Energy and Load Aware Fog Node Placement for Smart Farming

Jagruti Sahoo(✉)

Department of Computer Science and Mathematics, South Carolina State University,
Orangeburg, USA
jagrutisahoo@ieee.org

Abstract. Smart farming has enabled farmers to reduce cost, improve agricultural yield, and make better decisions using Internet of Things (IoT) technology. IoT nodes such as soil sensors and pH probes provide farmers with a real-time update on the farm. Traditionally, the farm data sensed by IoT nodes are processed by a cloud data center. However, it results in a higher delay in sending results to the farmer. Fog computing is a recent paradigm that reduces the delay by deploying fog nodes on the farm to process the farm data. However, the fog nodes need to be placed in proper locations as it will impact the energy consumption of IoT nodes in transmitting data to the fog node. Moreover, the placement must ensure a fair distribution of load among the fog nodes to ensure effective resource utilization. Therefore, it is critical to determine the optimal location of fog nodes to minimize the energy consumption of IoT nodes and balance load among the fog nodes. We ensure load balancing by minimizing the maximum load. In this paper, we model the fog node placement as an optimization problem and present an Integer Programming Formulation (ILP) formulation of the same. We also propose a placement algorithm designed based on *k*-means clustering. Our simulation results show that the proposed algorithm performs close to the optimal placement in terms of energy consumption and load distribution.

Keywords: Cloud computing · Fog computing · Internet of Things

1 Introduction

Agricultural production will need to increase by 60% to meet the food demand of the growing population in 2050. However, with diminishing natural resources, unpredictable weather conditions, and shrinking arable land, it is a significant challenge to increase the yield to meet future food demand [1]. Smart farming is an emerging technique that can improve agricultural yield by using IoT nodes (e.g., sensors, robots, and GPS) and advanced information and communication technologies [2]. With smart farming, the farm processes can be made smarter, automated, and data-driven, enabling the farmers to make informed decisions, reduce agricultural efforts, and minimize costs.

Typically, a smart farming system uses a remote cloud server for storage and computation. While a cloud server is needed to support compute-intensive applications, it

S. Paiva et al. (Eds.): SmartCity360° 2021, LNICST 442, pp. 77–91, 2022.
https://doi.org/10.1007/978-3-031-06371-8_6

does not meet the real-time requirement of latency-sensitive agricultural applications such as soil and water quality monitoring [3], intelligent greenhouses [3], disease and pest monitoring [3]. Fog computing is an effective paradigm that has emerged recently to allow the execution of latency-sensitive applications at the network edge, thereby reducing the latency [4, 5]. A fog infrastructure consists of edge devices such as gateways and mobile devices that can provide computational and storage resources for IoT applications. Those edge devices, also known as fog nodes, receive farm variables from the IoT nodes and execute tasks with varying resource requirements. Figure 1 shows the high-level view of a smart farming system.

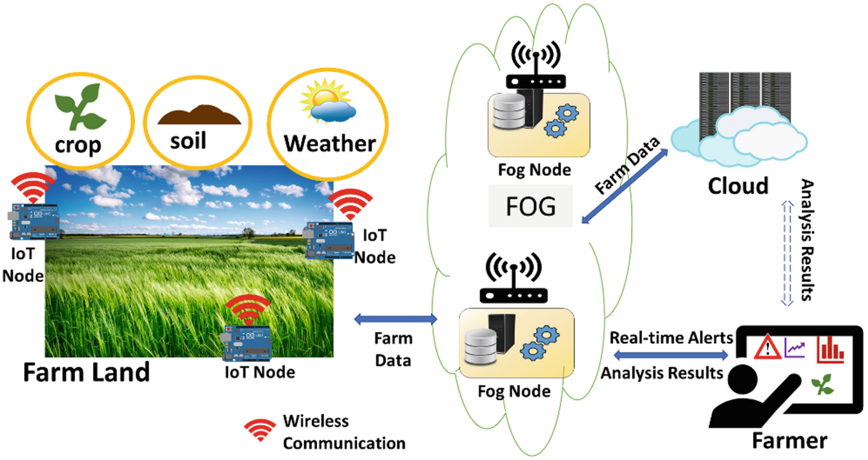

Fig. 1. Smart farming system

The IoT nodes used in smart farming are mostly battery-operated devices. As the nodes are deployed in remote locations, it is a tedious task to replace their batteries. IoT nodes need to conserve their energy to continue sensing the farm and sending the sensed measurement to the fog. However, the locations of fog nodes affect the energy consumption. Therefore, the fog nodes must be deployed in proper locations that minimize the energy consumption of the IoT nodes. Since fog nodes have limited processing capacity and IoT nodes generate workload with varying resource requirements, it is imperative that IoT nodes' workload need to be distributed as evenly as possible among the deployed fog nodes to ensure effective utilization of fog resources. The load balancing can be achieved by minimizing the maximum load on fog nodes.

