Hindawi
Security and Communication Networks
Volume 2024, Article ID 9871375, 1 page
https://doi.org/10.1155/2024/9871375

Retraction

Retracted: Security and Makespan Trade-Off Strategy in Fog-Enabled IoT Networks

Security and Communication Networks

Received 8 January 2024; Accepted 8 January 2024; Published 9 January 2024

This article has been retracted by Hindawi following an investigation undertaken by the publisher [1]. This investigation has uncovered evidence of one or more of the following indicators of systematic manipulation of the publication process:

(1) Discrepancies in scope

(2) Discrepancies in the description of the research reported

(3) Discrepancies between the availability of data and the research described

(4) Inappropriate citations

(5) Incoherent, meaningless and/or irrelevant content included in the article

(6) Manipulated or compromised peer review

The presence of these indicators undermines our confidence in the integrity of the article's content and we cannot, therefore, vouch for its reliability. Please note that this notice is intended solely to alert readers that the content of this article is unreliable. We have not investigated whether authors were aware of or involved in the systematic manipulation of the publication process.

Wiley and Hindawi regrets that the usual quality checks did not identify these issues before publication and have since put additional measures in place to safeguard research integrity.

We wish to credit our own Research Integrity and Research Publishing teams and anonymous and named external researchers and research integrity experts for contributing to this investigation.

The corresponding author, as the representative of all authors, has been given the opportunity to register their agreement or disagreement to this retraction. We have kept a record of any response received.

References

[1] X. Wang, B. Veeravalli, Y. Tan, and Z. Tong, "Security and Makespan Trade-Off Strategy in Fog-Enabled IoT Networks," *Security and Communication Networks*, vol. 2022, Article ID 5184453, 13 pages, 2022.

Hindawi
Security and Communication Networks
Volume 2022, Article ID 5184453, 13 pages
https://doi.org/10.1155/2022/5184453

Research Article

Security and Makespan Trade-Off Strategy in Fog-Enabled IoT Networks

Xiaoli Wang,[1] **Bharadwaj Veeravalli**,[2] **Yulong Tan**,[1] **and Zhihao Tong**[1]

[1]*School of Computer Science and Technology, Xidian University, Xi'an 710071, China*
[2]*Department of Electrical and Computer Engineering, National University of Singapore, 4 Engineering Drive 3, Singapore*

Correspondence should be addressed to Xiaoli Wang; wangxiaoli@mail.xidian.edu.cn

Received 28 January 2022; Accepted 5 April 2022; Published 23 May 2022

Academic Editor: Xingsi Xue

Fog computing relieves the Internet of Things (IoT) from a heavy burden of workload computation, consequently minimizes workload makespan, and reduces the considerable energy consumption of IoT devices. However, the fog-enabled IoT poses a risk to data security due to the intrinsic open and distributed structure of the fog. Hence, it is imperative to search for an optimal scheduling strategy on fog nodes to minimize workload makespan, while guaranteeing high data security. In this study, we investigate the trade-off between security and makespan and propose a multiobjective optimization model to optimize them simultaneously. To achieve this goal, we design an evolutionary algorithm together with two effective measures to handle large-scale partitionable workloads: a conflict-free multi-installment scheduling (ConfMIS) strategy as well as a security criterion based on Eigenvector centrality measure. Experimental results show that the proposed algorithm can generate a set of widely spread and uniformly distributed solutions in the decision space. Among these representative solutions, system designers/decision-makers can choose their favorite one based on specific IoT application demands.

1. Introduction

Along with other emerging technologies such as AI, the IoT has been regarded as an indispensable part of the Fourth Industrial Revolution [1]. Making use of web-enabled sensors, processors, and communication hardware, the IoT touches and enables nearly every industry, including telemedicine, automated manufacturing, diverse retail, distance learning, smart home, etc. Trends indicate that IoT use is both rapidly diversifying and becoming more commonplace.

The IoT is not just about connected sensors and devices—it is about the data those devices collect from their environments and especially the powerful information derived from that data after deep processing. As the majority of IoT devices are incapable of intensive computation due to their restricted hardware and limited battery life, they generally analyze partial data locally, while sending the remaining to the cloud or fog for intensive processing. Compared to the cloud, in recent years, fog computing is becoming increasingly preferable in undertaking IoT workloads [2]. This is because cloud computing is usually situated remotely and thus usually costs excessive bandwidth usage to communicate, resulting easily in unbearable long latency [3]. By contrast, fog computing provides services of computation at the proximity of IoT sensors, so it is more convenient for fog nodes to collect data from sensors and feedback results. By taking full advantage of fog computing, the IoT gets relieved from a heavy burden of workload computation and hence saves considerable energy consumption of IoT devices. For example, wearable devices, the future of healthcare, are electronics physically worn by individuals to track and analyze biometric data from heart rate to sleep patterns. Wearable devices can notify doctors of the emergency of patients, so that they can assist patients timely. Thus, rapid data processing and useful information feedback become particularly essential in biomedical applications, while putting a major challenge considering the energy shortage of battery-powered wearable devices. Works in [4] revealed that transferring the complex and

energy-consuming machine-learning computations to fog computing can significantly reduce the energy consumption of wearable devices and speed up workload computation.

It is referred to as fog-enabled IoT in this study for the kind of IoT that offloads computational workloads on fog computing to alleviate its energy-consumption stress and minimize workload makespan. Other research available also calls it fog-aided IoT [5], fog computing empowered IoT [6], or fog supporting IoT [7]. Existing works in the domain of fog-enabled IoT mainly studied optimal workload offloading decisions or optimal task scheduling strategies for the purpose of minimizing workload makespan to meet IoT applications' real-time requirements.

Authors in literature [8] optimize workload offloading decisions by investigating the trade-off between workload makespan and energy consumption. The offloading problem is modeled as a mixed-integer nonlinear program and solved by a suboptimal algorithm based on artificial intelligent algorithms. Work [9] pointed out that the offloading decision is affected by the environment's current dynamics as well since other IoT users are also competing to maximize their resource utilization. Hence, the competition between makespan and energy consumption is modeled as a game where IoT devices' decision for the optimal distribution of tasks is captured in a joint optimization problem. Similarly, two distributed fog resource allocation algorithms are developed in reference [10] to deploy offloading solutions efficiently in cases of resource competition. Besides, the uncertainty caused by the channel measurements is considered in reference [11], based on which the offloading problem is transformed into a mixed-integer nonlinear programming problem and is solved by applying Benders decomposition to find an optimal offloading solution. Literature [12] studied how to dynamically adapt the placement of IoT applications running on the fog, depending on application needs and evolution of resource usage. In reference [13], an optimal task allocation in the fog is investigated to minimize the makespan of workloads under the constraint of IoT device's battery capacity and each workload's completion deadline. An application placement technique based on a memetic algorithm is proposed in reference [14] to minimize the execution time and energy consumption of IoT applications, and a batch application placement decision for concurrent IoT applications is obtained. In reference [15], a distance-aware, occurrence-aware, and task-property-aware volatile upper confidence bound algorithm is designed to minimize the long-term delay of task offloading. Authors in reference [16] developed an energy-optimal dynamic offloading scheme to minimize energy consumption when workloads are accomplished within a desired energy overhead and delay.

However, previously mentioned methods may suffer from the following two major limitations concerning the specific architecture of fog-enabled IoT: on the one hand, fog computing consists of a wide variety of computational devices, referred to as fog nodes, such as routers, gateways, access points, base stations, and specific fog servers. As an open structure, fog nodes are geographically distributed and widely scattered in complex network environments, while forming into an expanded attack surface, which reveals underlying security vulnerabilities of the nodes. Hence, for the sake of security, workloads should be scheduled to nodes with high security as much as possible, that is, nodes with less possibility of being attacked by hackers. However, this objective of high security may conflict with the objective of minimum makespan. To achieve the latter objective, we tend to schedule workloads on fog nodes with high performance, including high-speed computing capability and strong network connectivity. Those nodes are happened to be the preferable ones for cyber hackers. Therefore, it is necessary to study the trade-off between data security and the makespan of workloads. Unfortunately, to the best of our knowledge, no studies available have systematically explored this issue. In this study, we focus on the two conflicting objectives and propose a multiobjective optimization model to optimize them simultaneously. With this model, we design an evolutionary algorithm to obtain a set of representative solutions for decision-makers to choose according to each IoT application's practical needs.

On the other hand, IoT sensors may collect data intermittently over several periods, so the data involved in workload computation could be transferred discontinuously to the fog. If the fog layer waits until all required data finish uploaded and then starts computing, then it inevitably results in a significant waste of time. Therefore, the discontinuity feature of IoT data should be carefully considered in the design of task-scheduling strategies on the fog. With this in mind, in this study, we design a conflict-free multi-installment scheduling (ConfMIS) strategy. It distributes load fractions to fog nodes in several installments instead of solely one time, which applies to the data characteristics of IoT applications. Moreover, in order to minimize the makespan of workloads, the ConfMIS strategy compacts scheduling installments as much as possible but without time conflicts between any two installments.

