drones

MDPI

Article

Task Offloading Optimization Using PSO in Fog Computing for the Internet of Drones

Sofiane Zaidi [1], Mohamed Amine Attalah [2], Lazhar Khamer [1] and Carlos T. Calafate [3,*]

[1] Department of Mathematics and Computer Science, Research Laboratory on Computer Science's Complex Systems (RELA(CS)2), University of Oum El Bouaghi, Oum El Bouaghi 04000, Algeria; zaidi.sofiane@univ-oeb.dz (S.Z.); khamer.lazhar@univ-oeb.dz (L.K.)

[2] Department of Electronics, University Center of Tipaza, Tipaza 42000, Algeria; attalah.mohamed@cu-tipaza.dz

[3] Department of Computer Engineering (DISCA), Universitat Politècnica de València, 46022 Valencia, Spain

* Correspondence: calafate@disca.upv.es

Abstract: Recently, task offloading in the Internet of Drones (IoD) is considered one of the most important challenges because of the high transmission delay due to the high mobility and limited capacity of drones. This particularity makes it difficult to apply the conventional task offloading technologies, such as cloud computing and edge computing, in IoD environments. To address these limits, and to ensure a low task offloading delay, in this paper we propose PSO BS-Fog, a task offloading optimization that combines a particle swarm optimization (PSO) heuristic with fog computing technology for the IoD. The proposed solution applies the PSO for task offloading from unmanned aerial vehicles (UAVs) to fog base stations (FBSs) in order to optimize the offloading delay (transmission delay and fog computing delay) and to guarantee higher storage and processing capacity. The performance of PSO BS-Fog was evaluated through simulations conducted in the MATLAB environment and compared against PSO UAV-Fog and PSO UAV-Edge IoD technologies. Experimental results demonstrate that PSO BS-Fog reduces task offloading delay by up to 88% compared to PSO UAV-Fog and by up to 97% compared to PSO UAV-Edge.

Keywords: Internet of Drones; fog computing networks; particle swarm optimization; task offloading in IoD; unmanned aerial vehicles

Academic Editors: Diego González-Aguilera and Pablo Rodríguez-Gonzálvez

Received: 28 November 2024
Revised: 23 December 2024
Accepted: 25 December 2024
Published: 30 December 2024

Citation: Zaidi, S.; Attalah, M.A.; Khamer, L.; Calafate, C.T. Task Offloading Optimization Using PSO in Fog Computing for the Internet of Drones. *Drones* **2025**, *9*, 23. https://doi.org/10.3390/drones9010023

1. Introduction

The Internet of Drones (IoD) has recently emerged as a prominent area of research, attracting considerable interest from both civilian and military researchers. This field combines drones, also known as unmanned aerial vehicles (UAVs), with the Internet of Things (IoT), to facilitate various applications that include smart agriculture, environmental disaster management, video surveillance, and smart city initiatives, among others [1].

Fog computing is an extension of cloud computing that allows IoT devices to access nearby distributed computing and storage resources in order to support various delay-sensitive IoT applications, leading to lower latency for accessing IoT services, along with quicker processing of IoT requests [2].

The existing IoD technologies aim to integrate drones with the IoT to enhance connectivity, reliability, scalability, stability, data storage and processing, as well as security for real-time IoD applications [3]. These IoD technologies can be classified into four categories: IoD cloud computing, IoD edge computing, IoD fog computing, and IoD cellular networks. The primary objectives of IoD solutions include ensuring connectivity and coverage, scalability, reliability, stability, low latency, control of UAVs, data storage and processing, reduction in energy consumption, and security. Moreover, IoD solutions tackle various challenges to achieve these objectives, such as optimizing UAV trajectory, task offloading, and routing optimization.

IoD cloud computing aims to ensure a high capacity of computation and storage for data collected by UAVs by transferring these data to cloud servers. These servers are characterized by their high capacity for data processing and storage. However, IoD cloud computing cannot guarantee low latency due to the significant transmission time needed to transfer data between the UAVs and the cloud server, which are often situated at considerable distances from one another [4].

IoD edge computing and IoD fog computing technologies are designed to minimize data transmission latency through the local processing and storage of data collected by UAVs at the edge and fog nodes [5]. However, IoD edge computing is constrained by the limited resources of edge nodes, restricting its ability to provide high computation capacity and manage large volumes of data. In contrast, IoD fog computing can achieve minimal latency while providing substantial computational and storage capabilities, as it effectively integrates the local resources of fog nodes with those of cloud nodes [6].

Metaheuristic-based approaches, such as particle swarm optimization (PSO), are widely used for task offloading due to their ability to explore complex search spaces and achieve near-optimal solutions. However, these methods are not well adapted to real-time application in fog IoD scenarios. The authors in [7] highlighted several limitations of metaheuristic approaches in such environments. One main drawback is their relatively high execution time, which arises from the iterative nature of these algorithms as they converge to optimal or near-optimal solutions. In time-sensitive IoD applications, such as emergency response or live surveillance, the delay introduced by metaheuristic computations can offset their benefits. This issue becomes even more pronounced when handling large-scale networks with high task densities and stringent latency requirements. In our work, we mitigate these limitations by deploying the PSO algorithm on FBSs instead of fog UAVs, aiming to exploit the higher FBS computing and storage capacity. Furthermore, in our proposed task offloading optimization model for IoD fog computing, we have chosen to apply PSO over other heuristic methods due to the following reasons [8]:

- **Efficiency and simplicity:** Compared to other heuristic methods such as the genetic algorithms (GAs) and ant colony optimization (ACO), PSO is computationally efficient and straightforward to implement. This is particularly important in resource-constrained environments like the IoD.
- **Suitability for continuous optimization:** Task offloading involves optimizing a continuous search space (e.g., task offloading to FBS or cloud nodes), which aligns well with the strengths of PSO in handling continuous optimization problems.
- **Low computational overhead:** Unlike methods such as GAs, which involve complex selection, crossover, and mutation operations, PSO relies on simple position and velocity updates, making it more lightweight and suitable for UAVs with limited resources, such as battery power and computing capacity.
- **Proven performance:** PSO has been widely used and demonstrated effective results in similar task offloading and resource allocation problems. Its ability to quickly converge to near-optimal solutions makes it ideal for real-time IoD applications where latency is critical.

In IoD networks, reducing the delay for task offloading from UAVs to fog base stations (FBSs) represents a crucial challenge due to the limited UAV resources. To address this issue, in this paper we propose a task offloading solution called PSO BS-Fog, which extends the traditional fog computing, primarily composed of FBSs, by using the PSO algorithm to optimize both the transmission and fog computing delays. Specifically, PSO BS-Fog allows onboard computing resources of UAVs to be integrated with the fog computing environment, which consists of stationary FBS computational entities. Moreover, this solution reaps the benefits of the fog computing capabilities of the base stations, including processing, storage, as well as sensing, to enhance traditional fog computing capabilities using UAVs.

The primary challenge addressed by our proposed model is ensuring optimized transmission and computing delays, particularly for real-time sensitive applications. Therefore, the proposed model incorporates two main aspects:

- **PSO heuristic:** The PSO approach is used to optimize UAV task offloading among FBSs, ensuring minimized offloading delay.
- **Three-layer architecture:** The architecture consists of the following layers:
  - **Edge UAV layer:** Acts as client layer, requiring services from the BS-Fog infrastructure.
  - **FBS layer:** Composed of a set of fog base stations with enhanced computing and storage capabilities, enabling the processing of tasks offloaded from edge UAVs.
  - **Cloud layer:** Provides cloud services that UAVs can access either directly or through the FBS layer, offering additional computational and storage support. This layered architecture ensures efficient resource allocation and reduced delays for UAV tasks in fog IoD environments.

Our proposed PSO BS-Fog model is based on fog BS computing, where edge UAVs offload their tasks to fog BS nodes for processing. In the IoD, the limited processing, storage, and energy capacities of UAVs make stationary fog nodes (fog BS) more suitable for this type of network, especially when handling the offloading of tasks with significant computational demands. For this reason, we have chosen to use fog BS in our proposed PSO BS-Fog model, and have not considered fog UAVs. While the practical usability of fog UAVs may surpass that of fog BS in certain scenarios, the computational complexity associated with fog UAVs is significantly higher than that of fog BS.

The rest of this paper is organized as follows. Section 2 presents the related work on fog computing in IoD. Section 3 outlines different IoD applications. Section 4 describes the architecture of our proposed PSO BS-Fog, along with the mathematical formulation and algorithms of PSO BS-Fog. Section 5 presents the experimental study and discusses the obtained results. Finally, Section 6 concludes the paper with a summary and proposes some future research directions.