In this paper, we address the problem of finding the optimal location of fog nodes, referred to as fog node placement. It entails two objectives: minimizing the energy consumption and minimizing the maximum load. We consider that the placement must meet a given budget, and we represent it in terms of the number of fog nodes that need to be deployed. Clearly, the fog node placement problem is a variation of the k-median problem that finds k centers for a given number of facilities to minimize the communication cost between the centers and the facilities [6].

The optimal placement of fog nodes for various IoT domains has been addressed in the literature [7–15]. However, there has been a handful of papers that solve the fog node placement for smart farming. Moreover, there are limited efforts on addressing the placement that ensures the energy-efficiency of IoT nodes.

In this paper, we present an ILP model of the fog node placement problem. We propose a placement algorithm, referred to as the k-fog node placement (*k*-FNP), to find the appropriate location of fog nodes. The *k*-FNP algorithm is designed based on *k*-means clustering [16]. In *k*-FNP, the IoT nodes are grouped into *k* clusters, and cluster centers represent the fog node location. In *k-means* clustering, IoT nodes are assigned to their nearest cluster, i.e., cluster that offers the smallest distance between the IoT nodes and the cluster centers. However, *k*-FNP uses a composite metric consisting of energy consumption and load within a cluster for the cluster selection process, where an IoT node selects the cluster with the smallest value of the composite metric.

The remainder of the paper is organized as follows. Section 2 discusses the related works. The fog node placement problem and the proposed algorithm are presented in Sect. 3. Section 4 presents the simulation results. Section 5 provides the conclusion.

2 Related Works

Yuan et al. [7] proposed a search and clustering algorithm to decide the fog node position based on sensor node peak density. Their algorithm is a modification of the k-means algorithm. In addition, an integrated optimization model is proposed to deal with all the resource constraints and delay. Density-based clustering algorithms perform well when the fog node has very high resource availability. However, in energy crunch situations, some fog nodes drain out all their energies very fast.

Jiang et al. [8] proposed an architecture for smart manufacturing. Also, they proposed a *k*-means based clustering algorithm and validated it through a prototype model. Three kinds of deployment, i) random deployment, ii) *k*-means clustering, and iii) improved *k*-means clustering, are considered. The SDN-based clustering is implemented on the fly based on available resources. A flow conservation-based optimization model is proposed to balance resource constraints and deal with the network delay and computing cost. The SDN-based algorithms are very effective for real-time decision-making. However, fog node placement is an infrequent activity, and usually, it is carried out when existing hardware infrastructure fails to handle growing resource needs. For each decision, fog nodes have to depend on SDN controllers/servers, which introduces an additional resource burden. As this method is implemented in a resource-rich scenario, it may not be suitable for other areas such as smart farming, connected drone coalition, and underwater sensor networks.

Manogaran & Rawal [9] proposed an optimal node placement and resource allocation algorithm to deal with communication delays for the Internet of Everything (IoE) environment. A profit function is defined to deal with resource allocation problem. The proposed RA-Fog algorithm decides the fog node placement based on latency, cost, and resource utilization. A novel fog-cloud architecture is proposed to minimize service delays. Considering the complex IoT/IoE environment scenario, only resource allocation consideration does not always find an optimal fog topology. Though resource allocation is an essential factor, the energy constraints of IoT nodes and the cost of implanting suggested fog topology are equally vital for a robust architecture. The RA-Fog algorithm will suffer heavily in energy crunch situations.

It is evident that in large-scale IoT environments, fog nodes work as a smooth transit point for heavy delay-sensitive applications. [10] proposes an edge node (EN) deployment architecture to be implemented in an airport to deal with traffic diversity and wireless diversity. In this deployment strategy, the ENs collect data from the candidate IoT nodes and sends maintenance instructions to designated IoT nodes. The throughput is the prime factor in this heavy data traffic scenario. Though node and network diversity are considered here, minimizing the number of edge nodes may lead to infrastructure fatality.

Lin et al. [11] addressed the optimal deployment of smart gateways for smart home environments. The deployment is modeled as a binary integer programming. The goal is to minimize the deployment cost while ensuring that the gateways cover all service areas in a smart home. To find the candidate gateway locations, the house floor plan is transformed into a new plan consisting of multiple nodes based on the coverage required for the service areas in the house. Although the deployment provides the required communication infrastructure for smart home IoT services, the model does not consider either energy consumption or IoT workload in choosing the gateway location.

Zhang et al. [12] addressed the edge server placement problem in smart farming that involves minimizing the data transmission delay between the sensors and edge servers and the energy cost of edge servers. Moreover, the load is balanced among the edge servers. However, the energy consumption of sensors is not considered in finding the edge server locations.