The ConfMIS strategy divides the scheduling process into two parts: the internal installments and the last installment. In order to simplify the scheduling process, every internal installment is set to be uniform, that is, an identical amount of load is distributed in every installment, while the last installment is designed to make sure all fog nodes involved in workload computation finish computing at the same time. The ConfMIS strategy is different from existing periodic multi-installment scheduling (PMIS) strategies [17, 18]. As for PMIS, the total load fraction assigned to fog nodes in the last installment is forced to be identical to that in each internal installment, which may give rise to time conflicts between the last installment and the last but one. We shall demonstrate the existence of time conflicts for PMIS with an example in this study. By contrast, the ConfMIS strategy avoids time conflicts by subtly adjusting workload assignments between internal installments and the last installment.

The remainder of this study is structured as follows: in Section 2, the task-scheduling problem in a fog-enabled IoT system is formulated, with which we propose a multi-objective optimization model. In Section 3, we introduce a multi-installment scheduling strategy ConfMIS to obtain an

optimal makespan of workloads. To solve the proposed model, Section 4 presents an evolutionary algorithm based on decomposition, which is evaluated in Section 5. Finally, the study concludes in Section 6.

2. Task-Scheduling Problem on Fog-Enabled IoT

A fog-enabled IoT system composed of N heterogeneous fog nodes, denoted as $F = \{f_1, f_2, \ldots, f_N\}$, is considered, as shown in Figure 1. They are connected by physical or virtual communication links, denoted as adjacency matrix $M_{N\times N}$, where each entry m_{ij} flags whether there exists a connection between fog nodes f_i and f_j in the network. Let $\mathcal{H}$ be the size of workload transferred from IoT to the fog. It shall be divided into small load fractions and distributed to n fog nodes, denoted as $P = \{p_1, p_2, \ldots, p_n\}$, to finish computing. Certainly, not all fog nodes available are supposed to get involved in workload computation. Hence, we have $n < N$ and $P \in F$. The challenge is to determine which fog nodes should be selected to accomplish workload computation (node selection scheme) and how to partition the workload into load fractions and efficiently distribute them on the selected fog nodes (load scheduling scheme), so that the data security S and the makespan T of the workload could strike a trade-off (be optimized to the maximum).

In this section, we first propose a multi-installment scheduling strategy ConfMIS to obtain the makespan T of the workload and then introduce a security criterion based on Eigenvector centrality to show how data security S is measured in this study. Based on this, a multiobjective optimization model is given at the end of this section.

2.1. Makespan **T**. Figure 2 illustrates the scheduling process of ConfMIS strategy. As can be seen, there are m installments in total, which are divided into two parts: $(m-1)$ identical internal installments and the last installment. In each internal installment, an identical amount of load V is distributed and completed among n fog nodes. The last installment differs from internal ones because it is responsible for guaranteeing that all nodes stop computing at the same time, so that the makespan could be minimized. The last installment completes hV size of workload, where h is a real number satisfying $h > 1$. Since $(m-1)V + hV = \mathcal{H}$, we have $V = \mathcal{H}/(m+h-1)$.

The master sends chunk $\alpha_i V$ and $\beta_i V$ to fog node p_i with $i = 1, 2, \ldots, n$ in each internal installment and the last installment, respectively. We have $\alpha_i > 0$, $\beta_i > 0$, and $\sum_{i=1}^{n} \alpha_i = \sum_{i=1}^{n} \beta_i = 1$. It takes the master time $o_i + z_i \alpha_i V$ to transmit chunk $\alpha_i V$ to node p_i and costs node p_i time $s_i + w_i \alpha_i V$ to finish computing this chunk, where o_i and s_i are communication and computation start-up overheads, respectively, while w_i represents the ratio of the time taken by node p_i to compute a given workload to that by a standard fog node, and z_i is the reciprocal of network transmission speed between the master and fog node p_i. The overall makespan is determined by the last fog node to finish its assigned chunk, which is equal to the finish time of the last

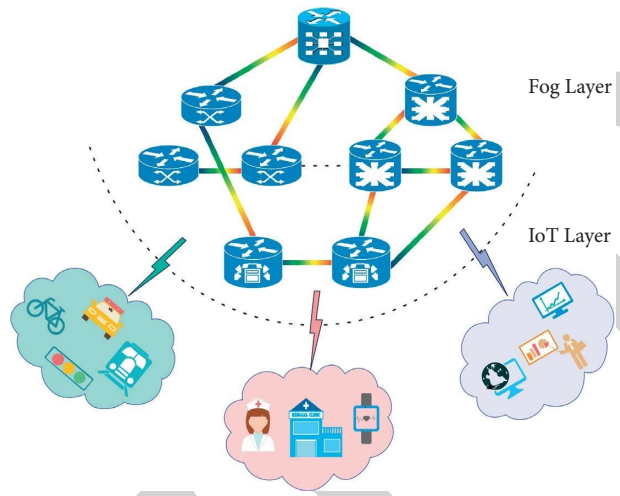

Figure 1: Hierarchical network of fog-enabled IoT.

installment since all fog nodes stop computing at the same instant. By observing Figure 2, we can obtain the function of makespan T as follows:

$$T(P) = (m-1)\left(\alpha_1 \frac{\mathcal{H}}{m+h-1} w_1 + s_1\right) + o_1 + \beta_1 \frac{h\mathcal{H}}{m+h-1} w_1 + s_1. \tag{1}$$

The makespan T depends on the node selection scheme and the load scheduling scheme. The former determines which fog nodes are selected to participate in workload computation, while the latter is an optimal solution to scheduling strategy ConfMIS. This solution includes optimal load partitions for internal installments and the last installment, an optimal number of installments, and a feasible value for h, which represents how large proportion of workloads the last installment completes compared to each internal installment. We shall further demonstrate in Section 3 that given a node selection scheme P, the load scheduling scheme (an optimal solution to ConfMIS strategy) can be obtained. Hence, the function of makespan T involves only one set of variables, that is node selection scheme P.

2.2. Security **S**. Since fog nodes are heterogeneous and widely scattered in a complex network environment, they form into an expanded attack surface, which reveals underlying security vulnerabilities. Hence, for the sake of data security, workloads ought to be primarily scheduled on fog nodes with high security, that is, nodes with less possibility of being attacked by hackers. To measure the vulnerability of fog nodes, we introduce the concept of node centrality.

Node centrality is the term used to describe how important a node is within a network. In the context of security, it means how favorable a node is for hackers. The node centrality is often calculated by simply counting the number of links in and out of each node in the network (degree centrality), by measuring how short the shortest paths are from one node to the others (closeness centrality) or calculating how often a node lies on the shortest path between

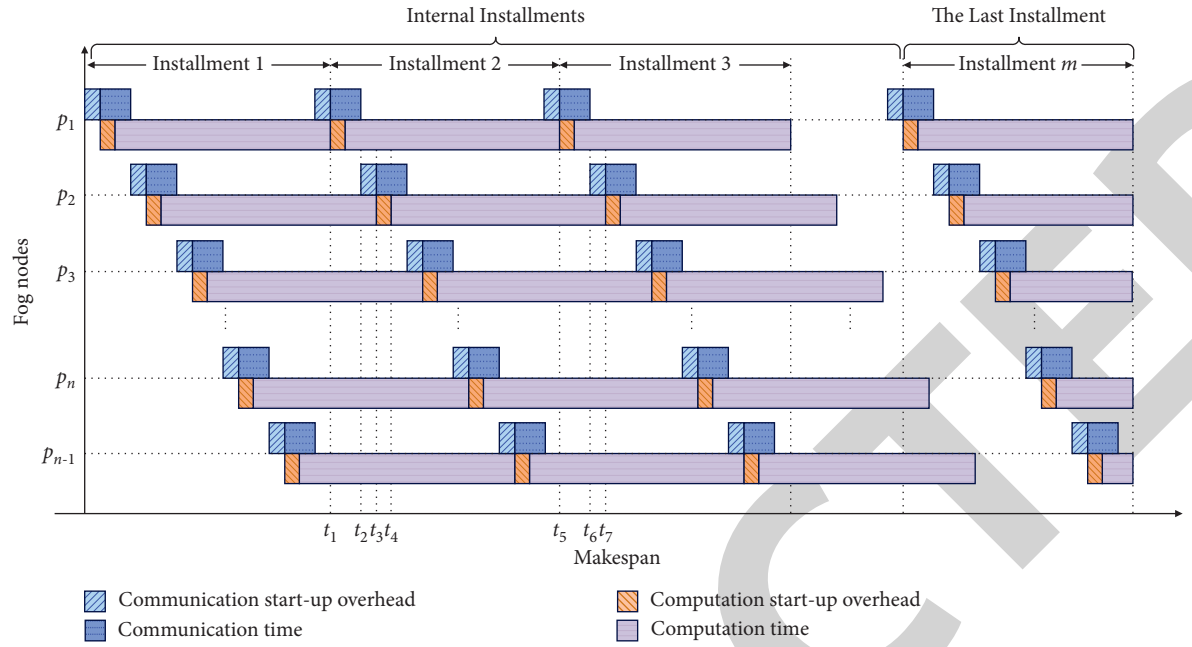

Figure 2: The Gantt chart of ConfMIS.

other nodes (betweenness centrality). These centrality measures are useful in some cases, but they rely heavily on the degree of nodes and do not consider each node's relationship with other important nodes. For example, if one fog node with high centrality in the network has a high possibility of getting attacked, then nodes in its neighborhood should also be ranked with a relatively high centrality value due to the continuity of attacks. Eigenvector centrality is a measure of the influence of a node in a network. It holds the idea that connections to high-scoring nodes contribute more to the score of the node than that to low-scoring nodes.