2. Related Work

Fog computing for the IoD has been explored in several recent studies. However, despite these initial efforts, there is currently no study that proves the effectiveness of task distribution for fog base stations based on heuristic methods (PSO, GA, etc.) against traditional task distribution methods (Uniform, Gaussian, etc.). Moreover, none of the studies in the related work (as shown in Table 1) have compared the different IoD technologies (base station fog computing, UAV fog computing, and UAV edge computing) for task offloading. This comparison is necessary to determine the most suitable computing technology for task offloading in an IoD environment.

Among the existing works, the authors of [9] introduced a framework that combines fog computing with a swarm of UAVs to effectively manage their computational tasks, ensuring low latency and high reliability. Additionally, they developed a genetic algorithm-based heuristic that optimizes the allocation of tasks to minimize energy consumption among the UAVs. The simulation results demonstrated that this algorithm can efficiently offload and process UAV tasks while achieving minimal energy consumption and meeting the necessary latency and reliability standards. However, it is required to simplify the algorithm's complexity to enhance its practical application.

The authors of [10] investigated the challenges of UAV-based fog computing in the context of smart industry 4.0. They introduced a framework for offloading computational tasks, enabling ground sensors to transfer their tasks to nearby fog UAVs. Additionally, this framework optimizes task allocation among fog UAVs, enhancing the total number of tasks processed while considering communication constraints and computation latency. The optimization method is based on a greedy algorithm. Simulation results indicated that this algorithm successfully optimized task allocation, achieving an optimality gap of no more than 7.5%. Moreover, there is potential for further enhancement of the platform by optimizing the trajectory of the UAVs.

Table 1. Comparison of fog computing studies for IoD.

Study	Entity Type of Fog Nodes		Objectives			Addressed Problems						IA Technique/Metaheuristic								Application				
	Stationary Fog Nodes	Mobile Fog Nodes	Real-Time Latency	Energy Consumption	Reliability	Task Offloading	Task Scheduling	Resources Allocation	UAVs Trajectory	Security	UAVs Collaboration	PSO	GA	AC	FWA	SSA	ML	DL	QLMDP	Smart Farming	Smart Industry	Robotic	Imagery	General
[9]		✓	✓	✓	✓	✓		✓					✓											✓
[10]		✓	✓			✓		✓													✓			
[11]		✓	✓																					✓
[12]		✓			✓					✓													✓	
[13]	✓			✓	✓					✓							✓			✓				
[14]	✓		✓	✓		✓		✓																✓
[15]		✓	✓			✓					✓								✓					✓
[16]	✓		✓			✓	✓					✓		✓										✓
[17]	✓		✓	✓		✓						✓	✓									✓		
[18]		✓	✓	✓		✓		✓					✓											✓
[19]	✓		✓			✓	✓								✓									✓
[20]		✓	✓			✓			✓								✓	✓						✓
[21]	✓		✓				✓					✓	✓											✓
[22]	✓		✓			✓										✓								✓
Our proposal	✓		✓			✓						✓												✓

In [11], the authors introduced a UAV-based fog computing system called UAV-Fog, which aims to provide data storage, flexible communication, and minimal latency for Internet of Things (IoT) applications. UAV-Fog leverages the capabilities of fog computing alongside the mobility of UAVs to facilitate IoT applications across various locations. Additionally, UAV-Fog provides many IoT services, including the discovery and integration of IoT resources, broker services, location-based services, and invocation and security services. A prototype of UAV-Fog was developed, and experimental results demonstrated its effectiveness in reducing latency.

In [12], the authors examined the challenges related to the security, safety, and privacy of fog UAVs within an airborne fog computing framework. Consequently, they introduced a GPS spoofing detection technique using a monocular camera along with the UAV's

Inertial Measurement Unit (IMU). The experimental findings indicated that the proposed approach outperforms the use of the IMU alone.

In [13], the authors proposed a UAV-based fog computing framework for smart agriculture to collect data from IoT sensors. This framework offloads data to stationary fog nodes at the network edge and enables UAVs to obtain tokens from these nodes for battery recharging. Both an intrusion detection system (IDS) and machine learning (ML) techniques are implemented at the stationary fog nodes to classify UAV behavior as either normal or malicious. The system demonstrated 99.7% accuracy in detecting intrusions and efficiently conserved energy through token-based elimination, ensuring reliable data collection despite attacks. However, this work focused only on single-model intrusion detection verification in IoD.

In [14], the authors introduced a method for resource allocation that optimizes radio and computational resources within a fog-assisted IoD framework to reduce both energy consumption and service latency. Each drone transmits its data to a fog node located at the base station (BS). A resource allocation algorithm is developed to assign bandwidth to drones operating in remote computing mode (RCM) and modifies the CPU frequency of edge and fog nodes to achieve a balance between latency and energy efficiency. Simulation results indicate that this algorithm significantly improves network performance compared to non-optimized drone computing modes or uniformly distributed bandwidth.

In [15], the authors proposed an architecture of a UAV–Fog collaborative network for data offloading in real-time applications to improve the quality of service required by drones in terms of latency and throughput.

In [16], the authors focused on the dynamic task scheduling technique in fog computing, aiming to balance efficiency and energy consumption. Therefore, the authors proposed a hybrid scheduling algorithm that combines ACO and PSO to improve task scheduling and rescheduling, which are key challenges in controlling edge devices in fog computing systems. The simulation results proved that the proposed method minimizes the processing time for data and requests in fog computing compared to the ACO and PSO heuristics.

The authors of [17] proposed a task offloading model based on fog computing to minimize energy consumption and meet task deadlines in Internet of Robotic Things (IoRT) environments. The model's performance was validated through extensive simulations and compared with optimization algorithms, including the GA, PSO, the whale optimization algorithm (WOA), and others. Simulation results showed that the proposed schema achieved the highest energy reduction compared to the GA, ABC, and other algorithms like ALO, WOA, PSO, and GWO.

In [18], the authors explored the impact of BS impairments on network performance and investigated how task offloading and resource allocation can be optimized through UAV-assisted edge computing. The authors focused on a Mixed Integer Nonlinear Programming (MINLP) problem involving the joint optimization of: task offloading decisions, uplink transmission power of the mobile vehicle, and computational resource allocation on the UAV. Moreover, the authors decomposed the optimization problem into two sub-problems: resource allocation (RA), which is solved using convex optimization techniques, and task offloading (TO), which is solved using the GA to optimize the RA function. Simulation results show that the proposed algorithm achieves the optimal solution and significantly improves vehicle unloading benefits compared to conventional methods.

In [19], the authors addressed the multi-task offloading problem in UAV-enabled fog computing networks by proposing a scheduling algorithm and an improved multi-task offloading scheme based on the Fireworks Algorithm (FWA) in order to minimize the total task delay. Simulation results demonstrated that the proposed approach outperforms the

GA and random algorithm, offering effective optimization for multi-task offloading delays in UAV-enabled fog computing networks.

In [20], the authors addressed the challenge of computation offloading and trajectory planning in UAV-assisted MEC systems, where a UAV provides services for multiple User Equipment (UE). The Joint Computation Offloading and Trajectory (JCOT) problem was formulated to optimize UE scheduling, computation offloading ratios, and UAV trajectory. To solve this problem, the authors proposed the KNN-DDPG algorithm with Prioritized Experience Replay (PER). Simulation results demonstrated that KNN-DDPG achieves a lower delay compared to the standard DDPG algorithm.

The authors of [21] proposed a hybrid GA and PSO approach to optimize task scheduling in fog computing environments. By combining the GA's exploration capabilities and PSO's exploitation strengths, the hybrid algorithm outperforms single algorithms (GA, PSO) and the hybrid PWOA. The results demonstrate that the proposed algorithm optimizes task allocation more effectively, enhancing system performance in terms of time metrics (execution time, response time, and completion time).

In [22], the authors proposed a multi-objective task offloading method based on the modified sparrow search algorithm (MOTO-MSSA) to optimize task offloading to fog nodes (FNs) by reducing cost and response time. Extensive simulations confirm that MOTO-MSSA significantly improves cost efficiency and response time while maintaining minimal overhead, outperforming existing optimization methods across various scenarios.

IoD fog computing improves the storage and processing capabilities of IoD cloud computing by extending their functionalities to IoD edge computing to minimize service latency and provide higher computational power to end users. However, a significant challenge in IoD fog computing is integrating UAVs at the edge computing layer with the cloud computing layer. This integration can be achieved through a range of communication technologies, including WiFi, WiMAX, and cellular networks, among others.