3 Optimal Fog Node Placement

3.1 Problem Description

The fog node placement problem involves finding the optimal location of fog nodes and assigning IoT nodes to appropriate fog nodes to minimize the energy consumption of IoT nodes and ensure load balancing. Load balancing can be achieved by minimizing the maximum load.

3.2 ILP Formulation

Minimize

$$\sum\nolimits_{i\in F}\sum\nolimits_{j\in S} E_{ij}x_{ij} + \lambda \tag{1}$$

Subject to

$$\sum\nolimits_{i\in F} y_i \le K \tag{2}$$

$$\sum\nolimits_{i\in F} x_{ij} = 1 \quad \forall j \in S \tag{3}$$

$$x_{ij} \leq y_i \, \forall i \in F, j \in S \tag{4}$$

$$x_{ij} d_{ij} \leq R \tag{5}$$

$$\sum\nolimits_{j \in S} \sum\nolimits_{k \in A_j} x_{ij} L_{jk} \leq \lambda, \forall i \in F \tag{6}$$

$$0 \leq \lambda \leq P \tag{7}$$

The first term in (1) represents the energy consumption objective. We used the energy model proposed in [17] to compute the energy consumption of IoT nodes. As the receiving energy of IoT nodes is negligible, the energy consumption includes the transmission energy only. Given a fog node location *i,* the transmission energy of an IoT node *i* in transmitting b_j bits to the location *i* is computed as follows:

$$E_{ij} = E_{elec} * b_j + \varepsilon_{amp} * b_j * d_{ij}^2 \tag{8}$$

where E_{elec} is the electronic energy required for coding, modulation, filtering, etc., ε_{amp} is the amplification energy, and d_{ij} is the distance between the IoT node *j* and the fog node location *i.* The second term in (1) represents the objective for load balancing. The decision variable λ represents the maximum load on a fog node that needs to be minimized (Table 1 and 2).

Table 1. ILP notations

Notation	Meaning
F	Set of candidate fog node locations
S	Set of IoT nodes
A_j	Set of tasks generated by an IoT node *j*
C_{ij}	Energy cost between location *i* and IoT node *j*
d_{ij}	Distance between location *i* and IoT node *j*
b_j	Size of data generated by IoT node *j*
L_{jk}	Size of task *k* requested by IoT node *j*
R	Transmission Range of IoT Node
P	Processing Capacity of a Fog node

Table 2. Decision variables

Notation	Meaning
x_{ij}	1 if IoT node j is served by a fog node deployed in location i and 0 otherwise
y_i	1 if location i is chosen to deploy a fog node 0 otherwise
λ	Maximum load among the fog nodes

Constraint (2) ensures that the number of fog nodes must be limited to k. It represents the maximum budget for deploying fog nodes. Constraint (3) ensures that an IoT node j is assigned to exactly one of the fog nodes. Constraint (4) indicates that an IoT node j is assigned to location i only if a location has been selected to deploy a fog node. Since the IoT nodes have a limited transmission range, we introduce a distance constraint (5). It ensures that the distance between an IoT node and the location to which it is assigned must not exceed the transmission range of the IoT node. Constraint (6) ensures that the total load on any fog node is limited by the selected maximum load. As fog nodes have limited processing capacity, constraint (7) ensures that the maximum load must not exceed that fog capacity.

3.3 *K*-means Based Fog Node Placement (k-FNP) Algorithm

The fog node placement problem can be solved using the clustering concept, where the IoT nodes are grouped to form a given number of clusters and the cluster center provide the location for deploying fog nodes. We design our algorithm based on k-means clustering which is an effective way to cluster a set of given data points into k distinct clusters. In k-means clustering, the Euclidean Distance between the data point and the centroid of a cluster is used in the cluster selection procedure, where the data point selects the cluster with the smallest distance.

Algorithm 1: k-FNP

```
Input   A set of IoT nodes, $S = \{s_1, s_2, \ldots s_m\}$
        $k$: Number of clusters
Output  Cluster centers, $C = \{c_1, c_2, \ldots c_k\}$
Select $k$ random initial cluster centers
$itr$=1
while $itr$ <= $MaxItr$ // Stopping condition
  do
    for $j$=1,2,….$m$ do //Cluster Selection
      $Z_j = \emptyset$ // Set of Candidate Clusters for IoT node $s_j$
      for $i$=1,2,….$k$ do
        //Check Distance '$d$' and Capacity Constraints
        //$G_i$: Current load of cluster $c_i$, $w_j$: load of IoT Node $s_j$, $P$: fog capacity
        $if\ d(s_j, c_i) <= R$ and $G_i + w_j <= P$
          $Z_j = Z_j \cup \{c_i\}$
      //$h_j$ Index of cluster selected for IoT node $s_j$
      $h_j = \underset{c_i \epsilon Z_j}{argmin} M(s_j, c_i)$ // Cluster Selection Metric
      $G_{h_j} = G_{h_j} + w_j$ //Update Cluster Load
    for $i$=1,2,….$k$ do //Update Centroid
      //$Loc(s_j)$: Location of IoT node $s_j$
      $c_i = mean\ (\{Loc(s_j), s_j \epsilon S \mid h_j = i\}$
    $itr$= $itr$+1
```