The Eigenvector centrality of a fog node f_i is given by the following equation:

$$c_i = v(f_i), \tag{2}$$

where vector v is the eigenvector of unit length corresponding to the largest eigenvalue of the adjacency matrix $M_{N\times N}$ of the network and $v(f_i)$ is the i-th entry of vector v. Hence, one could determine the Eigenvector centrality of every fog node by just computing this eigenvector v.

Figure 3 shows a network that contains two local substructures—a hierarchy and a core-periphery, with one connection linking them. We calculate the degree centrality, betweenness centrality, closeness centrality, and Eigenvector centrality and compare them in Table 1. As can be seen, node 8 has the highest Eigenvector centrality value, followed by nodes 9 and 10. Although node 1 connects the complete hierarchy of the left subnetwork, its Eigenvector centrality is smaller than node 8, which connects the right subnetwork. In comparison to the degree centrality, in which nodes 8 and 9 have equal degree centrality values, here since node 8 is connected to nodes that are "more connected" and thus

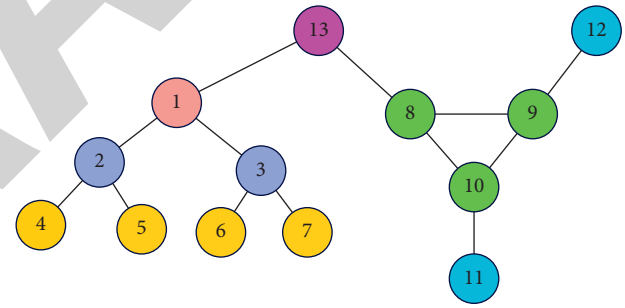

Figure 3: An example of Eigenvector centrality.

more important, and since node 9 is connected to isolated node 12, node 8 has a higher Eigenvector centrality value. As for the betweenness centrality and closeness centrality, node 13 is considered more important than node 8. This is a result of the fact that all the paths between the left and the right subnetworks must pass through node 13.

Assume that only n fog nodes $P = \{p_1, p_2, \ldots, p_n\}$ are selected to take part in workload computation, we define the security of node selection scheme P as follows:

$$\begin{aligned} S(P) &= \frac{n}{\sum_{i=1}^{n}\{c_j | p_i = f_j, j \in [1, N]\}} \\ &= \frac{n}{\sum_{i=1}^{n} v(p_i)}. \end{aligned} \tag{3}$$

2.3. Multiobjective Optimization Model. Based on the above analysis on security and makespan functions, we propose a multiobjective optimization model as follows:

Table 1: Comparison among different centrality measures.

Node	Degree centrality	Betweenness centrality	Closeness centrality	Eigenvector centrality
1	3	0.255682	0.444444	0.239731
2	3	0.119318	0.352941	0.144301
3	3	0.119318	0.352941	0.144301
4	1	0.000000	0.266667	0.058403
5	1	0.000000	0.266667	0.058403
6	1	0.000000	0.266667	0.058403
7	1	0.000000	0.266667	0.058403
8	3	0.181818	0.387097	0.510702
9	3	0.062500	0.315789	0.479057
10	3	0.062500	0.315789	0.479057
11	1	0.000000	0.244898	0.193888
12	1	0.000000	0.244898	0.193888
13	2	0.198864	0.428571	0.303722

$$\begin{cases} \min_{P} T(P) = (m-1)\left(\alpha_1 \dfrac{\mathscr{H}}{m+h-1} w_1 + s_1\right) \\ \quad + o_1 + \beta_1 \dfrac{h\mathscr{H}}{m+h-1} w_1 + s_1, \\ \min_{P} \dfrac{1}{S(P)} = \dfrac{1}{n} \sum_{i=1}^{n} v(p_i). \end{cases} \tag{4}$$

Here, $P = \{p_1, p_2, \ldots, p_n\} \in F = \{f_1, f_2, \ldots, f_N\}$.

There are two objectives considered in our model, where function $T(P)$ refers to makespan minimization, while function $1/S(P)$ represents the reciprocal of security maximization. This model optimizes data security and makespan simultaneously. It involves only one set of variables, that is, the node selection scheme P. This is because once P is given, an optimal workload scheduling scheme can be obtained by ConfMIS strategy. We shall illustrate in the next section how ConfMIS strategy works and how to derive its solution, including optimal load partitions α_1 and β_1, optimal number m of installments, and a feasible value for h, which are involved in the proposed model.

3. Conflict-free Multi-Installment Scheduling Strategy

In this section, we first derive closed-form solutions to an optimal load partition, and then by rigorous proof of two theorems, we derive a closed-form solution to an optimal number of installments. Subsequently, we demonstrate by an example that when $h = 1$ (as adopted in the existing studies on PMIS [17, 18]), there exist time conflicts between the last installment and the last but one. To avoid these time conflicts, we derive a feasible value for h at the end of this section.

3.1. Optimal Load Partition. To achieve a minimum makespan, there is no idle time between any two adjacent internal installments for each fog node. The time period from t_1 to t_5 in Figure 2 is taken as an example. It costs fog node p_1 time $(\alpha_1 V w_1 + s_1)$ to compute load fraction $\alpha_1 V$ in the second installment. Meanwhile, the time period from t_1 to t_5 equals the time period from t_1 to t_2 (communication time $z_1\alpha_1 V$ for node p_1 in the second installment) plus time period from t_2 to t_3 (communication start-up overhead o_2 for node p_2 in the second installment) plus time period from t_3 to t_4 (computation start-up overhead s_2 for node p_2) plus time period from t_4 to t_7 (computation time $w_2\alpha_2 V$ for node p_2 in the second installment) minus time period from t_6 to t_7 (communication start-up overhead o_2 for node p_2 in the third installment) minus time period from t_5 to t_6 (communication time $z_1\alpha_1 V$ for node p_1 in the third installment). Hence, we have $\alpha_1 V w_1 + s_1 = z_1\alpha_1 V + o_2 + s_2 + \alpha_2 V w_2 - o_2 - z_1\alpha_1 V$. That is, $\alpha_1 V w_1 + s_1 = s_2 + \alpha_2 V w_2$. Similarly, we arrive at the following equations:

$$\begin{aligned} \alpha_1 V w_1 + s_1 &= \alpha_2 V w_2 + s_2 \\ &= \cdots = \alpha_n V w_n + s_n. \end{aligned} \tag{5}$$

Thus, α_i with $i = 1, 2, \ldots, n$ can be written in terms of α_1 as follows:

$$\alpha_i = \frac{\alpha_1 V w_1 + s_1 - s_i}{V w_i}. \tag{6}$$

Let

$$\begin{aligned} \Delta_i &= \frac{w_1}{w_i}, \\ \Phi_i &= \frac{s_1 - s_i}{w_i}. \end{aligned} \tag{7}$$

Simplifying equation (6) by equation (7), we have

$$\begin{aligned} \alpha_i &= \alpha_1 \Phi_i + \frac{1}{V}\Phi_i, \\ i &= 2, 3, \ldots, n. \end{aligned} \tag{8}$$

Combining equation (8) with $\sum_{i=1}^{n} \alpha_i = 1$, we can obtain

$$\frac{1}{V}\sum_{i=2}^{n} \Phi_i + \left(1 + \sum_{i=2}^{n} \Delta_i\right)\alpha_1 = 1. \tag{9}$$

Rearranging equations (8) and (9) yields the closed-form solutions to an optimal load partition for internal installments:

$$\begin{cases} \alpha_1 = \dfrac{1 - 1/V \sum_{i=2}^{n} \Phi_i}{1 + \sum_{i=2}^{n} \Delta_i}, \\ \alpha_i = \dfrac{1}{V}\Phi_i + \Delta_i \alpha_1, i = 2, 3, \ldots, n. \end{cases} \tag{10}$$

A necessary condition for a feasible load partition is given by $\alpha_i > 0$ with $i = 1, 2, \ldots, n$.