Discussion

The existing literature works on fog computing in IoD suggest integrating fog computing with IoD in order to improve the connectivity, reliability, scalability, stability, data storage and processing, and security for IoD real-time applications. In the following, we explain and discuss the existing fog computing IoD characteristics summarized in Table 1.

- **Entity type of fog nodes:** Fog computing was implemented on UAV nodes (mobile nodes) or on base stations fog (stationary nodes).
- **Objectives**
    - **Real-time latency:** The network ability to guarantee a reduced transmission delay for real-time services.
    - **Energy consumption:** Capacity of the network to supply and manage the powered energy of UAVs and IoT ground devices.
    - **Reliability:** Measured based on error-free operations on the network. Ideal network reliability means that no errors or failures were produced in this network.
- **Addressed problems**
    - **Task offloading:** The UAVs transmit their tasks towards the IoT cloud for processing and storage.
    - **Task scheduling:**
    - **Resource allocation:** Proposing a resource allocation strategy in order to mitigate collision and interference problems.
    - **UAV trajectory:** Takes into account UAV trajectory optimization in order to minimize the transmission delay and UAV energy consumption.
    - **Security:** Network safety against external threats and attacks.

- **UAV collaboration:** UAVs collaborate in order to accomplish a mission.
- **IA technique/Metaheuristic:** The IA techniques or metaheursrics used in order to optimize transmission delays or energy consumption or to grantee reliability. In our study, we have chosen the most used IA techniques/metaheuristics, such as PSO, AG, AC, FAW, ML, deep learning (DL), QLMDP.
  Application: The general or specific application domain of the proposed work, including smart farming, smart industry, robotics, imagery.

As shown in Table 1, each work has specific characteristics. We can classify these works into two categories: those based on stationary fog nodes and those based on mobile fog nodes. In the IoD, due to the limited processing and storage capacity of UAVs, stationary fog nodes (FBSs) are more appropriate for this type of network, especially when offloading tasks of significant size. For this reason, we have chosen to use stationary fog nodes in our proposed PSO BS-Fog model. Table 1 also highlights that the most commonly considered objective in the literature is real-time latency, which is a crucial factor addressed in our proposed model. Furthermore, the table illustrates that a primary challenge in related works is determining how to offload multiple tasks from edge UAVs to fog nodes, a challenge that our proposal also addresses. Additionally, the table presents various AI techniques and metaheuristics used in the cited works. We observe that many works employed PSO and GA metaheuristics; however, none has used PSO specifically for task offloading in fog IoD. It is important to note that the works [16,17,21] are not within the IoD context. Given the numerous advantages of PSO (such as its efficiency and simplicity, suitability for continuous optimization, low computational overhead, and proven performance), our proposed work is the first to utilize PSO for task offloading in fog IoD. Finally, Table 1 shows the application domains of the related works. We observe that most of these works, including our proposal, were designed for general applications rather than specific application domains.

3. IoD Applications

In the IoD literature, many solutions have been proposed for various applications, including delivery services, surveillance, search and rescue operations, path planning, mapping, and surveying.

3.1. Delivery Services

The use of drones for delivery services is becoming increasingly popular because of the many benefits that UAVs offer. These benefits include quicker delivery times, cost effectiveness, the ability to navigate geographical barriers and obstacles, and the ability to reach areas with traffic restrictions in both urban and rural areas. In order to adequately satisfy customer demand, businesses such as Amazon, Google, and iFood are beginning to show interest in the possibility of using drones for delivery services [23]. In that regard, IoD delivery services may be the best last-mile delivery option available [24]. However, the IoD presents a number of difficulties, including noise, privacy concerns, civilian law, safety concerns, limited payload, restricted flight range, battery capacity, and inclement weather. Moreover, as of right now, only small drones with payloads under 2.5 kg are permitted for delivery [25].

3.2. Surveillance

The observation of a target, whether it is an individual or a group, a location, or an activity, is referred to as surveillance [26]. This procedure includes gathering data from both rural and urban regions using a variety of technologies and tools such as cameras, sensors, and drones for observation [27]. For example, in smart cities, the IoD can be used for resolving surveillance issues linked to various surveillance activities, including, transportation, environmental monitoring, traffic flow analysis, environmental condition assessment, and public safety enhancement.

Drones must be equipped with antennas and passive sensors such as spectrometers, radiometers, and cameras for surveillance missions [28]. According to [29], drones should also be equipped with specific sensors and GPS, along with image processing techniques.

3.3. Search and Rescue Missions

The IoD is crucial for these missions because it can quickly and efficiently reach challenging environments [30] and save time. In addition, they carry out different search and rescue operations by providing supplies inside buildings, in addition to keeping an eye on the situation, modeling and analyzing it, and remotely identifying survivors or other targets. Some of the specifications for search and rescue missions are as follows: UAVs must have integrated real-time communication systems, high-resolution imaging cameras, sensors, and navigation systems [31].

To identify the position, condition, and environmental factors of targets, UAVs must process adequate stability and flight capabilities, in addition to a variety of sensors. These sensors include sonar, optical, infrared, thermal imaging, and other types. To transport the necessary supplies and equipment, it is also necessary to have a high load capacity and lengthy endurance.

3.4. Mapping and Surveying

Drone mapping is an alternative to traditional methods for creating 3D maps in the IoD, making such operations more efficient, particularly in hazardous or hard-to-reach regions. This reduces the amount of time that must be spent walking around the area, enhancing safety.

Drones with fixed or rotary wings can be employed for mapping purposes. Because of their faster speeds and endurance, the former ones are more practical for outdoor surveying and mapping in broad areas. However, the latter works better indoors and in smaller spaces [32].

4. Proposed Approach

4.1. PSO BS-FOG Architecture

To enhance fog computing capabilities, a new fog computing model for the IoD named PSO BS-Fog is proposed in this paper. Unlike conventional IoD infrastructure that typically depends on cloud computing or edge computing, PSO BS-Fog integrates both FBSs, edge UAVs, and cloud computing. This integration includes onboard computers on UAVs and computing resources on FBSs, expanding the computing resources available. Consequently, PSO BS-Fog enables drones to access computing resources of fog stationary nodes in order to enhance computational performance while reducing latency. Moreover, the proposed model facilitates direct access for UAVs to the FBS and cloud computing resources based on three layers (i.e., edge layer, static fog layer and cloud layer). The proposed model ensures improved reliability and lower latency for real-time applications. Furthermore, it supports various computing functions, such as processing, storage, and networking.

In this proposed model, all available computing resources (base station fog and cloud) are accessible to edge UAVs through a direct or hierarchical method. The PSO BS-Fog model is structured as a cross-layer framework organizing infrastructure components, as illustrated in Figure 1 and described below.

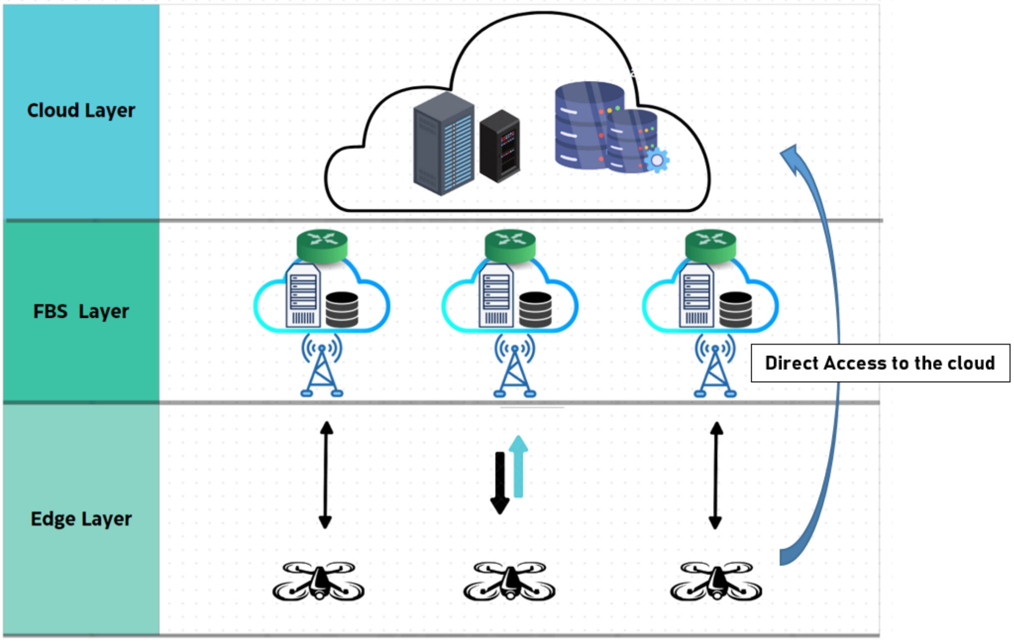

Figure 1. Proposed PSO BS-Fog architecture.