However, we design a composite metric consisting of two parameters: 1) the energy consumption of IoT node incurred by transmitting data to the centroid, 2) the current load of the cluster. The current load of a cluster is considered so that IoT nodes will be assigned to a cluster with a minimum load, minimizing the maximum load. Since our goal is to minimize the energy consumption and balance the load among the fog nodes, we sort the clusters using the composite metric and assign an IoT node to the cluster with the smallest value of the metric.

Since the composite metric components are in different units, they are normalized by converting each parameter into a range (0,1). The cluster selection metric for an IoT node s_j and cluster c_i is computed as follows:

$$M(s_j, c_i) = \alpha E_{ij} + \beta L_{ij} \tag{9}$$

where E denotes the transmission energy consumed by IoT node s_j if it is assigned to cluster c_i. L denotes the total load (including the workload of IoT node s_j) of cluster c_i. α and β denote the parameters that control the importance of energy efficiency and load balancing, respectively on the cluster selection metric. These parameters can be selected by the network designer during the deployment phase depending on whether the parameters are given equal or difference preference. For example, if more preference is given for energy efficiency, then α should be set to a value higher than β.

A pseudo-code of k-FNP is shown in Algorithm 1. The algorithm starts by selecting *k* random centroids from IoT node locations which are considered as candidate locations for fog nodes. As shown in the ILP, the distance between an IoT node and the centroid of its cluster must be limited to one hop. Moreover, the total load in a cluster must not exceed the processing capacity of a fog node. Thus, in our algorithm, for each IoT node, clusters that meet the distance and capacity requirements are selected. The clusters are then sorted using the composite metric given by (9). The IoT node is assigned to the cluster with the smallest value of the composite metric. After all the IoT nodes are assigned to one of the *k* clusters, the center of each cluster is updated as the mean of locations of IoT nodes that belong to the cluster. The algorithm repeats until a certain number of iterations, *MaxItr* is reached.

4 Performance Evaluation

4.1 Simulation Setup

We used IBM Cplex optimization studio [18] to implement the proposed ILP model. We compared the optimal solution with the proposed k-FNP algorithm under a small-scale scenario that consists of 4 to 10 IoT nodes randomly deployed in an area of 100 m × 100 m. The placement algorithm is executed offline prior to deploying the fog infrastructure. We considered two configurations of the small-scale scenario: 1) small-scale scenario-I that has variable number of IoT nodes, and 2) small-scale scenario-II that has fixed number (i.e., 10) of IoT nodes. We consider fog nodes with a capacity of 15000 MIPS. We consider a transmission range of 60 m for the IoT nodes. We consider that each IoT node generates a number of tasks with varying resource requirements. Each task will execute on a fog node and process the data received from the IoT nodes.

We also consider a large-scale scenario consisting of 50 to 200 nodes deployed in an area of 300 m × 300 m. The large-scale scenario is used to compare the k-FNP algorithm with a random placement algorithm. The random placement algorithm randomly selects k fog nodes that satisfy the distance and capacity constraints. Both k-FNP and the random placement algorithms are implemented using Python. In our large-scale scenario, the capacity of fog nodes is set to 40000 MIPS. The transmission range of IoT nodes is set to 125 m. We set the parameter k to 50% and 20% of the number of IoT nodes in small-scale and large-scale scenarios, respectively. The simulation parameters are listed in Table 3.

4.2 Performance Metrics

We consider the following metrics to evaluate the effectiveness of the proposed algorithm.

1) *Energy Consumption*: It is given by the sum of energy consumed by IoT nodes in transmitting data to the fog node that execute their tasks.
2) *Standard Deviation of load*: We use standard deviation to measure the distribution of load among the deployed fog nodes.

3) *Maximum Load*: It is the maximum load (in MIPS) that a fog node can host. This metric is also used to evaluate the load balancing performance of the proposed algorithm.

