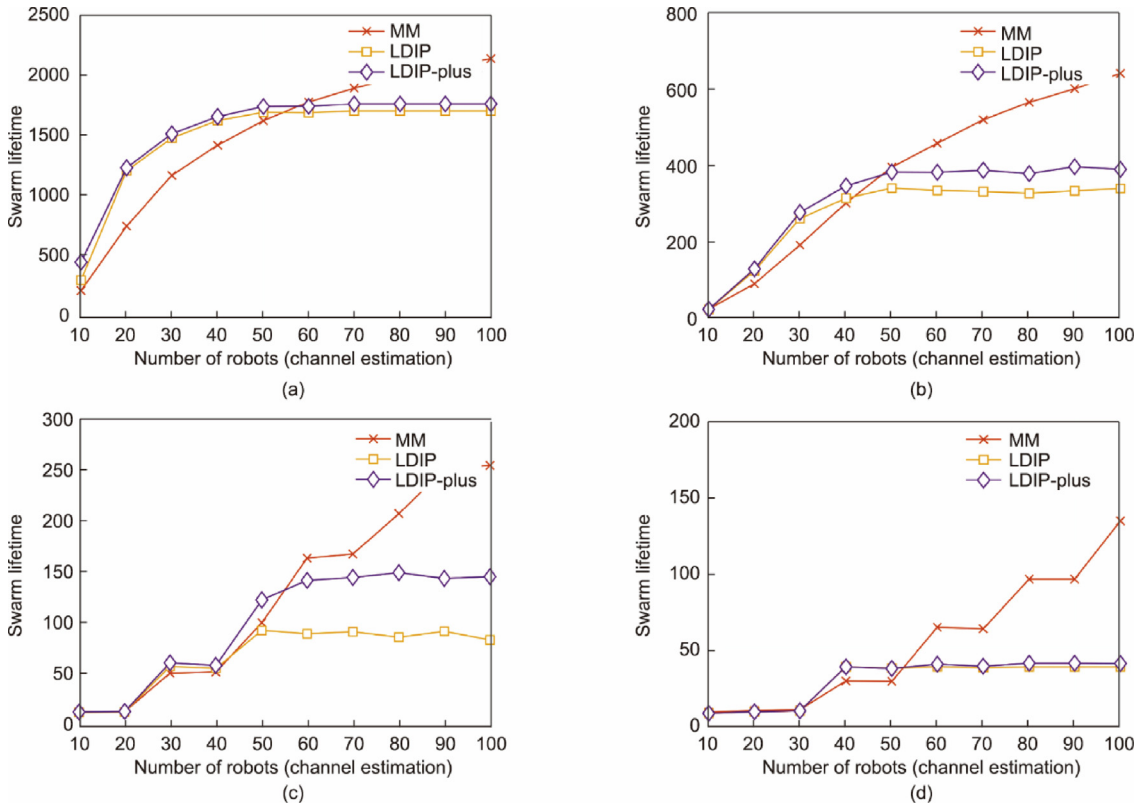

Fig. 6. Average swarm lifetime as a function of V in IID. Rayleigh fading channels: (a) K = 5, (b) K = 10, (c) K = 15, and (d) K = 20.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by the UK Department for Science, Innovation and Technology under the Future Open Network Research Challenge project Towards Ubiquitous 3D Open Resilient Network (TUDOR). The views expressed are those of the authors and do not necessarily represent the project.