Likewise, to achieve the minimum makespan, all fog nodes stop computing at the same time in the last installment. By observing Figure 2, we have the following equations:

$$\begin{aligned} s_i + w_i \beta_i hV = o_{i+1} + z_i \beta_i hV + s_{i+1} + w_{i+1} \beta_{i+1} hV, \\ i = 1, 2, \ldots, n-1. \end{aligned} \tag{11}$$

One can express β_{i+1} with $i = 1, \ldots, n-1$ in terms of β_i by rearranging equation (11) as follows:

$$\beta_{i+1} = \frac{s_i - (o_{i+1} + s_{i+1})}{hVw_{i+1}} + \frac{w_i - z_i}{w_{i+1}} \beta_i. \tag{12}$$

Let

$$\begin{aligned} \delta_{i+1} &= \frac{s_i - (o_{i+1} + s_{i+1})}{w_{i+1}}, \\ \varepsilon_{i+1} &= \frac{w_i - z_i}{w_{i+1}}. \end{aligned} \tag{13}$$

Simplifying equation (12) by equation (13) yields

$$\begin{aligned} \beta_{i+1} = \frac{1}{hV} \delta_{i+1} + \varepsilon_{i+1} \beta_i, \\ i = 1, 2, \ldots, n-1. \end{aligned} \tag{14}$$

Denoting

$$\mathrm{E}_i = \prod_{j=2}^{i} \varepsilon_j \text{ and } \Gamma_i = \sum_{j=2}^{i} \left(\delta_j \prod_{k=j+1}^{i} \varepsilon_k \right). \tag{15}$$

By means of equation (15), β_i can be written in terms of β_1 from recursive derivation of equation (14) as follows:

$$\begin{aligned} \beta_i = E_i \beta_1 + \frac{1}{hV} \Gamma_i, \\ i = 2, 3, \ldots, n. \end{aligned} \tag{16}$$

Together equation (16) with $\sum_{i=1}^{n} \beta_i = 1$, closed-form solutions to an optimal load partition can be obtained for the last installment:

$$\begin{cases} \beta_1 = \left(1 - \dfrac{1}{hV} \sum_{i=2}^{n} \Gamma_i \right) \Big/ \left(1 + \sum_{i=2}^{n} E_i \right), \\ \beta_i = \dfrac{1}{hV} \Gamma_i + E_i \beta_1, i = 2, 3, \ldots, n. \end{cases} \tag{17}$$

It is worth noting that $\beta_i > 0$ with $i = 1, 2, \ldots, n$ should hold to ensure a feasible load partition.

3.2. Optimal Number of Installments. Through the above analysis, we know that given the values of m and h, an optimal load partition can be obtained by equations (10) and (17). Note that not all values of $m\,(m > 0)$ and $h\,(h > 0)$, denoted as (m, h), lead to a feasible load scheduling scheme since $\alpha_i > 0$ and $\beta_i > 0$ with $i = 1, 2, \ldots, n$ should be guaranteed in the first place. We regard (m, h) as a feasible solution to ConfMIS strategy if the load partitions obtained based on (m, h) satisfy the above constraints; otherwise, (m, h) is an infeasible solution.

Theorem 1. *If (m, h) is a feasible solution to ConfMIS strategy, then (m', h) with $m' < m$ is also a feasible solution.*

Proof. Since (m, h) is a feasible solution to ConfMIS strategy, load partitions α_i and β_i obtained by equations (10) and (17) satisfy that $\alpha_i > 0$ and $\beta_i > 0$ for $\forall i \in \{1, 2, \ldots, n\}$. When $i = 1$, we have

$$\begin{cases} \alpha_1 = \dfrac{1 - [(m + h - 1)/\mathscr{H}] \sum_{i=2}^{n} \Phi_i}{1 + \sum_{i=2}^{n} \Delta_i} > 0, \\ \beta_1 = \dfrac{1 - [(m + h - 1)/h\mathscr{H}] \sum_{i=2}^{n} \Gamma_i}{1 + \sum_{i=2}^{n} \mathrm{E}_i} > 0. \end{cases} \tag{18}$$

Hence,

$$\left\{ \sum_{i=2}^{n} \Phi_i < \frac{\mathscr{H}}{m + h - 1}, \sum_{i=2}^{n} \Gamma_i < \frac{h\mathscr{H}}{m + h - 1}. \right. \tag{19}$$

As for solution $(m\prime, h)$ with $m\prime = m - 1$, one can also obtain load partitions α_i' and β_i' via equations (10) and (17). For α_i' and β_i', we have

$$m \begin{cases} \alpha_1' = \dfrac{1 - [(m + h - 2)/\mathscr{H}] \sum_{i=2}^{n} \Phi_i}{1 + \sum_{i=2}^{n} \Delta_i} > \dfrac{1 - [(m + h - 2)/\mathscr{H}]\,[\mathscr{H}/(m + h - 1)]}{1 + \sum_{i=2}^{n} \Delta_i} > 0, \\ \beta_1' = \dfrac{1 - m + h - 2/h\mathscr{H} \sum_{i=2}^{n} \Gamma_i}{1 + \sum_{i=2}^{n} E_i} > \dfrac{1 - [(m + h - 2)/h\mathscr{H}]\,[h\mathscr{H}/(m + h - 1)]}{1 + \sum_{i=2}^{n} E_i} > 0. \end{cases} \tag{20}$$

Given that $\alpha_i > 0$ and $\beta_i > 0$ with $i = 1, 2, \ldots, n$, we have proved in the above that $\alpha_1' > 0$ and $\beta_1' > 0$. Then, we proceed to prove that $\alpha_i' > 0$ and $\beta_i' > 0$ for $i = 2, 3, \ldots, n$ by contradiction. Assume that $\exists j \in \{2, 3, \ldots, n\}$, $\alpha_j' < 0$, that is, $\alpha_j' = \Phi_j/V + \Delta_j \alpha_1' < 0$ according to equation (8). Substituting $V = \mathscr{H}/(m + h - 2)$ into α_j', we have $\Phi_j < -\Delta_j \alpha_1' \mathscr{H}/(m + h - 2)$. Since $\alpha_j > 0$, we have $\alpha_j = \Delta_j \alpha_1 + (1/V)\Phi_j > 0$. Likewise, substituting V by $\mathscr{H}/(m + h - 2)$ yields $\Phi_j > -(\Delta_j \alpha_1 \mathscr{H})/(m + h - 1)$. Therefore, we arrive at $-(\Delta_j \alpha_1 \mathscr{H})/(m + h - 1) < \Phi_j < -(\Delta_j \alpha_1' \mathscr{H})/(m + h - 2)$. After simplifying this, one can obtain that $\alpha_1 (m + h - 2) > \alpha_1' (m + h - 1)$. Expanding α_1 and α_1', we have

$$(m+h-2)-\frac{(m+h-1)(m+h-2)}{\mathscr{H}}\sum_{i=2}^{n}\Phi_i>(m+h-1)-\frac{(m+h-1)(m+h-2)}{\mathscr{H}}\sum_{i=2}^{n}\Phi_i. \tag{21}$$

Since $m+h-2>m+h-1$, a contradiction occurs. Hence, the initial assumption that $\exists j\in\{2,3,\ldots,n\}$, $\alpha_j'<0$ must be false. In the same way, one can prove that $\beta_i'>0$. To summarize, when (m,h) is a feasible solution to ConfMIS strategy, that is $\alpha_i>0$ and $\beta_i>0$ with $i=1,2,\ldots,n$, $(m\prime,h)$ with $m\prime=m-1$ is also a feasible solution to ConfMIS strategy since $\alpha_i'>0$ and $\beta_i'>0$. By mathematical induction, we can conclude that if (m,h) is a feasible solution to ConfMIS strategy, then $(m\prime,h)$ with $m\prime<m$ is also a feasible solution. □

Theorem 2. *Consider all feasible solutions (m,h) under a certain node selection scheme P, as number m of installments increases, the makespan T first decreases and then increases.*

Proof. For a certain node selection scheme P, one can implement ConfMIS strategy with feasible values of (m,h) on the selected fog nodes and further obtain its corresponding makespan T by equation (1). Substituting V for $\mathscr{H}/(m+h-1)$ into equation (1) and expanding α_1 and β_1 by equations (10) and (17), we have

$$T(m,h)=(m-1)\left(\frac{1-(1/V)\sum_{i=2}^{n}\Phi_i}{1+\sum_{i=2}^{n}\Delta_i}w_1V+s_1\right)+\frac{1-(1/hV)\sum_{i=2}^{n}\Gamma_i}{1+\sum_{i=2}^{n}E_i}hVw_1+o_1+s_1. \tag{22}$$

Denoting

$$a=\sum_{i=2}^{n}\Phi_i,\quad b=1+\sum_{i=2}^{n}\Delta_i,\quad c=\sum_{i=2}^{n}\Gamma_i,\quad d=1+\sum_{i=2}^{n}E_i. \tag{23}$$

Then, $T(m,h)$ can be simplified as follows:

$$T(m,h)=\left(\frac{(m-1)\mathscr{H}}{b(m+h-1)}-\frac{(m-1)a}{b}\right)w_1+\left(\frac{h\mathscr{H}}{d(m+h-1)}-\frac{c}{d}\right)w_1+o_1+ms_1. \tag{24}$$