Table 3. Simulation parameters

Notation	Meaning
Number of IoT Nodes	4–10, 50–200
CPU Capacity of Fog Node	15000 MIPS, 40000 MIPS
Data Size of IoT Node	1 Mbps-4 Mbps
Number of Tasks	5–10
Task Size	500 MIPS-1000 MIPS
Transmission Range	60 m, 125 m
E_{elec} (nJ/bit)	50
ε_{amp} (pj/bit/m^2)	100

4.3 Results and Discussions

Figure 3 shows the energy consumption of IoT nodes for small-scale scenario-I. The energy consumption of *k*-FNP remains close to that of ILP. Both ILP and *k*-FNP show consistent energy performance irrespective of the number of IoT nodes. An increase in the number of IoT nodes results in more clusters as *k* increases proportionally to

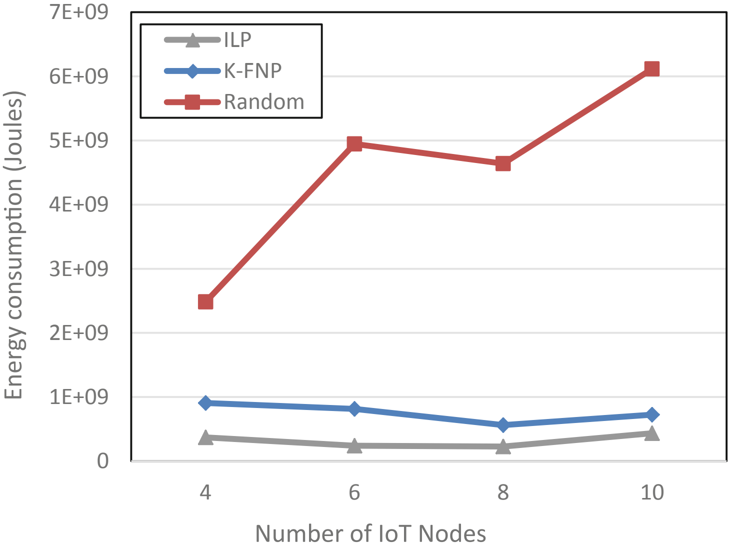

Fig. 2. Energy consumption (Small-Scale Scenario-I)

the number of nodes. With more clusters, the distance between nodes and their cluster center decreases, and the total energy remains the same compared to when there are fewer nodes. On the other hand, the energy consumption of random placement is significantly higher than that of ILP and *k*-FNP. This is because, random placement does not aim to optimize the energy efficiency of IoT nodes (Fig. 2).

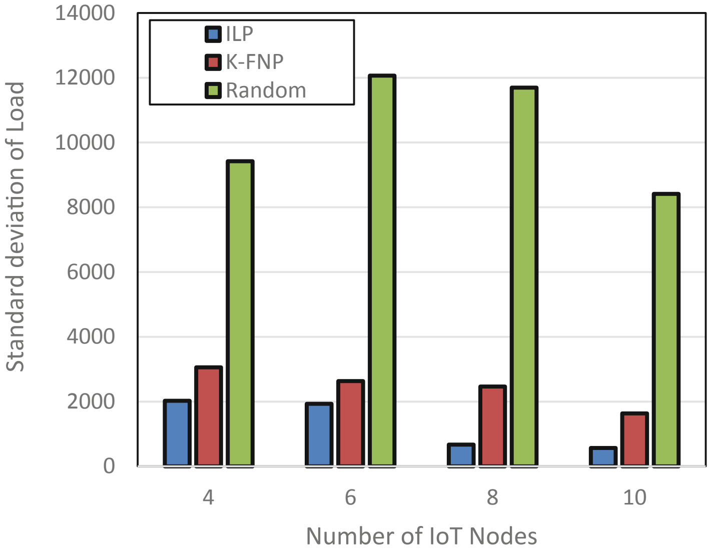

Fig. 3. Standard deviation (Small-Scale Scenario-I)

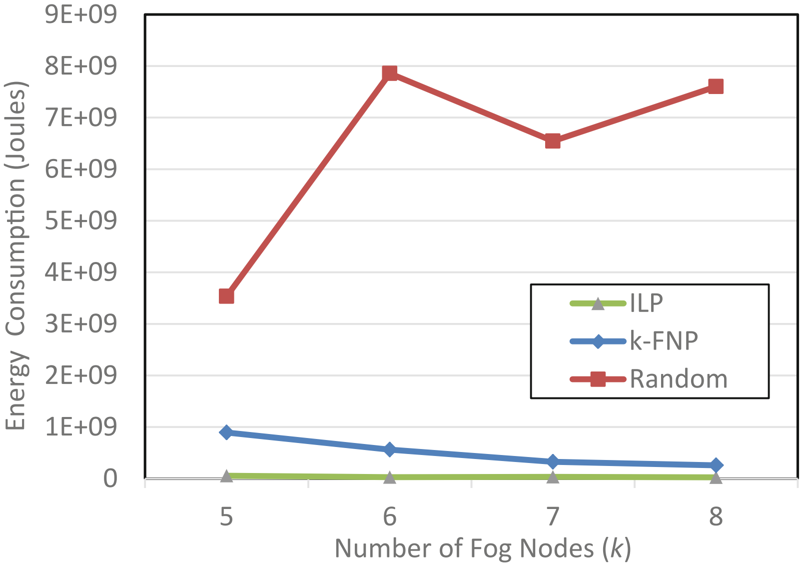

Fig. 4. Energy consumption (Small-Scale Scenario-II)