4.1.1. Edge Layer

The edge layer is the lowest level of PSO BS-Fog and the closest to the UAVs, which act as a general customer that requires BS-Fog services. Depending on the computational load, the UAV determines which infrastructure component (i.e., mobile edge, stationary fog, or cloud) is best suited to accomplish the requested tasks.

4.1.2. FBS Layer

This layer serves as an intermediary between the edge layer and the cloud layer. Moreover, the FBS layer is composed of a set of FBSs and is capable of communicating with the cloud Layer through the Internet, as well as with edge UAVs via a wireless connection. The FBS is distinguished by its higher computing and storage capabilities, which allow for the processing of tasks offloaded from UAVs because of the limited resources of edge UAVs. In our model, we propose using a PSO heuristic to optimize tasks distribution among FBSs.

In the context of our proposed PSO BS-Fog model, we suggest that each BS-Fog processes multiple tasks received from various concurrent drones in a simplified manner, following the FIFO (First In, First Out) strategy.

4.1.3. Cloud Layer

The cloud layer consists of IoD data centers, which offer cloud computing services. IoD data centers include both traditional stationary cloud data centers (static IoD cloud) and a dynamic IoD cloud comprising computing devices from IoD entities, such as temporarily allocated UAVs. These IoD cloud services can be accessed by UAVs either directly or through the stationary fog layer. This model leverages the benefits of the traditional IoD cloud and extends them to IoD entities. The PSO BS-Fog model is augmented by incorporating a temporary IoD cloud that utilizes IoD computing resources.

In our proposed model, we assume the following assumptions underlying the integration of FBSs, cloud computing, and UAVs:

- **Connectivity:** UAVs have stable wireless connections to FBSs and cloud servers, either directly or hierarchically via the FBS layer. This ensures task offloading can occur efficiently despite UAV mobility.

- **Heterogeneous resources:** FBSs provide intermediate computational and storage resources with lower latency compared to traditional cloud servers, while the cloud layer offers extensive computational capacity when tasks exceed FBS capabilities.
- **Task characteristics:** Tasks generated by UAVs are latency-sensitive and resource-demanding, which necessitates dynamic selection between the edge, fog, and cloud layers to balance delay and performance.
- **PSO-based optimization:** The PSO heuristic assumes that tasks can be distributed dynamically across available resources (FBSs and cloud) to minimize latency while leveraging the strengths of each layer.

By combining FBS and cloud resources, our model capitalizes on the proximity of fog computing for low-latency processing and the scalability of cloud computing for more complex tasks. These assumptions reflect practical IoD environments where UAVs operate in resource-constrained and dynamic scenarios.

4.2. PSO BS-FOG Formulation

In the context of our proposed PSO BS-Fog model, the content of a task typically depends on the application but generally involves data to be processed. These data can include raw information from sensors, images, video streams, or user inputs. For example, in a surveillance system, drones offload their video streams to the base station (BS), and this video stream becomes part of the task that needs to be processed.

The main challenge in task offloading within IoD fog computing is the optimization of offloading delay, which includes both transmission and computing delays. This section is divided into four subsections. The first subsection presents the channel model that affects the calculation of these delays. The second subsection details the computing model, which is based on the channel model to calculate transmission and computing delays. The third subsection introduces the PSO heuristic model employed to optimize these calculated delays. Finally, the last subsection describes our proposal for integrating the presented models to effectively optimize both transmission and computing delays.

Task offloading to various FBSs and the task scheduling in one FBS are the crucial challenges in IoD networks based on FBSs. For the first challenge, we have proposed an algorithm which is based on PSO heuristics to produce optimal task offloading solutions that provide minimal transmission and fog computing delays. For the second challenge, we have used a scheduling algorithm that calculates the transmission and computing delays of each task, and after that it calculates the maximum delay of all tasks allocated only in one FBS [19].

4.2.1. PSO BS-Fog Channel Model

Figure 2 represents the PSO BS-Fog channel model. To achieve connectivity over expansive geographic regions, the communication range between aerial nodes and ground devices must be calculated. To maintain network quality, it is essential to calculate path loss to determine the optimal path for UAVs, ensuring uninterrupted data delivery between nodes. The overall path loss (considered as free space) from UAV to ground FBS is calculated using Equations (1) and (2) [15]:

$$L[dB] = PL(freespace) + PL(excess) \tag{1}$$

$$L[dB] = 20log_{10}(\frac{4\pi d f_c}{c}) + L_0 \tag{2}$$

In Equation (1), the total path loss is formulated to compute the attenuation along the path from air devices (UAV) to the FBS, where f_c is the channel frequency, c is the speed of the light, d_i is the distance between UAV_j and FBS_i (as shown in Figure 2), and L_0 is the excess path loss including line of sight (LoS) for direct communications, and non-line-of-sight (NLoS) for indirect communication [15]. The total received power by FBS devices with respect to the air devices, considering path loss and the range, is given by Equation (3), as in [15]:

$$Pr_{(i)}d_i(t) = Pt_{(j)}\frac{G\lambda}{(4\pi d_i)^2 L[dB]_{d_i}} \tag{3}$$

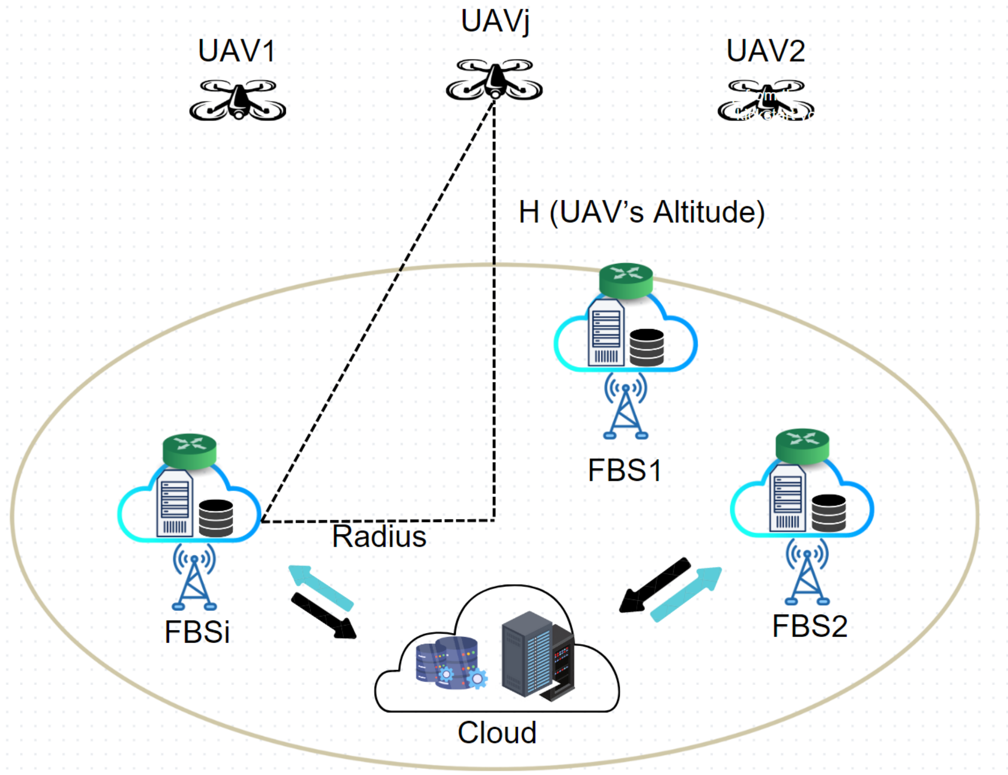

Figure 2. PSO BS-Fog channel model.

In this context, $Pr_{(i)}$ represents the power received by FBS_i, which depends on the distance $d_i(t)$, while $Pt_{(j)}$ denotes the power transmitted by the air device UAV_j. G is the gain of the transmitter (UAV) Tx and the receiver Rx (FBSs) antennas measured in dB, λ is the wavelength of the signal, and $L[dB]_{d_i}$ is the total path loss between the air device and ground device.