Dividing both sides of the equation by w_1 yields

$$\frac{T(m,h)}{w_1}=\frac{(m-1)\mathscr{H}}{b(m+h-1)}-\frac{(m-1)a}{b}+\frac{h\mathscr{H}}{d(m+h-1)}-\frac{c}{d}+m\frac{s_1}{w_1}+\frac{o_1}{w_1}. \tag{25}$$

Suppose that $(m+1,h)$ is a feasible solution to ConfMIS strategy, then (m,h) and $(m-1,h)$ are also feasible solutions according to Theorem 1. Similarly, $T(m+1,h)/w_1$ and $T(m-1,h)/w_1$ can be obtained by equation (25). Comparing $T(m,h)/w_1$ with them, we have

$$\frac{T(m+1,h)}{w_1}-\frac{T(m,h)}{w_1}=\frac{\mathscr{H}h(d-b)}{b\,d(m+h)(m+h-1)}+\frac{s_1}{w_1}-\frac{a}{b}, \tag{26}$$

$$\frac{T(m,h)}{w_1}-\frac{T(m-1,h)}{w_1}=\frac{\mathscr{H}h(d-b)}{b\,d(m+h-1)(m+h-2)}+\frac{s_1}{w_1}-\frac{a}{b}. \tag{27}$$

Let

$$\lambda=\frac{s_1}{w_1}-\frac{a}{b}. \tag{28}$$

Equations (26) and (27) can be rewritten as follows:

$$\frac{T(m+1,h)}{w_1}-\frac{T(m,h)}{w_1}=\frac{\mathscr{H}h(d-b)}{b\,d(m+h)(m+h-1)}+\lambda, \tag{29}$$

$$\frac{T(m,h)}{w_1}-\frac{T(m-1,h)}{w_1}=\frac{\mathscr{H}h(d-b)}{b\,d(m+h-1)(m+h-2)}+\lambda. \tag{30}$$

According to the definitions of a, b, and d in equation (23), we have

$$\lambda=\frac{s_1}{\omega_1}-\frac{\sum_{i=1}^{n}[(s_1-s_i)/w_i]}{\sum_{i=1}^{n}w_1/w_i}=\frac{\sum_{i=1}^{n}(s_i/w_i)}{\sum_{i=1}^{n}\Delta_i}>0, \tag{31}$$

and

$$b-d=\sum_{i=2}^{n}(\Delta_i-E_i)=\sum_{i=2}^{n}\left(\frac{w_1}{w_i}-\prod_{j=2}^{i}\frac{w_{j-1}-z_{j-1}}{w_j}\right)=\sum_{i=2}^{n}\left(w_1\prod_{j=2}^{i-1}w_j\prod_{j=2}^{i}\frac{1}{w_j}-\prod_{j=2}^{i}(w_{j-1}-z_{j-1})\prod_{j=2}^{i}\frac{1}{w_j}\right)=\sum_{i=2}^{n}\left(w_1\prod_{j=2}^{i-1}w_j-(w_1-z_1)\prod_{j=2}^{i-1}(w_j-z_j)\right)>0. \tag{32}$$

From equations (31) and (32), we know that $\lambda>0$ and $b>d$. Thus, it can be inferred from equation (29) that $T(m+1)>T(m)$ when $(m+h)(m+h-1)>\mathscr{H}h(b-d)/(b\,d\lambda)$. By solving this inequality for m, we can conclude that the

makespan T increases as m increases when $m < [(3/2) - h + \sqrt{h\mathscr{H}(b-d)/(b\ d\backslash\lambda) + 1/4}]$. Similarly, by analyzing equation (30), one can claim that the makespan T decreases as m increases when $m > [(1/2) - h + \sqrt{h\mathscr{H}(b-d)/(b\ d\backslash\lambda) + 1/4}]$. In summary, the makespan first decreases and then increases as the number of installments increases. □

Theorem 3. *This reveals a fact that there exists a knee point for the makespan T with respect to m. This knee point satisfies the bounds on m given by* $1/2 - h + \sqrt{h\mathscr{H}(b-d)/(b\,d\backslash\lambda) + 1/4} < m < 3/2 - h + \sqrt{h\mathscr{H}(b-d)/(b\,d\backslash\lambda) + 1/4}$. *The upper bound of m is exactly 1 larger than the lower bound, so that the makespan T reaches its minimum when*

$$m = \frac{3}{2} - h + \sqrt{h\mathscr{H}(b-d)/(b\ d\backslash\lambda) + \frac{1}{4}}. \tag{33}$$

Equation (31) gives an optimal solution to the number of installments.

3.3. Time Conflict When h = 1. In the existing studies on PMIS [17, 18], the total load fraction assigned to fog nodes in the last installment is forced to be identical to that in each internal installment. This can be considered as a special case in our scheduling model as $h = 1$. Then, we demonstrate by an example that PMIS may give rise to time conflicts between the last installment and the last but one.

Three heterogeneous fog nodes scheduled in three installments are considered, that is, $n = m = 3$. For simplicity of calculation, workload size is set as $\mathscr{H} = 270$. Relevant parameters of the three fog nodes are listed as follows:

$$\begin{aligned} &o_1 = 2, s_1 = 3, z_1 = 0.3, w_1 = 10, \\ &o_2 = 1, s_2 = 2, z_2 = 1.5, w_2 = 30. \\ &o_3 = 3, s_3 = 1, z_3 = 1.8, w_3 = 50. \end{aligned} \tag{34}$$

The scheduling sequence of the three fog nodes follows the increasing order of z_i, that is, $P = (p_1, p_2, p_3)$. According to equations (10) and (17), we obtain an optimal load partition as follows:

$$\begin{aligned} &\alpha_1 \approx 0.652, \alpha_2 \approx 0.217, \alpha_3 \approx 0.131, \\ &\beta_1 \approx 0.664, \beta_2 \approx 0.214, \beta_3 \approx 0.122 \end{aligned} \tag{35}$$

The start time T_b^i of arbitrary fog node p_i in the last installment can be calculated by

$$T_b^i = \sum_{j=1}^{i-1}\left(o_j + z_j\beta_j hV\right) + o_i + (m-1)(s_1 + w_1\alpha_1 V). \tag{36}$$

Meanwhile, the finish time T_e^i of p_i in the last installment can be computed by

$$T_e^i = \sum_{j=1}^{i-1}\left(o_j + z_j\alpha_j V\right) + o_i + (m-1)(s_i + w_i\alpha_i V). \tag{37}$$

Here, we have the start time T_b^3 and finish time T_e^3 of fog node p_3 as follows:

$$\begin{cases} T_b^3 = \sum_{j=1}^{2}\left(o_j + z_j\beta_j hV\right) + o_3 + 2(s_1 + w_1\alpha_1 V) \approx 1231.84, \\ T_e^3 = \sum_{j=1}^{2}\left(o_j + z_j\alpha_j V\right) + o_3 + 2(s_3 + w_3\alpha_3 V) \approx 1231.93. \end{cases} \tag{38}$$

It is noted that $T_e^3 > T_b^3$, which means there is a time conflict for node p_3 between the last installment and second to the last. If such time conflicts are allowed in the scheduling, then workload cannot be completed as planned. Therefore, it is imperative to search for a feasible scheduling scheme without time conflicts. We shall fulfill this goal by finding a feasible value for h.

3.4. Feasible Value for h. To avoid time conflicts, the start time T_b^i of arbitrary fog node p_i in the last installment should be greater than or equal to its finish time T_e^i in the last installment but one. Combining equations (36) and (37), we have

$$\begin{aligned} T_b^i - T_e^i = &\sum_{j=1}^{i-1}\left(o_j + z_j\beta_j hV\right) + o_i + (m-1)(s_1 + w_1\alpha_1 V) \\ &- \sum_{j=1}^{i-1}\left(o_j + z_j\alpha_j V\right) - o_i - (m-1)(s_i + w_i\alpha_i V). \end{aligned} \tag{39}$$

According to equation (5), $s_1 + w_1\alpha_1 V = s_i + w_i\alpha_i V$. Thus, we have

$$\begin{aligned} T_b^i - T_e^i &= \sum_{j=1}^{i-1}\left(o_j + z_j\beta_j hV\right) - \sum_{j=1}^{i-1}\left(o_j + z_j\alpha_j V\right) \\ &= V\sum_{j=1}^{i-1} z_j\left(\beta_j h - \alpha_j\right). \end{aligned} \tag{40}$$

Expanding α_j and β_j according to equations (10) and (17) yields

$$T_b^i - T_e^i = \sum_{j=1}^{i-1} z_j\left[\left(\Gamma_j + E_j\beta_1 hV\right) - \left(\Phi_j + \Delta_j\alpha_1 V\right)\right] \tag{41}$$