Figure 4 shows the standard deviation of load in all three schemes. We observe that k-FNP performs close to the ILP in terms of balancing the load. It outperforms random placement for any number of IoT nodes. This is because, random placement considers neither the cluster load nor the fog capacity in selecting a cluster for an IoT node. We observe that ILP and k-FNP experience a reduction in standard deviation as the number of nodes increases. Since increase in number of nodes results in more clusters and as a result, allows for efficient distribution load among the clusters.

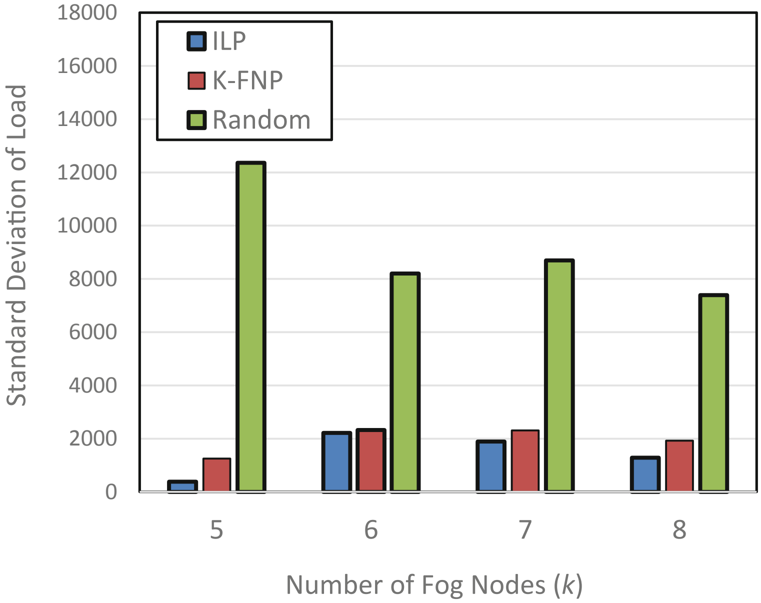

Fig. 5. Standard deviation (Small-Scale Scenario-II)

Figure 5 shows the energy efficiency performance for small-scale scenario-II with respect to the number of fog nodes (i.e., k). We observe a considerable gap between the random placement and k-FNP. Both ILP and k-FNP show a decrease in energy consumption with increase in k. This is because higher values of k lead to more clusters, decreasing the distance between IoT nodes and their selected clusters, eventually decreasing the energy requirement of IoT nodes. The performance of k-FNP is slightly lower than that of ILP, which leads to the conclusion that k-FNP can yield placements close to the optimal solutions.

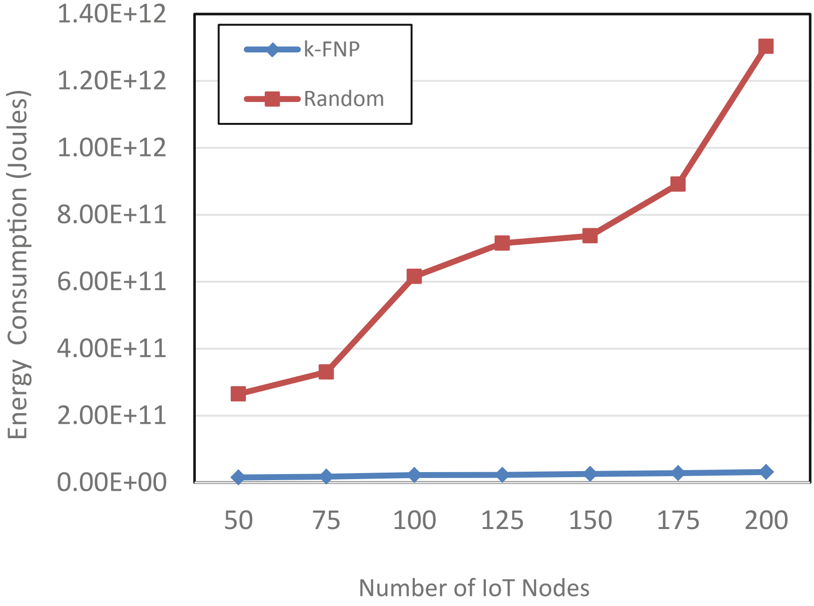

Fig. 6. Energy consumption (Large-Scale Scenario)

Figure 6 shows the standard deviation of load for small-scale scenario-II. It shows that k-FNP performs close to ILP, whereas random placement exhibits drastically poor performance across all nodes.