From Equation (3), the transmitted power Pt_j from the air devices UAV_j and the ground devices is used to calculate the signal-to-noise ratio (SNR). Therefore, the $SNR_{(i)}$ is calculated using Equation (4) as in [19]:

$$SNR_{(i)} = \frac{Pt_{(j)}/L[dB]_{(d_i)}}{n_0} \tag{4}$$

where $L[dB]$ is the total path loss, and n_0 is the noise power. The maximum transmission data rate T_i from air device to ground device can be expressed by Equation (5) as shown in [19]:

$$T_i = B.log_2(1 + SNR_{(i)}) \tag{5}$$

In this context, B represents the channel bandwidth.

4.2.2. PSO BS-Fog Computing Model

Initially, we define the vector indicating the size of each task as $S = \{S1, S2, \ldots, Sj, \ldots, Sn\}$. Next, we can determine the transmission delay $Trans(i, j)$ associated with offloading $Task_j$ to the FBS_i using the transmission rate Ti as follows [19]:

$$Trans_{(i,j)} = \frac{S_j}{T_i} \tag{6}$$

The processing time of FBS_i depends on the CPU frequency f_i of the FBS_i. The CPU cycles required to process each bit of data are represented as α_i, while $P_{(i,j)}$ refers to the computing delay associated with $Task_j$ in the FBS_i, and $P_{(i,j)}$ can be calculated using the following equation [19]:

$$P_{(i,j)} = \frac{S_j.\alpha_i}{f_i} \tag{7}$$

4.2.3. Algorithmic Structure of Standard PSO

PSO employs a swarm of particles that update their positions and velocities in each iteration to conduct a search for optimal solution. Each particle moves towards its own previous best position (*pbest*) and the global best position (*gbest*) within the swarm algorithm [33]. Thus, one has:

$$pbest(i,t) = \underbrace{arg \;\; min}_{\text{k=1,...,t}}[f(P_i(k)], \quad i \in \{1, 2, ..., N_p\} \tag{8}$$

$$gbest(t) = \underbrace{arg \;\; min}_{\text{k}}[f(P_i(k)] \tag{9}$$

where i is the index of the particle, f represents the fitness function, t is the current iteration number, N_p is the total number of particles, and P denotes the position of the particle. P and the velocity V of particles are updated by the following equations [33]:

$$V_i(t+1) = \omega V_i(t) + c_1 r_1 (pbest(i,t) - P_i(t)) + c_2 r_2 (gbest(t) - P_i(t)) \tag{10}$$

$$P_i(t+1) = P_i(t) + V_i(t+1) \tag{11}$$

where ω is the inertia weight used to balance the global exploration and local exploitation, r_1 and r_2 are uniformly distributed random variables within range [0, 1], and c_1 and c_2 are positive constant parameters called "acceleration coefficients".

4.3. PSO BS-Fog Algorithms

The general proposed PSO BS-Fog algorithm is presented in Algorithm 1. The UAV executes the pseudo-code shown in this figure when it wants to offload tasks to a set of FBSs. Before generating a set of a population of particles (NPopulation), Step 1 consists of the problem definition, which defines the cost function that represents the total offloading delay for each FBS generated by the algorithm shown in Algorithm 2. After the first step, Step 2 initializes a vector of NPopulation cost, velocity, best position, and best cost. Step 3 applies the PSO heuristic Equations (8)–(11) in order to find the best position of each task, and it calculates the task cost that represents the offloading delay (transmission and fog computing delay). Finally, Step 4 generates the minimal delay of all offloading tasks.

Algorithm 2 shows the pseudo-code of the total delay calculation algorithm for offloading tasks in the proposed PSO BS-Fog. This code is applied in Step 3 of the PSO BS-Fog algorithm (Algorithm 1) to calculate transmission delay and fog computing delay for all offloading tasks to a single FBS. Step 1 of this code calculates the data rate based on Equations (1)–(5). Following this, Step 2 calculates the transmission delay and fog computing delay of each offloading task based on Equations (6) and (7) and then generates the total task offloading delay, which is defined as the maximum of all task offloading delays.

Algorithm 1 Proposed PSO BS-Fog algorithm

Inputs: Number of fog base stations (NFBS), number of offloading tasks (NTasks), Iteration number (NIt), Population number (NPopulation), Inertia Weight (w), Local acceleration coefficient (c1), Global acceleration coefficient (c2).

Outputs: Optimized tasks offloading total delay for all fog base stations (Best_Cost)

Step 1. Problem Definition

CostFunction = Total delay for all fog base stations (Algorithm 2)

Step 2. Initialize a vector of NPopulation particles that contains for each case the interested particles data: position, cost, velocity, best position, and best cost.

for i=1: NPopulation

Step 2.1. Initialize randomly (within the interval [1, NFBS]) the position of each particle of the population i (NTasks particles)

Step 2.1. Initialize randomly the velocity of each particle of the population i

Step 2.2. Evaluation CostFunction of each particle of the population i

Step 2.3. Update local Best of each particle of the population i

Step 2.4. Update Global Best of all NTasks particles of the population i

end

Step 3. PSO Main Loop

for it=1: NIt

    for i=1: NPopulation

Step 3.1. Update the velocity of each particle of the population i by Equation (10)

Step 3.2. Update the position of each particle of the population i by Equation (11)

Step 3.3. Evaluation CostFunction of each particle of the population i

Step 3.5. Update local Best of each particle of the population i

Step 3.6. Update Global Best of all NTasks particles of the population i

    end

end

Step 4. Generate best cost of all populations.

Algorithm 2 New total delay calculating algorithm of offloading tasks for single fog Base Station

Inputs: Task sizes vector (S), Radius, Channel Bandwidth (B), UAVs Altitude (H), BS-Fog CPU cycles ( ), BS-Fog CPU frequency (fi), UAV Transmitting power (Pt), Noise power (n0)

Output: Total delay of all tasks(Total_Delay)

Step 1. Calculate Data Rate (T)

Step 1.1. Calculate free space propagation and communication Range

lambda = physconst('LightSpeed')/fc;

Range = round(hypot(H,Radilambdaus));

Step 1.2. Calculate Packet Loss, SNR and T

L = fspl(Range, lambda);

SNR = (Pt/L)/n0;

T = B * log2(1+SNR);

Step 2. Calculate total delay (Total_Delay)

Step 2.1. Calculate transmission Delay (Trans) and fog computing delay (F)

Trans = S/T;

F = (S*α)/fi;

Step 2.2. Calculate total transmission Delay (Total_TDelay)

and total fog computing delay (Total_Fdelay)

TmTotal = Trans(1);

TpTotal = Trans(1);

for i = 1:(length(S)-1)   TotalTDelay = TmTotal + Trans(i+1);

  TotalFdelay = TpTotal + F(i);

  if (TmTotal>TpTotal);

  TpTotal = TmTotal;

  end

end

Total_Delay = TpTotal + P(length(S));

5. Performance Evaluation

In this section, we present the experimental results to evaluate the performance of our proposed PSO BS-Fog for task offloading in IoD.

5.1. Simulation Setup

To measure the performance of PSO BS-Fog for task offloading in the IoD, we have conducted several simulations performed on MATLAB R2016a version. The simulation parameters are listed in Table 2. We examined a scenario involving various types of smart devices, UAV-Edge, UAV-Fog nodes, and FBSs.

Table 2. General simulation parameters.

Parameter	Value
Radius	500
Channel bandwidth (B)	0.2 MHz
BS-Fog number (NFBS)	{1, . . . , 10}
UAV altitude (H)	360 m
Distribution of the fog nodes	Uniform Distribution
Dimension of the tasks	Generated within the range [1 kbit, 200 kbit]
TASK number (Ntasks)	{50, . . . , 450}
BS-Fog CPU cycles (C_{cpu})	2000 cycle/bit
UAV transmitting power (Pt)	10 dBm
BS-Fog CPU frequency (f_i)	2.4×10^9 cycle/s
Noise power (n_0)	−105 dBm
PSO population number (NPopulation)	100
Iteration number (NIt)	100
Inertia Weight (w)	1
Local acceleration coefficient (c1)	1.5
Global acceleration coefficient (c2)	2.0

We compared the following task distribution methods under FBS in the IoD:

- **PSO BS-Fog:** Our proposed solution, which utilizes PSO for task offloading to FBSs and provides optimized minimal delay;
- **Uniform:** A traditional uniform distribution method that allocates tasks among FBSs in a uniform way;
- **Gaussian:** A traditional Gaussian distribution method that allocates tasks among FBSs based on a Gaussian distribution;
- **Pareto:** A Pareto distribution method that allocates tasks among FBSs based on the Pareto function.