It is considered that both communication and computation start-up overheads are nearly neglectable compared to considerable communication time and computation time. Rearranging equation (41) by deleting start-up overheads yields

$$T_b^i - T_e^i \approx V\sum_{j=1}^{i-1} z_j\left(E_j\beta_1 h - \Delta_j\alpha_1\right). \tag{42}$$

Let $T_b^i - T_e^i \geq 0$, we have $E_j\beta_1 h - \Delta_j\alpha_1 \geq 0$ with $\forall j \in [1, i-1]$. Hence, $E_j\beta_1 h/\Delta_j\alpha_1 \geq 1$. Comparing equations (10) and (17), it is easy to find that $\beta_1 > \alpha_1$. Therefore,

$$\frac{E_j\beta_1 h}{\Delta_j\alpha_1} > \frac{hE_j}{\Delta_j} = h\frac{\prod_{k=1}^{j-1}(w_k - z_k)}{\prod_{k=2}^{j} w_k \dfrac{w_k}{w_1}} = h\prod_{k=1}^{j-1}\left(1 - \frac{z_k}{w_k}\right), j = 2, 3, \ldots, i-1. \tag{43}$$

One can observe from equation (43) that if $j = i-1$ and $h\prod_{k=1}^{j-1}(1 - z_k/w_k) = 1$, then for $\forall j = 2, 3, \ldots, i-2$, $h\prod_{k=1}^{j-1}(1 - z_k/w_k) > 1$ must be true. Hence, we have $E_j\beta_1 h/\Delta_j\alpha_1 > hE_j/\Delta_j \geq 1$ hold for equation (43). Consequently, $T_b^i - T_e^i > 0$ holds for equation (42), that is, no time conflicts exist. Therefore, from equation $h\prod_{k=1}^{j-1}(1 - z_k/w_k) = 1$ with $j = i-1$, one can find a feasible value for h as follows:

$$h = \frac{1}{\prod_{k=1}^{i-2}(1 - z_k/w_k).} \tag{44}$$

In summary, given a certain node selection scheme P, one can first calculate an optimal number m of installments by equation (33) and then find a feasible value for h by equation (44). Taking the values of m and h from equations (10) and (17), one can get an optimal load partition for ConfMIS strategy. Hence, as mentioned in the earlier, the makespan T of workloads on the fog only determines by node selection scheme P.

4. Evolutionary Algorithm Based on Decomposition

In this section, we put forward an evolutionary algorithm based on decomposition to solve the proposed model in Section 2. Since the objectives of security and makespan contradict each other, the best trade-offs between security and makespan can be stated by Pareto optimality [19]. The pareto optimality is the state at which objectives are optimized in a way that one objective cannot improve without a second worsening. The goal of the evolutionary algorithm is to obtain an approximation to the pareto front (PF), from which a system designer or decision-maker can choose an appropriate solution based on IoT application demands.

As there is only one set of variables involved in the proposed model, we encode individuals directly as the node selection scheme P, which is a sequence of fog nodes selected from the fog. For example, suppose there are $N = 200$ fog nodes in total, from which an IoT application requires $n = 10$ fog nodes to participate in workload computation. A possible individual could be encoded as $P = (p_1, p_2, \ldots, p_{10}) = (5, 184, 30, 42, 107, 44, 2, 97, 143, 65)$, where $p_i \in [1, N] = [1, 200]$.

To obtain a set of uniformly distributed solutions, we decompose the two objective functions into K subproblems via the Tchebycheff approach and optimize them simultaneously. Let $\{\gamma^1, \gamma^2, \ldots, \gamma^K\}$ be a set of evenly spread weight vectors in the objective space, where $\gamma^i = \{\gamma_1^i, \gamma_2^i\}$. Each weight vector γ^i has a neighborhood of V closest weight vectors, denoted as set B^i, which can be obtained by calculating the Euclidean distances between any two weight vectors in $\{\gamma^1, \gamma^2, \ldots, \gamma^K\}$. Let $r^b = (r_1^b, r_2^b)$ and $r^w = (r_1^w, r_2^w)$ be the reference points, where r^b records the best values (the minimum) of the two objectives $T(P)$ and $1/S(P)$, while r^w corresponds to the worst values (the maximum) found so far. Each subproblem can be described as follows:

$$g_j(P) = \max\left\{\gamma_1^j \times \frac{\left|T(P) - r_1^b\right|}{r_1^w - r_1^b}, \gamma_2^j \times \frac{\left|1/S(P) - r_2^b\right|}{r_2^w - r_2^b}\right\} \text{where } j = 1, 2, \ldots, K. \tag{45}$$

The evolutionary algorithm works as follows:

Step 1. *Initialization*: set external population $EP = \varnothing$, which is used to contain nondominated solutions found so far. Generate an initial population $POP = \{P^1, \ldots, P^K\}$, where $P^i = (p_1^i, p_2^i, \ldots, p_n^i)$ with $i = 1, \ldots, K$ satisfies $p_j^i \in [1, N]$ and $\forall j, k \in [1, n], p_j^i \neq p_k^i$. Compute each individual's objective functions $T(P^i)$ and $1/S(P^i)$ according to equations (1) and (3). Initialize $r^b = (r_1^b, r_2^b)$ and $r^w = (r_1^w, r_2^w)$, respectively, based on current population POP.

Step 2. *Evolution*: for $i = 1, 2, \ldots, K$, do

Step 2.1. *Crossover*: randomly select two indexes $a, b \in [1, V]$ from set B^i and then apply crossover operator on individuals P^a and P^b to generate offspring O^1 and O^2.

Step 2.2. *Mutation*: apply mutation operator on O^1 and O^2 according to mutation probability p_{mut} to generate offspring O^3 and O^4. Compute the objective functions of O^3 and O^4 according to equations (1) and (3).

Step 2.3. *Update Reference Points r^b and r^w*: for $k = 3, 4$, if $T(O^k) < r_1^b$, set $r_1^b = T(O^k)$; if $1/S(O^k) < r_2^b$, set $r_2^b = 1/S(O^k)$; if $T(O^k) > r_1^w$, set $r_1^w = T(O^k)$; if $1/S(O^k) > r_2^w$, set $r_2^w = 1/S(O^k)$.

Step 2.4. *Update Neighbors*: for each index $j \in B^i$ and $k = 3, 4$, if $g_j(O^k) < g_j(P^j)$, set $P^j = O^k$.

Step 2.4. *Update EP*: remove all solutions in EP dominated by point $(T(O^3), 1/S(O^3))$ and $(T(O^4), 1/S(O^4))$ in the objective space. Meanwhile, add O^3 to EP if no solutions in EP dominate it, so does O^4.

Step 3. *Stopping Criteria*: if Step 2 iterates more than times, then stop and output ; otherwise, go to Step 2.

In Step 2.1, a crossover operator is adopted on individuals P^a and P^b to generate offsprings O^1 and O^2. The crossover operator works as follows:

Step 1. *Exchange Genes*: create two random crossover points in the parents and copy the genes between them from the first parent to the second offspring while the second parent to the first offspring.

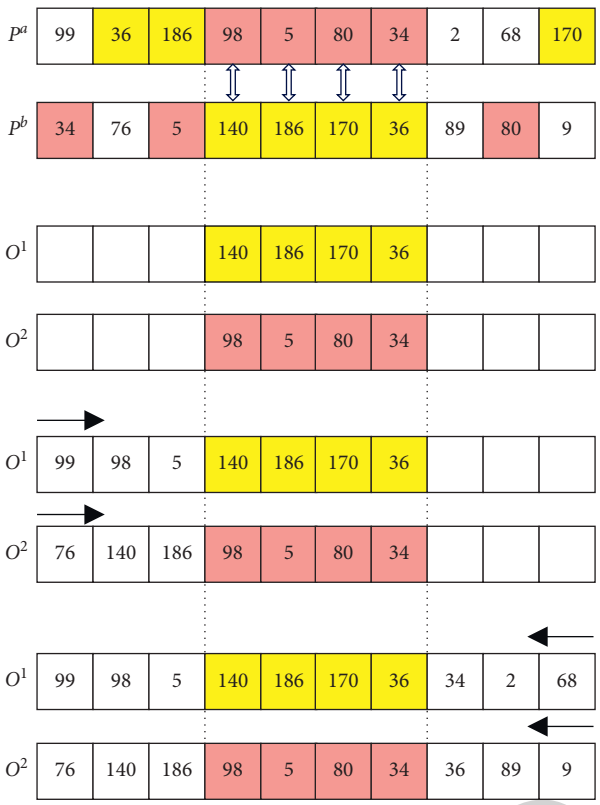

Figure 4: An example of crossover operator.

Step 2. *Copy from Head to Tail*: starting from the first gene in the parents until the first crossover point, copy the remaining unused genes from the first parent to the first offspring while the second parent to the second offspring.

Step 3. *Copy from Tail to Head*: starting from the last gene in the parents until the second crossover point, copy the remaining unused genes from the first parent to the first offspring while the second parent to the second offspring.