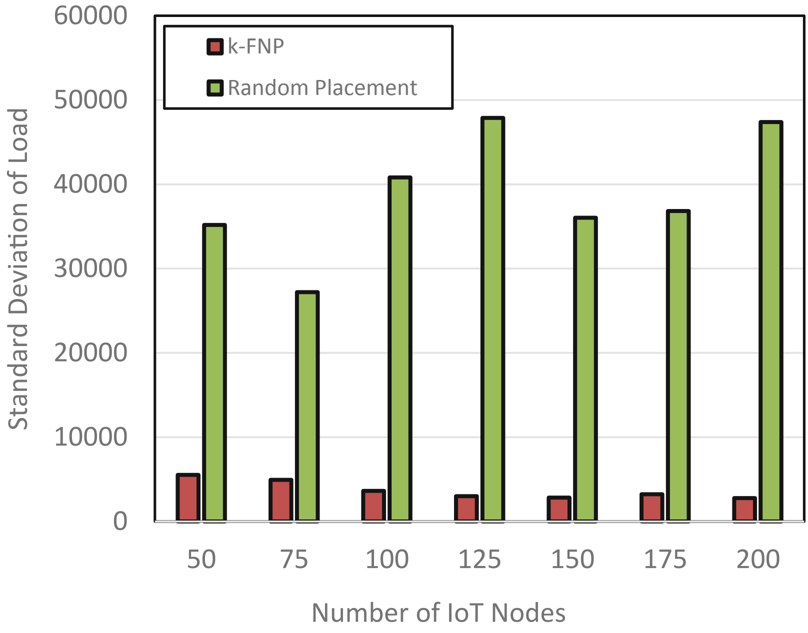

Fig. 7. Standard deviation (Large-Scale Scenario)

Figure 7 and Fig. 8 show the energy consumption and standard deviation obtained with *k*-FNP and random placement under a large-scale scenario. In Fig. 7, we observe an increase in energy consumption for random placement as the number of nodes increases. As *k* is set to only 20% of the number of IoT nodes, some fog nodes are still farther from the IoT nodes compared to when *k* is set to 50%, resulting in higher energy consumption. The growth in energy consumption of *k*-FNP is significantly low than that of the random placement, confirming the suitability of *k*-FNP in finding appropriate fog nodes.

In Fig. 8, we observe a considerable performance gap between *k*-FNP and random placement. Although the random placement considers the transmission range and fog capacity constraint in selecting clusters, it does not consider the load of IoT nodes, thereby resulting in a non-uniform distribution of load among the fog nodes.

Table 4. Maximum load (MIPS) (Small Scale Scenario-I)

Number of IoT Nodes	ILP	k-FNP	Random Placement
4	13392	13500	18000
6	13498.5	13800	24700
8	11606	13400	25800
10	11262.5	12700	21700

Table 5. Maximum load (MIPS) (Small Scale Scenario-II)

Number of Fog Nodes	ILP	k-FNP	Random Placement
5	11262.5	12238.7	21566.3
6	10934.5	11636.45	21218.05
7	9954	11062.35	22306
8	8882.5	10143.45	19316.9

Table 4 shows the maximum load for small-scale scenario-I. Both ILP and *k*-FNP experience a slight variation maximum load with an increase in the number of IoT nodes. *k*-FNP shows a maximum load close to optimal placement, thereby confirming its ability to minimize the maximum load. Moreover, on average, *k*-FNP yields 69% better performance than random placement.

Table. 5 shows the maximum load on a fog node with respect to the number of fog nodes for small-scale scenario-II. As the number of fog nodes increases, the maximum load decreases slightly for both ILP and *k*-FNP. As the IoT tasks have heterogeneous computational requirements, with higher value of *k,* the fog load can be balanced more effectively than with a lower value of *k*. We observe that random placement yields significantly higher maximum load irrespective of the number of IoT nodes. On average, Random placement results in a maximum load that is twice as much as the maximum load of the optimal placement obtained by the ILP. On the other hand, *k*-FNP remains close to ILP for any number of fog nodes.

Table 6. Maximum load (MIPS) (Large Scale Scenario)

Number of IoT Nodes	k-FNP	Random Placement
50	35415.4	79717
75	34356	76616.85714
100	31414.6	101471.2
125	31044	127496
150	33034.2	83322
175	34041.4	99661.8
200	33690	148673.3

The maximum load for the large-scale scenario is shown in Table. 6. *k*-FNP shows consistent performance in balancing load even in the presence of many IoT nodes. On the contrary, random placement shows significant variation in maximum load. Moreover, the maximum load in random placement peaks at 148673 MIPS for 200 nodes, which is 4.4 times higher than that of *k*-FNP.