5.2. Experimental Results

Figure 3 shows the Best Delay (in seconds) as a function of the FBS number (NFBS), varying from 1 to 10 (with the number of offloaded tasks fixed at 100). As illustrated in this figure, as the NFBS increases, the Best Delay for all offloading methods decreases due to the higher FBS capacities in terms of storage and processing. Moreover, the figure demonstrates that our proposed BS-Fog, based on the PSO heuristic, achieves significantly lower delays compared to the Uniform, Gaussian, and Pareto offloading methods. For example, when NFBS equals 6, BS-Fog achieves a Best Delay of 0.0015 s, while the other traditional methods have a Best Delay around 0.18 s. This represents a delay reduction of approximately 99.17%. By utilizing the PSO heuristic, BS-Fog provides an optimal task offloading solution, minimizing both transmission and processing delays.

Figure 4 presents the Best Delay achieved by each offloading method as a function of task number (TN), with NFBS fixed at 5. As shown in this figure, when the TN increased, the Best Delay for all offloading methods also increased. Specifically, using the proposed algorithm, as the number of tasks increased from 50 to 450, the Best Delay increased from 0.1 s to 0.22 s. This increase is due to the higher number of tasks being offloaded, which leads to greater delays in processing. Moreover, Figure 4 illustrates that our proposed PSO BS-Fog achieves a lower Best Delay compared to the other methods. This is attributed to the PSO heuristic's ability to generate optimal task offloading solutions. For a task number of 250, the best delay achieved by the PSO-based BS-Fog is 0.11 s, while other distribution methods such as Uniform, Gaussian, and Pareto show Best Delays ranging from 0.27 s to 0.36 s.

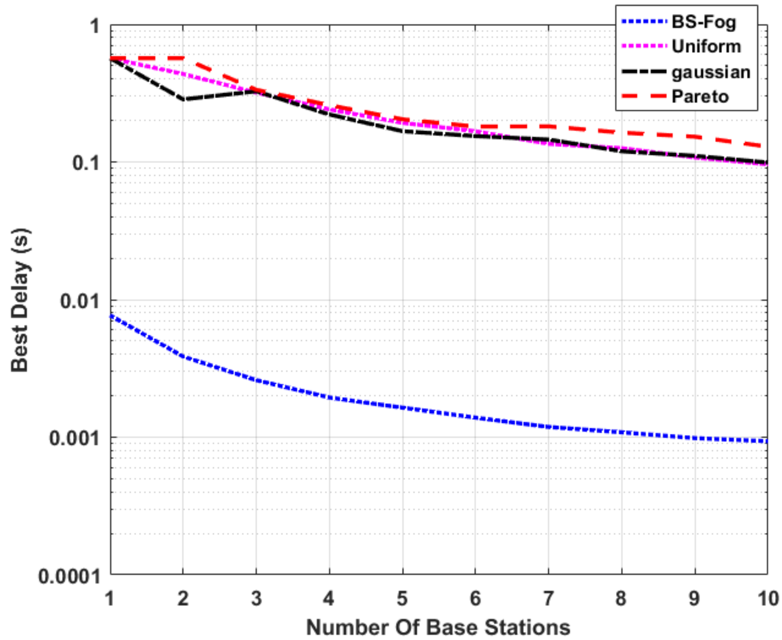

Figure 3. Variation in Best Delay with number of fog base stations for various task offloading methods.

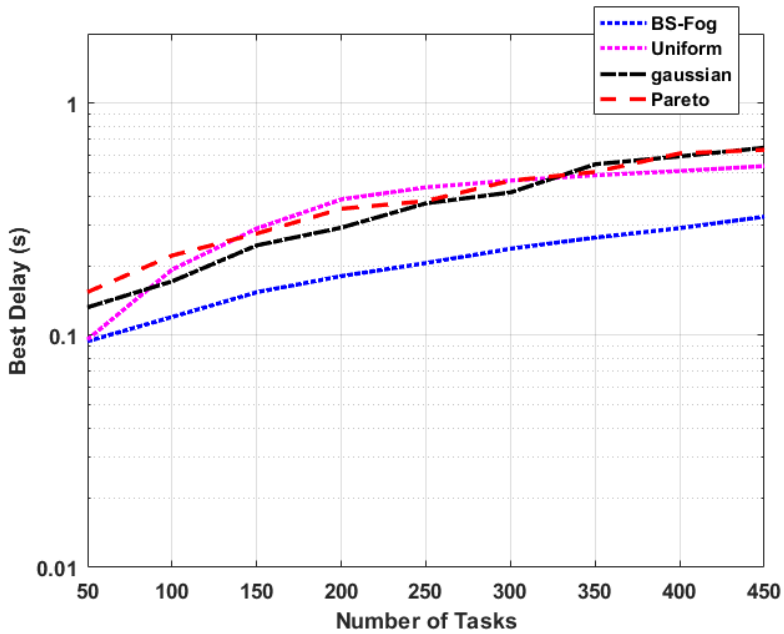

Figure 4. Variation in Best Delay with task number for various task offloading methods.

The Best Delay achieved by each IoD technology (PSO BS-Fog, PSO UAV-Fog, and PSO UAV-Edge) as a function of the number of nodes (i.e., BS number for PSO BS-Fog, UAV number for PSO UAV-Fog and PSO UAV-Edge) is presented in Figure 5, with the number of offloaded tasks fixed at 100. As shown in this figure, when the NFBS for PSO BS-Fog (respectively the UAV number for PSO UAV-Fog and PSO UAV-Edge) increases, the Best Delay decreases due to the higher processing and storage capacities of these nodes. Moreover, Figure 5 demonstrates that our proposed PSO BS-Fog achieves a significantly lower Best Delay compared to PSO UAV-Fog and PSO UAV-Edge technologies. For example, when the number of nodes is six, and the number of tasks is 100, the Best Delay using PSO BS-Fog is 0.05 s, whereas using PSO UAV-Fog results in a delay of 0.35 s, and PSO UAV-Edge shows a delay of 2 s. This means that PSO BS-Fog achieves a delay reduction of approximately 85.7% compared to PSO UAV-Fog and 97.5% compared to PSO UAV-Edge.

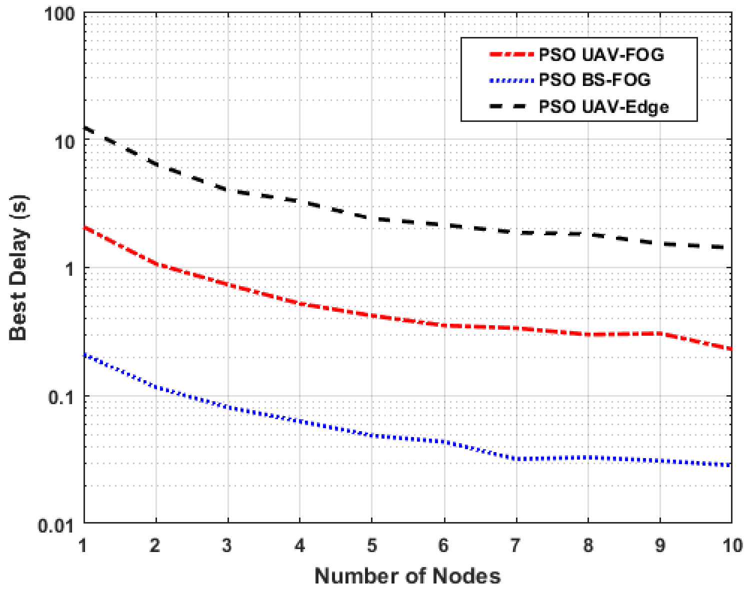

Figure 5. Variation in Best Delay with number of nodes for IoD technologies.

Figure 6 presents the variation in Best Delay with task number (TN) for IoD technologies, with NFBS or UAV-Fog number fixed at five. As shown in this figure, when TN increases, all IoD technologies (PSO BS-Fog, PSO UAV-Fog, and PSO UAV-Edge) experience higher Best Delays due to the limited capacities of the nodes. However, PSO BS-Fog achieves a significantly lower Best Delay compared to both PSO UAV-Fog and PSO UAV-Edge. For example, when the number of tasks is 200, the Best Delay using the proposed PSO BS-Fog method is 0.1 s, while using PSO UAV-Fog results in a delay of 0.9 s and using PSO UAV-Edge results in a delay of 5 s. This means that PSO BS-Fog achieves a delay reduction of approximately 88.9% compared to PSO UAV-Fog and 98% compared to PSO UAV-Edge. This difference is primarily due to the higher processing and storage capacities of the fog base stations (FBSs) in PSO BS-Fog.