Figure 4 demonstrates how the crossover operator works under the circumstance that $N = 200$ and $n = 10$. As demonstrated in Figure 4, parents P^a and P^b generate offsprings O^1 and O^2 via crossover.

In Step 2.2, a mutation operator is adopted on individuals O^1 and O^2 to generate offsprings O^3 and O^4. For each gene in the individuals produced by crossover, the mutation operator first generates a random real number $p_{rand} \in [0, 1]$ and compares it with mutation probability p_{mut}. If $p_{rand} \leq p_{mut}$, then this gene is changed to a random integer g_r, satisfying that $g_r \in [1, N]$ and that g_r differs from other genes in the individual.

5. Experiments and Result Analysis

We first introduce the parameters adopted in our experiments and then investigate the performance of our proposed evolutionary algorithm on a set of test instances.

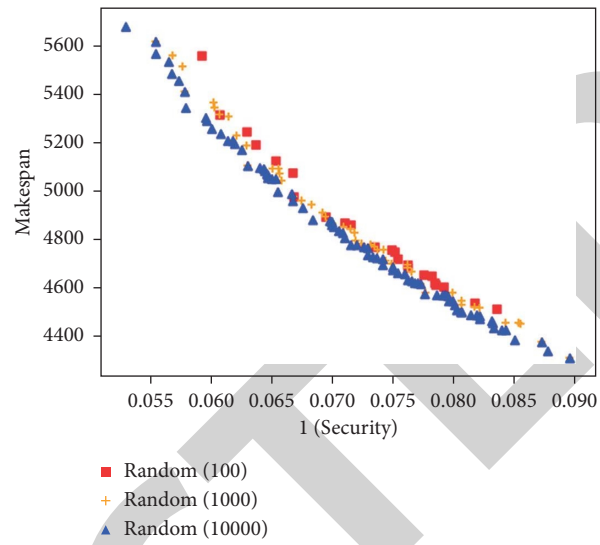

Figure 5: PFs obtained by random solutions.

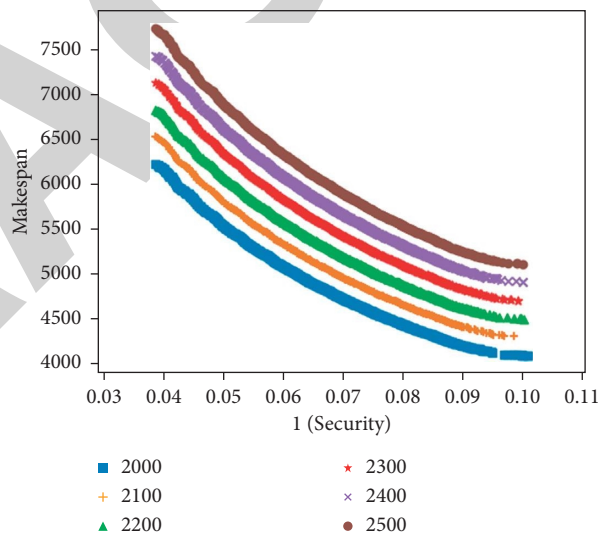

Figure 6: PFs obtained by a proposed algorithm.

The complexity of the scheduling problem in this study is $A_N^n = n!/(n-m)!$, referring to a selection of n fog nodes from a set of N nodes in which order matters. Without a doubt, we could not explore all possible solutions in the variable space when n and N are large. Hence, we randomly enumerate 100, 1000, and 10000 solutions respectively under the circumstance of $\mathscr{H} = 2000$ and $n = 10$ and calculate out their PFs, which are demonstrated in Figure 5. As can be seen, nondominated solutions grow in number and stretch out more widely in the objective space as the number of random solutions increases. Due to the time limit, we adopt 10000 random solutions as a benchmark for comparison.

Without loss of generality, we set $n = 10$ and vary workload size $\mathscr{H}$ from 2000 to 2500. Figure 6 shows the PFs obtained by our proposed algorithm. It can be observed in Figure 6 that the two objectives (maximizing security and minimizing makespan) conflict with each other. When the makespan reaches its minimum (points in the bottom right),

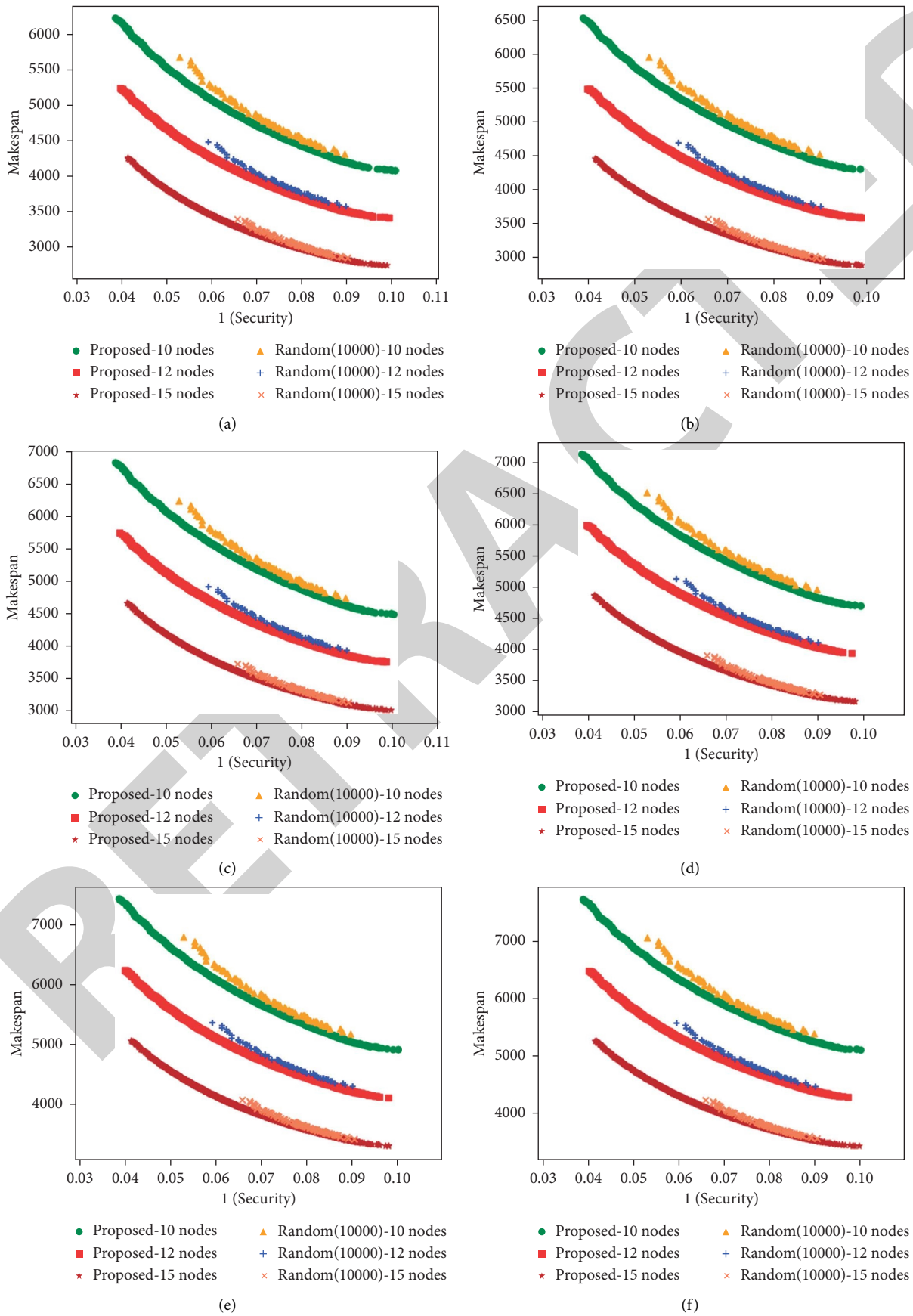

Figure 7: Comparison between the proposed algorithm and the random one. (a) $\mathscr{H} = 2000$. (b) $\mathscr{H} = 2100$. (c) $\mathscr{H} = 2200$. $\mathscr{H} = 2300$. (d) $\mathscr{H} = 2400$. (e) $\mathscr{H} = 2500$.