5 Conclusion

In this paper, we address the fog node placement problem that entails finding appropriate fog nodes to minimize the energy consumption of IoT nodes and minimize the maximum load to ensure an even distribution of IoT workload among fog nodes. First, we present an ILP formulation of the fog node placement problem. Then, we propose a placement algorithm that is based on k-means clustering. A composite metric consisting of two parameters: energy consumption and cluster load, is designed to address the two objectives: minimization of energy consumption and load balancing. Our simulation results show that the proposed algorithm performs close to the ILP in terms of both objectives and hence can produce good quality solutions. In the future, we would like to address the placement issue considering complex scenarios that involve heterogeneous fog nodes that vary in terms of resource and security requirements.

Acknowledgment. This work is supported by the National Institute of Food and Agriculture, United States Department of Agriculture, Evans-Allen project number SCX-314–02-19.

References

1. Elijah, O., Rahman, T.A., Orikumhi, I., Leow, C.Y., Hindia, M.N.: An overview of Internet of Things (IoT) and data analytics in agriculture: benefits and challenges. IEEE Internet Things J. **5**(5), 3758–3773 (2018)
2. Ivanov, S., Bhargava, K., Donnelly, W.: Precision farming: sensor analytics. IEEE Intell. Syst. **30**(4), 76–80 (2015)
3. Ahmed, N., De, D., Hussain, I.: Internet of Things (IoT) for smart precision agriculture and farming in rural areas. IEEE Internet Things J. **5**(6), 4890–4899 (2018)
4. Naha, R.K., et al.: Fog computing: survey of trends, architectures, requirements, and research directions. IEEE Access **6**, 47980–48009 (2018)
5. Mouradian, C., Naboulsi, D., Yangui, S., Glitho, R.H., Morrow, M.J., Polakos, P.A.: A comprehensive survey on fog computing: state-ofthe-art and research challenges. IEEE Commun. Surveys Tutorials **20**(1), 416–464 (2018)
6. Xia, X., et al.: Budgeted data caching based on k-median in mobile edge computing. In: IEEE International Conference on Web Services (ICWS), pp. 197–206 (2020)
7. Yuan, X., He, Y., Fang, Q., Tong, X., Du, C., Ding, Y.: An improved fast search and find of density peaks-based fog node location of fog computing system. In: IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), pp. 635–642 (2017)
8. Jiang, C., Wan, J., Abbas, H.: An edge computing node deployment method based on improved k-means clustering algorithm for smart manufacturing. IEEE Syst. J. **15**(2), 2230–2240 (2021)
9. Manogaran, G., Rawal, B.S.: An efficient resource allocation scheme with optimal node placement in iot-fog-cloud architecture. IEEE Sensors J. **21**(22), 25106–25113 (2021)
10. Zhao, Z., Min, G., Gao, W., Wu, Y., Duan, H., Ni, Q.: Deploying edge computing nodes for large-scale iot: a diversity aware approach. IEEE Internet Things J. **5**(5), 3606–3614 (2018)
11. Lin, P.: Optimal smart gateway deployment for the Internet of Things in smart home environments. In: IEEE 4th Global Conference on Consumer Electronics (GCCE), pp. 273–274 (2015)
12. Zhang, J., Li, X., Zhang, X., Xue, Y., Srivastava, G., Dou, W.: Service offloading oriented edge server placement in smart farming. Softw Pract Exper. 1– 18, (2020)
13. Gravalos, I., Makris, P., Christodoulopoulos, K., Varvarigos, E.A.: Efficient gateways placement for internet of things with QoS constraints. In: IEEE Global Communications Conference (GLOBECOM), pp. 1–6 (2016)
14. Lee, J.-H., Chung, S.-H., Kim, W.-S.: Fog server deployment technique: an approach based on computing resource usage. Int. J. Distrib. Sensor Netw. **15**(1), 1550147718823994 (2019)
15. da Silva, R.A.C., da Fonseca, N.L.S.: On the Location of Fog Nodes in Fog-Cloud Infrastructures. Sensors **19**(11), 2445 (2019)
16. Guo, X., Lin, H., Wu, Y., Peng, M.: A new data clustering strategy for enhancing mutual privacy in healthcare IoT systems. Futur. Gener. Comput. Syst. **113**, 407–417 (2020)
17. Heinzelman, W.R., Chandrakasan, A., Balakrishnan, H.: Energy-efficient communication protocol for wireless microsensor networks. In: The 33rd Annual Hawaii International Conference on System Sciences, vol. 2, p. 10 (2000)
18. IBM CPLEX Optimizer [Online] Available: https://www.ibm.com/analytics/cplex-optimizer