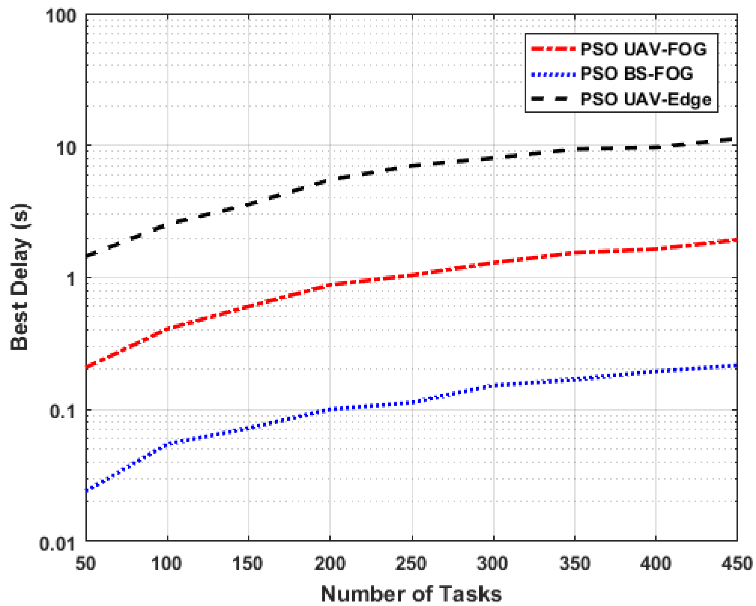

Figure 6. Variation in Best Delay with number of tasks for IoD technologies.

Figure 7 illustrates the variation in Best Delay as a function of the transmission data rate (DR) for IoD technologies, with NFBS or UAV number fixed at five. It is important to note that the variation in the data rate affects all the parameters mentioned in Equations (2)–(6). The results show that the PSO BS-Fog method achieves a lower Best Delay compared to both PSO UAV-Fog and PSO UAV-Edge. This is primarily due to the higher computing capacities of fog base stations (FBSs) compared to the limited computing capabilities of UAVs, which allow PSO BS-Fog to process tasks more efficiently and with reduced delay.

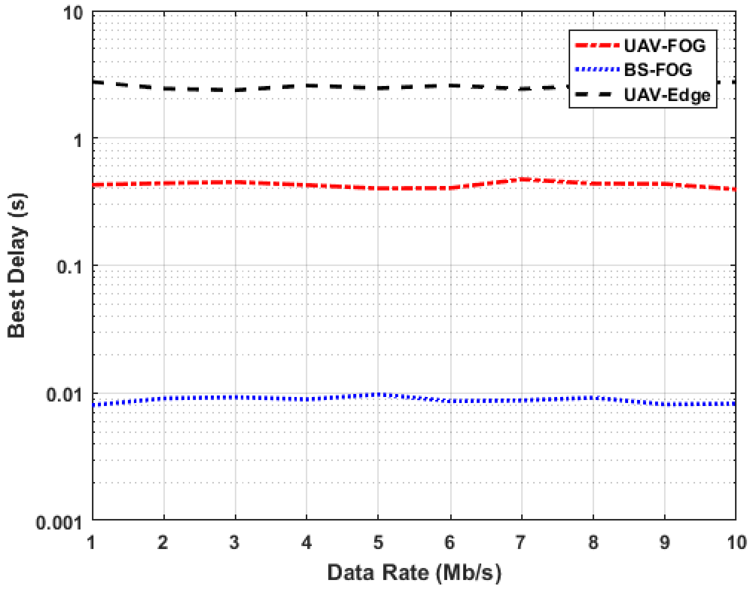

Figure 7. Variation in Best Delay with data rate for IoD technologies.

Figure 8 illustrates the variation in Best Delay as a function of UAV altitudes, with the number of FBSs or UAVs fixed at five and the number of tasks set to 100. As shown in this figure, the proposed PSO BS-Fog achieves a lower Best Delay compared to PSO UAV-Fog and PSO UAV-Edge due to the superior computing capabilities of FBSs compared to those of FUAVs and edge UAVs.

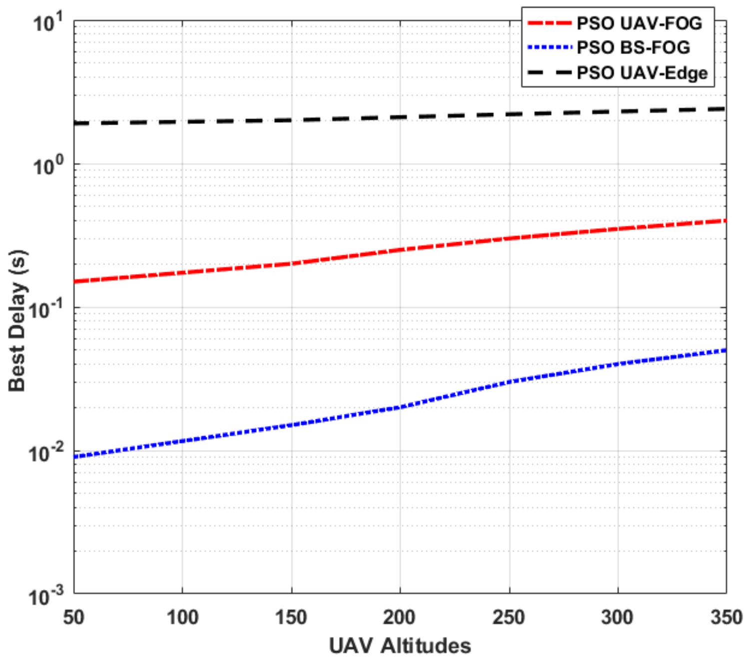

Figure 8. Variation in Best Delay with UAV altitudes for IoD technologies.

6. Conclusions

In this paper, we have proposed PSO BS-Fog for task offloading optimization in IoD networks. The proposed solution utilizes the PSO heuristic to optimize UAV task dispatching among fog base stations (FBSs) to minimize offloading delay (transmission delay and fog computing delay). The simulation results demonstrated that PSO BS-Fog achieves better performance in terms of task offloading delay compared to traditional offloading methods (i.e., Uniform distribution, Gaussian distribution, and Pareto distribution). Moreover, the simulation proved that PSO BS-Fog achieves a significantly lower Best Delay compared to existing IoD technologies (i.e., PSO UAV-Fog and PSO UAV-Edge) due to the higher storage and processing capacities of FBSs. For example, when the number of tasks is 200, PSO BS-Fog achieved a Best Delay of 0.1 s, PSO UAV-Fog showed a delay of 0.9 s, and PSO UAV-Edge had a delay of 5 s, representing a delay reduction of approximately 88.9% compared to PSO UAV-Fog, and 98% compared to PSO UAV-Edge. The experimental results highlight the influence of the number of nodes, the number of tasks, and the transmission data rate on offloading delay, which is affected by the range and channel conditions.

As future work, we aim to design a hybrid task offloading solution integrating PSO BS-Fog, PSO UAV-Fog, and PSO UAV-Edge, leveraging the strengths of these IoD technologies. Additionally, we plan to incorporate energy consumption into the optimization framework, exploring energy-aware strategies to balance delay and power usage. We will also address security concerns by integrating secure communication protocols and authentication methods to protect against risks like data interception and hijacking. Lastly, we intend to compare our approach with AI techniques for task offloading, dynamic task allocation, and real-time decision-making under unpredictable conditions. For instance, we plan to compare our approach with machine learning techniques such as reinforcement learning and deep learning to assess the trade-offs and benefits of both PSO-based and ML-driven methods for task offloading in the IoD.

Author Contributions: Conceptualization, S.Z. and M.A.A.; methodology, S.Z.; software, L.K.; validation, M.A.A., L.K. and C.T.C.; formal analysis, S.Z.; investigation, M.A.A.; resources, C.T.C.; data curation, L.K.; writing—original draft preparation, S.Z. and M.A.A.; writing—review and editing, C.T.C.; visualization, L.K.; supervision, S.Z.; project administration, M.A.A.; funding acquisition, C.T.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially funded by EU project REMARKABLE (grant agreement ID: 101086387) under program HORIZON-MSCA-2021-SE-01-01; by project CIPROM/2023/29, funded by "Direcció General de Ciència i Investigació", Generalitat Valenciana, Spain; by R&D project PID2021-122580NB-I00, from MICIU/AEI/10.13039/501100011033, and "ERDF A way of making Europe".