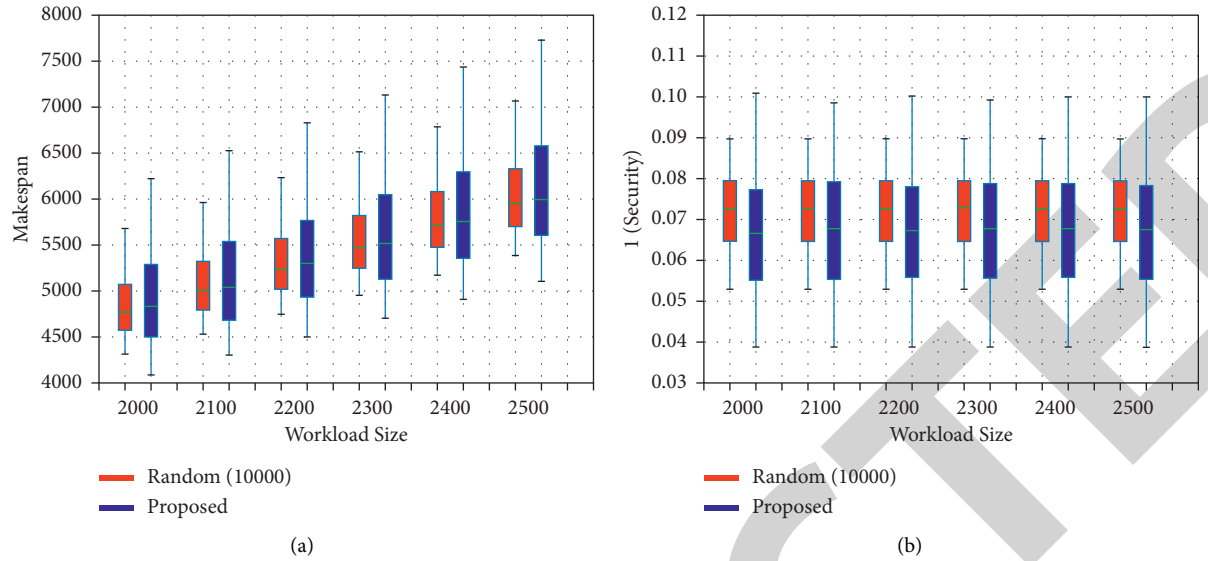

Figure 8: Boxplots showing the variation range of solutions. (a) Boxplot for makespan. (b) Boxplot for security.

security becomes the worst (the reciprocal of security is the highest). On the contrary, those solutions with high-security values (points in the top left) perform badly in terms of makespan. Moreover, as the workload size increases, the makespan gets larger and larger, while the general trend and variation range of security stay unchanged. This is because the number of fog nodes involved in workload computation is the same in this experiment.

We compare the solutions obtained by the proposed algorithm and the random one under different cases when $n \in \{10, 12, 15\}$ and $\mathscr{H} \in [2000, 2500]$. Figure 7 shows the experimental results. We can observe from this figure that for a certain workload, as more fog nodes get involved in workload computation, the makespan decreases while the variation range of security turns out to be stable for the proposed algorithm but narrows sharply for the random algorithm. This proves the powerful global searching ability of our proposed algorithm. Furthermore, for any workload cases with any number of computational fog nodes, the solutions obtained by our proposed algorithm can dominate those by the random algorithm. This means the proposed algorithm can obtain better solutions with a shorter makespan and higher security. Most important of all, Figure 7 shows that the solutions obtained by the random algorithm aggregate to a relatively small area, while those obtained by the proposed algorithm spread widely and distribute uniformly in the objective space. With these representative solutions, decision-makers own more choices in variety and can choose whichever they need according to diverse IoT application demands.

Figure 8 shows the variation range of solutions in terms of each objective function when $n = 10$ and $\mathscr{H} \in [2000, 2500]$. One can observe from Figure 8(a) that as the workload enlarges, the makespan becomes longer for both algorithms. Meanwhile, a nearly stable behavior can be observed from Figure 8(b) for both algorithms with regard to security. This is because security values are partially determined by the number of fog nodes getting involved in workload computation and that we set $n = 10$ as a fixed value in this experiment. Furthermore, it can be observed from Figure 8 that for both objectives, the proposed algorithm yields a much wider variation range of solutions than the random one does. More diverse candidate solutions imply that decision-makers can have richer and better choices to cope with various IoT application demands.

6. Conclusions

In this study, we explored the trade-offs between security and makespan when offloading workloads on fog nodes from IoT. Since these two objectives contradict each other, we proposed a multiobjective optimization model in this study, where Eigenvector centrality is adopted to measure node security and a conflict-free multi-installment scheduling (ConfMIS) strategy is designed to minimize the makespan. To solve the proposed model, we put forward an evolutionary algorithm based on decomposition and a tailor-made crossover operator. Experimental results show that the proposed algorithm achieves our goal and presents salient features. First, as more fog nodes get involved in workload computation, a nearly stable behavior can be observed for our proposed algorithm with regard to solutions' variation range of security, which proves its powerful global searching ability. Second, the proposed algorithm has superior performance in obtaining better solutions with a shorter makespan and higher security, compared to the benchmark. Third, the proposed algorithm can yield representative solutions that are widely spread and uniformly distributed in the objective space. With more diverse candidate solutions, system designers or decision-makers can have richer and better choices when facing various IoT application demands.

Data Availability

No particular data were used to support this study.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (nos. 62172457 and 61872281) and the Fundamental Research Funds for the Central Universities.

References

[1] U. Javaid and B. Sikdar, "A checkpoint enabled scalable blockchain architecture for industrial Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 11, pp. 7679–7687, 2021.

[2] Y. Gu, Z. Chang, M. Pan, L. Song, and Z. Han, "Joint radio and computational resource allocation in IoT fog computing," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 8, pp. 7475–7484, Aug. 2018.

[3] G. Li, J. Wu, J. Li, K. Wang, and T. Ye, "Service popularity-based smart resources partitioning for fog computing-enabled industrial Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4702–4711, 2018.

[4] F. Forooghifar, A. Aminifar, and D. Atienza, "Resource-aware distributed epilepsy monitoring using self-awareness from edge to Cloud," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 13, no. 6, pp. 1338–1350, 2019.

[5] J. Yao and N. Ansari, "QoS-aware fog resource provisioning and mobile device power control in IoT networks," *IEEE Transactions on Network and Service Management*, vol. 16, no. 1, pp. 167–175, 2019.

[6] Q. Fan and N. Ansari, "Towards workload balancing in fog computing empowered IoT," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 1, pp. 253–262, 2020.

[7] S. Z. Tajalli and M. E. A. A. M. T. Mardaneh, "DoS-resilient distributed optimal scheduling in a fog supporting IIoT-based smart microgrid," *IEEE Transactions on Industry Applications*, vol. 56, no. 3, pp. 2968–2977, 2020.

[8] O.-K. Shahryari, H. Pedram, V. Khajehvand, and M. D. Takhtfooladi, "Energy and task completion time trade-off for task offloading in fog-enabled Iot networks," *Pervasive and Mobile Computing*, vol. 74, Article ID 101395, 2021.

[9] A. Mebrek and A. Yassine, "Intelligent resource allocation and task offloading model for IoT applications in fog networks: a game-theoretic approach," *IEEE Transactions on Emerging Topics in Computational Intelligence*, pp. 1–15, 2021.

[10] H. Tran-Dang and D.-S. Kim, "FRATO: fog resource based adaptive task offloading for delay-minimizing IoT service provisioning," *IEEE Transactions on Parallel and Distributed Systems*, vol. 32, no. 10, pp. 2491–2508, 2021.

[11] Z. Wu, B. Li, Z. Fei, Z. Zheng, B. Li, and Z. Han, "Energy-efficient robust computation offloading for fog-IoT systems," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4417–4425, 2020.

[12] B. Donassolo, A. Legrand, P. Mertikopoulos, and I. Fajjari, "Online reconfiguration of IoT applications in the fog: the information-coordination trade-off," *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 5, pp. 1156–1172, 2022.

[13] J. Yao and N. Ansari, "Task allocation in fog-aided mobile IoT by lyapunov online reinforcement learning," *IEEE Transactions on Green Communications and Networking*, vol. 4, no. 2, pp. 556–565, 2020.

[14] M. Goudarzi, H. Wu, M. Palaniswami, and R. Buyya, "An application placement technique for concurrent IoT applications in edge and fog computing environments," *IEEE Transactions on Mobile Computing*, vol. 20, no. 4, pp. 1298–1311, 2021.

[15] Z. Zhou, H. Liao, B. Gu, S. Mumtaz, and J. Rodriguez, "Resource sharing and task offloading in IoT fog computing: a contract-learning approach," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 4, no. 3, pp. 227–240, 2020.

[16] S. Chen, Y. Zheng, W. Lu, V. Varadarajan, and K. Wang, "Energy-optimal dynamic computation offloading for industrial IoT in fog computing," *IEEE Transactions on Green Communications and Networking*, vol. 4, no. 2, pp. 566–576, 2020.

[17] A. Shokripour, M. Othman, H. Ibrahim, and S. Subramaniam, "New method for scheduling heterogeneous multi-installment systems," *Future Generation Computer Systems*, vol. 28, no. 8, pp. 1205–1216, 2012.

[18] X. Wang and B. Veeravalli, "Performance characterization on handling large-scale partitionable workloads on heterogeneous networked compute platforms," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 10, pp. 2925–2938, 2017.

[19] A. Zhou, Q. Zhang, and G. Zhang, "A multiobjective evolutionary algorithm based on decomposition and probability model," in *Proceedings of the 2012 IEEE Congress on Evolutionary Computation*, pp. 1–8, 2012.