References

[1] Arnold R, Carey K, Abruzzo B, Korpela C. What is a robot swarm: a definition for swarming robotics. In: Proceedings of IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference; 2019 Oct 10–12; New York City, NY, USA. New York City: IEEE; 2019. p. 0074–81.
[2] Zhang S, Yi N, Ma Y. Robot subset selection for swarm lifetime maximization in computation offloading with correlated data sources. In: Proceedings of IEEE International Conference on Communications; 2023 May 28–Jun 1; Rome, Italy. New York City: IEEE; 2023. p. 1–7.
[3] Hong Z, Huang H, Guo S, Chen W, Zheng Z. QoS-aware cooperative computation offloading for robot swarms in cloud robotics. IEEE Trans Veh Technol 2019;68(4):4027–41.
[4] Yuan S, Alam K, Han B, Krummacker D, Schotten HD. Exploring 6G potential for industrial digital twinning and swarm intelligence in obstacle-rich environments. 2024. arXiv:2406.19930.
[5] Yetgin H, Cheung KTK, El-Hajjar M, Hanzo LH. A survey of network lifetime maximization techniques in wireless sensor networks. IEEE Commun Surv Tutor 2017;19(2):828–54.
[6] Gupta S, Chakareski J. Lifetime maximization in mobile edge computing networks. IEEE Trans Veh Technol 2020;69(3):3310–21.
[7] Heidarpour AR, Heidarpour MR, Ardakani M, Tellambura C, Uysal M. Soft actor–critic-based computation offloading in multi-user MEC-enabled IoT—a lifetime maximization perspective. IEEE Internet Things J 2023;10(20):17571–84.
[8] Wang K, Fang F, Costa DB, Ding Z. Sub-channel scheduling, task assignment, and power allocation for OMA-based and NOMA-based MEC systems. IEEE Trans Commun 2021;69(4):2692–708.
[9] Qiu T, Chi J, Zhou X, Ning Z, Atiquzzaman M, Wu DO. Edge computing in industrial Internet of Things: architecture, advances and challenges. IEEE Commun Surv Tutor 2020;22(4):2462–88.
[10] Zhao P, Tian H, Qin C, Nie G. Energy-saving offloading by jointly allocating radio and computational resources for mobile edge computing. IEEE Access 2017;5:11255–68.
[11] Zhang S, Yi N, Ma Y. Correlation-based device energy-efficient dynamic multi-task offloading for mobile edge computing. In: Proceedings of IEEE 93rd Vehicular Technology Conference; 2021 Apr 25–28; Helsinki, Finland. New York City: IEEE; 2021. p. 1–5.
[12] Zhang S, Yi N, Ma Y. A survey of computation offloading with task types. IEEE Trans Intell Transp Syst 2024;25(8):8313–33.
[13] Lyu X, Tian H, Sengul C, Zhang P. Multiuser joint task offloading and resource optimization in proximate clouds. IEEE Trans Veh Technol 2017;66(4):3435–47.
[14] Liu L, Yuan X, Chen D, Zhang N, Sun H, Taherkordi A. Multi-user dynamic computation offloading and resource allocation in 5G MEC heterogeneous networks with static and dynamic subchannels. IEEE Trans Veh Technol 2024;72(11):14924–38.
[15] Cao K, Hu S, Shi Y, Colombo AW, Karnouskos S, Li X. A survey on edge and edge–cloud computing assisted cyber–physical systems. IEEE Trans Ind Informat 2021;17(11):7806–19.
[16] Mohajer A, Hajipour J, Leung VC. Dynamic offloading in mobile edge computing with traffic-aware network slicing and adaptive TD3 strategy. IEEE Commu Lett 2025;29(1):95–9.
[17] Bellal Z, Nour B, Mastorakis S. CoxNet: a computation reuse architecture at the edge. IEEE Trans Green Commun Netw 2021;5(2):765–77.
[18] Strinati EC, Barbarossa S. 6G networks: beyond Shannon towards semantic and goal-oriented communications. Comput Netw 2021;190:107930.
[19] Slepian D, Wolf J. Noiseless coding of correlated information sources. IEEE Trans Inf Theory 1973;19(4):471–80.
[20] Friedman N, Mosenzon O, Slonim N, Tishby N. Multivariate information bottleneck. 2013. arXiv:1301.2270.
[21] Shao J, Mao Y, Zhang J. Task-oriented communication for multidevice cooperative edge inference. IEEE Trans Wirel Commun 2023;22(1):73–87.
[22] Stavrou PA, Kountouris M. A rate distortion approach to goal-oriented communication. In: Proceedings of IEEE International Symposium on Information Theory; 2022 Jun 26–Jul 1; Espoo, Finland. New York City: IEEE; 2022. p. 590–5.
[23] Xu C, Mashhadi MB, Ma Y, Tafazolli R, Wang J. Generative semantic communications with foundation models: perception-error analysis and semantic-aware power allocation. 2024. arXiv:2411.04575.
[24] Farshbafan MK, Saad W, Debbah M. Curriculum learning for goal-oriented semantic communications with a common language. IEEE Trans Commun 2023;71(3):1430–46.
[25] Getu TM, Kaddoum G, Bennis M. Making sense of meaning: a survey on metrics for semantic and goal-oriented communication. 2023. arXiv:2303.10867.
[26] Wyner A, Ziv J. The rate-distortion function for source coding with side information at the decoder. IEEE Trans Inf Theory 1976;22(1):1–10.
[27] Li A, Wu S, Meng S, Zhang Q. Towards goal-oriented semantic communications: new metrics, open challenges, and future research directions. 2023. arXiv:2304.00848.
[28] Okegbile SD, Maharaj BT, Alfa AS. A multi-user tasks offloading scheme for integrated edge–fog–cloud computing environments. IEEE Trans Veh Technol 2022;71(7):7487–502.
[29] Liu CF, Bennis M, Debbah M, Poor HV. Dynamic task offloading and resource allocation for ultra-reliable low-latency edge computing. IEEE Trans Commun 2019;67(6):4132–50.
[30] Proakis JG, Salehi M. Digital communications. New York City: McGraw-Hill; 2001.
[31] Hou J, Ma Y, Yi N, Tafazolli R. Reduced-complexity coordinated beamforming for multicell downlink max–min SINR problem. IEEE Wirel Commun Lett 2014;3(4):353–6.
[32] Wu Z, Karimi HR, Dang C. An approximation algorithm for graph partitioning via deterministic annealing neural network. Neural Netw 2019;117:191–200.
[33] West DB. Introduction to graph theory. Upper Saddle River: Prentice Hall; 2001.
[34] Diestel R, Schrijver A, Seymour P. Graph theory. In: Proceedings of the Workshop on Graph Theory; 2010 Feb 21–27; Oberwolfach, Germany. Zurich: European Mathematical Society; 2010.
[35] Devika E, Saravanan A. A study of energy optimization in wireless sensor networks based on efficient protocols with algorithms. In: Proceedings of the 13th International Conference on Computing Communication and Networking Technologies; 2022 Oct 3–5; Kharagpur, India. New York City: IEEE; 2022.
[36] Du R, Gkatzikis L, Fischione C, Xiao M. On maximizing sensor network lifetime by energy balancing. IEEE Trans Control Netw Syst 2018;5(3):1206–18.
[37] Liu J, Xiong K, Ng DWK, Fan P, Zhong Z, Letaief KB. Max–min energy balance in wireless-powered hierarchical fog–cloud computing networks. IEEE Trans Wirel Commun 2020;19(11):7064–80.