Data Availability Statement: Dataset available on request from the authors.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zaidi, S.; Atiquzzaman, M.; Calafate, C.T. Internet of flying things (IoFT): A survey. *Comput. Commun.* **2021**, *165*, 53–74. [CrossRef]
2. Asghari, A.; Sohrabi, M.K. Server placement in mobile Cloud computing: A comprehensive survey for Edge computing, Fog computing and Cloudlet. *Comput. Sci. Rev.* **2024**, *51*, 100616. [CrossRef]
3. Zhu, H.; Demirbaga, U.; Aujla, G.S.; Shi, L.; Zhang, P. Explainable Edge AI Framework for IoD-Assisted Aerial Surveillance in Extreme Scenarios. *IEEE Internet Things J.* 2024, *Early Access*. [CrossRef]
4. Reddy, P.B.; Sudhakar, C. IOTD: Intelligent offloading of tasks with deadlines in Edge-Fog-Cloud computing environment using hybrid approach. *Clust. Comput.* **2024**, *27*, 9873–9885. [CrossRef]

5. Srirama, S.N. A decade of research in Fog computing: Relevance, challenges, and future directions. *Softw. Pract. Exp.* **2024**, *54*, 3–23. [CrossRef]
6. Pallewatta, S.; Kostakos, V.; Buyya, R. Reliability-aware Proactive Placement of Microservices-based IoT Applications in Fog Computing Environments. *IEEE Trans. Mob. Comput.* **2024**, *23*, 11326–11341. [CrossRef]
7. Javanmardi, S.; Sakellari, G.; Shojafar, M.; Caruso, A. Why it does not work? Metaheuristic task allocation approaches in Fog-enabled Internet of Drones. *Simul. Model. Pract. Theory* **2024**, *133*, 102913. [CrossRef]
8. Jain, M.; Saihjpal, V.; Singh, N.; Singh, S.B. An overview of variants and advancements of PSO algorithm. *Appl. Sci.* **2022**, *12*, 8392. [CrossRef]
9. Hou, X.; Ren, Z.; Cheng, W.; Chen, C.; Zhang, H. Fog based computation offloading for swarm of Drones. In Proceedings of the International Conference on Communications (ICC), Shanghai, China, 20–24 May 2019; pp. 1–7.
10. Lee, G.; Saad, W.; Bennis, M. Online optimization for UAV-assisted distributed Fog computing in smart factories of industry 4.0. In Proceedings of the Global Communications Conference (GLOBECOM), Abu Dhabi, United Arab Emirates, 9–13 December 2018; pp. 1–6.
11. Mohamed, N.; Al-Jaroodi, J.; Jawhar, I.; Noura, H.; Mahmoud, S. UAVFog: A UAV-based Fog computing for Internet of Things. In Proceedings of the International Conference on SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation, (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI-2017), San Fransisco Bay Area, CA, USA, 4–8 August 2017; pp. 1–8.
12. He, D.; Qiao, Y.; Chan, S.; Guizani, N. Flight security and safety of Drones in airborne Fog computing systems. *IEEE Commun. Mag.* **2018**, *56*, 66–71. [CrossRef]
13. Sajid, J.; Hayawi, K.; Malik, A.W.; Anwar, Z.; Trabelsi, Z. A Fog computing framework for intrusion detection of energy-based attacks on UAV-assisted smart farming. *Appl. Sci.* **2023**, *13*, 3857. [CrossRef]
14. Shams, F.; Lottici, V.; Giannetti, F. Joint Spectrum and Computing Resource Allocation in Fog-Assisted Drone Communications for Ambiental Services. *IEEE Access* **2023**, *11*, 115348–115366. [CrossRef]
15. Gupta, A.; Gupta, S.K. Unmanned aerial vehicles in collaboration with Fog computing networks for improving quality of service. *Int. J. Commun. Syst.* **2024**, *37*, E5759. [CrossRef]
16. Gautam, A.K.; Raza, N.; Mishra, J. Hybrid Optimal Offloading Approach in Fog Computing using PSO. In Proceedings of the International Conference on Big Data & Machine Learning (ICBDML), Singapore, 24–25 February 2024; pp. 157–160.
17. El Menbawy, N.; Ali, H.A.; Saraya, M.S.; Ali-Eldin, A.M.; Abdelsalam, M.M. Energy-efficient computation offloading using hybrid GA with PSO in internet of robotic things environment. *J. Supercomput.* **2023**, *79*, 20076–20115. [CrossRef]
18. Dong, A.; Hu, Y.; Mei, Z.; Feng, K. Joint Task Offloading and Resource Allocation Optimisation for BS-Impaired UAV-Assisted Edge Computing. In Proceedings of the International Conference on Data-driven Optimization of Complex Systems (DOCS), Hangzhou, China, 16–18 August 2024; pp. 607–612.
19. Li, X.; Zhou, L.; Sun, Y.; Ulziinyam, B. Multi-task offloading scheme for UAV-enabled fog computing networks. *EURASIP J. Wirel. Commun. Netw.* **2020**, *2020*, 230. [CrossRef]
20. Lu, Y.; Xu, C.; Wang, Y. Joint Computation Offloading and Trajectory Optimization for Edge Computing UAV: A KNN-DDPG Algorithm. *Drones* **2024**, *8*, 564. [CrossRef]
21. Saad, M.; Enam, R.N.; Qureshi, R. Optimizing multi-objective task scheduling in fog computing with GA-PSO algorithm for big data application. *Front. Big Data* **2024**, *7*, 1358486. [CrossRef] [PubMed]
22. Khedr, A.M.; Alfawaz, O.; Alseid, M.; El-Moursy, A. MOTO-MASSA: Multi-objective task offloading based on modified sparrow search algorithm for fog-assisted IoT applications. *Wirel. Netw.* **2024**, 1–16. [CrossRef]
23. Shirabayashi, J.V.; Ruiz, L.B. Toward uav path planning problem optimization considering the internet of Drones. *IEEE Access* **2023**, *11*, 136825–136854. [CrossRef]
24. Chi, N.T.K.; Phong, L.T.; Hanh, N.T. The Drone delivery services: An innovative application in an emerging economy. *Asian J. Shipp. Logist.* **2023**, *39*, 39–45. [CrossRef]
25. Alkouz, B.; Shahzaad, B.; Bouguettaya, A. Service-based Drone delivery. In Proceedings of the International Conference on Collaboration and Internet Computing (CIC), Virtual, 13–15 December 2021; pp. 68–76.
26. Gohari, A.; Ahmad, A.B.; Rahim, B.A.R.; Supa'at, A.S.M.; Abd Razak, S.; Gismalla, M.S.M. Involvement of surveillance Drones in smart cities: A systematic review. *IEEE Access* **2022**, *10*, 56611–56628. [CrossRef]
27. Abualigah, L.; Diabat, A.; Sumari, P.; Gandomi, A.H. Applications, deployments, and integration of internet of Drones (iod): A review. *IEEE Sens. J.* **2021**, *21*, 25532–25546. [CrossRef]
28. Fan, J.; Zhang, L.; Shen, G.; Li, Z.; Cao, P.; Lei, L. A self-deployment algorithm for flying ad hoc networks (fanets) towards low probability of detection coverage in directional antennas. In Proceedings of the International Conference on Communication Technology (ICCT), Nanning, China, 28–31 October 2020; pp. 898–902.
29. Huang, H.; Savkin, A.V. An algorithm of reactive collision free 3-d deployment of networked unmanned aerial vehicles for surveillance and monitoring. *IEEE Trans. Ind. Inform.* **2019**, *16*, 132–140. [CrossRef]

30. Hassanalian, M.; Abdelkefi, A. Classifications, applications, and design challenges of Drones: A review. *Prog. Aerosp. Sci.* **2017**, *91*, 99–131. [CrossRef]
31. Ghamari, M.; Rangel, P.; Mehrubeoglu, M.; Tewolde, G.S.; Sherratt, R.S. Unmanned aerial vehicle communications for civil applications: A review. *IEEE Access* **2022**, *10*, 102492–102531. [CrossRef]
32. Alsayed, A.; Nabawy, M.R. Stockpile volume estimation in open and confined environments A review. *Drones* **2023**, *7*, 537. [CrossRef]
33. Yudong, Z.; Shuihua, W.; Genlin, J. A comprehensive survey on particle swarm optimization algorithm and its applications. *Math. Probl. Eng.* **2015**, *1*, 9312.