Received: 17 May 2021 | Revised: 7 July 2021 | Accepted: 18 August 2021

DOI: 10.1002/dac.4974

RESEARCH ARTICLE

WILEY

A distributed swarm intelligence-based energy-saving method among massive edge nodes

Jianpeng Qi[1] | **Suli Ren**[1] | **Boran Wang**[1,2] | **Rui Wang**[1,3,4]

[1]School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, China

[2]Beijing TianTan Hospital, Capital Medical University, Beijing, China

[3]Shunde Graduate School, University of Science and Technology Beijing, Foshan, China

[4]University of Science and Technology Beijing, Beijing, China

Correspondence

Rui Wang, University of Science and Technology Beijing, Beijing, China.
Email: wangrui@ustb.edu.cn

Funding information

National Natural Science Foundation of China, Grant/Award Numbers: 72004147, 62173158, 61803391; Scientific Technological Innovation Foundation of Shunde Graduate School, USTB, Grant/Award Numbers: BK19CF010, BK20BF012; National Key Research and Development Program of China, Grant/Award Number: 2016YFC0901303

Summary

In edge computing, how to save energy among sustainable edge nodes is a hot topic. ON/OFF switching of edge nodes as a key point is efficient but still suffers from the long round-trip time problem because of its centralized control manner. Especially in the wireless network, service coverage is proved to be NP-Complete. To this end, we propose a Distributed Swarm intelligence-based Energy-saving algorithm (DSE). In DSE, pheromone and residual energy are used to calculate the wake-up probability. Through the wake-up probability, the edge node can be activated periodically and efficiently. In order to balance the energy in the whole system that contains massive edge nodes, we further use a correction factor, that is, DSE+, to adjust the wake-up probability of the nodes. The proposed methods allow for distributed implementation without requiring a centralized control by the coordinator, and the pheromone accumulated temporally and spatially. In addition, they do not require node localization. Experiments show that both DSE and DSE+ can work as expected, and DSE+ with the correction factor improves the lifetime of the whole system at least 12.6% compared with the DSE without the correction factor.

KEYWORDS

ant colony optimization, edge computing, energy-balanced framework, decentralized control, renewable energy sources

1 | INTRODUCTION

Energy saving or carbon footprint is a critical issue for both cloud computing and edge computing. Some recent investigations show that the power consumption of a 5G site exceed 10 kW,[1] and the carbon emissions of training common large AI models are more than 284 tonnes,[2] which provide us many benefits in practice except for higher-energy consumption. To moderate this impact, in cloud computing, there are often two typical solutions including the workload consolidation and turning off spare servers to reduce carbon footprint.[3] Unfortunately, edge computing is not capable of the impact due to its various and volatile characteristics (e.g., geographical distribution, mobility, and heterogeneity.[4,5]). More specifically, geographical distribution makes the centralized management stuck at bandwidth and lead the distributed service to inefficiency; mobility causes the metastore or master hard to deal with the nodes moving in different directions back and forth; heterogeneity leads the runtime task scheduler failed to align the diverse distributed tasks.[6] This evidence bears out that simply integrating the centralized cloud computing methods of energy management into edge computing is really a pressure not only for the platform but also for the developer. In particular, for uncertain mobile devices, for example, the walking people or the commuters, collaborative task processing service is hardly to promise.

There are, of course, innumerable related concepts that aim to dispatch moving users' tasks to massive edge nodes (edge nodes can be smartphones, wearable devices, sensors, etc.) such as mobile edge computing, cloudlet computing, and crowdsourcing[7] All these concepts and their scenarios make decentralized edge native algorithms gained wide-spread public attention. Scaling out with a lot cheaper nodes in these realms can bring us much more benefits than just scaling up a single server. However, on the basis of above-mentioned characteristics, minimizing energy cost or energy management through a centralized way on these scenarios is a nontrivial task. As one of the most popular optimization methods, ant colony optimization[8] (ACO) can be used to accelerate the progress of edge computing. For example, find a more reliable path to provide a distributed service and solve the service placement problem.[9] In this work, take advantage of the ACO, which breaks down a top task into parallel subtasks, we propose a Distributed Swarm intelligence-based Energy-saving (DSE) method. DSE intelligently uses a probability to control the edge node ON/OFF state. Most importantly, to solve the unbalanced energy consumption problem caused by the frequent patterns or activities, which is usually happens in the ant colony optimization where many ants scanning the food in the same path, we also put forward an energy correction factor to adjust the probability. To the best of our knowledge, this is the first work that leverages the bio-inspired algorithm to avoid the edge nodes in a working or standby state all the time in edge computing. The main contributions of this work are listed in the following.

1. We apply the bio-inspired algorithm into the edge node ON/OFF state control and further propose a novel DSE method to save the energy consumption of edge nodes;
2. In the face of an unbalanced energy consumption problem caused by the frequent moving patterns, we carry out a correction factor tunning process plus DSE (DSE+) to prolong the life of the service; and
3. We evaluate the performance of DSE and DSE+ by extensive experiments and give a full analysis of these two algorithms.

The remainder of this paper is organized as follows. Section 2 introduces the related work for energy saving in edge computing. Section 3 presents the DSE and DSE+ algorithms. Section 4 carries out the simulation experiments and analyzes the experimental results. Section 5 concludes the paper.

2 | RELATED WORK

In this section, we give a brief overview in regard to energy-saving in edge computing. Generally speaking, workload consolidation and turning off (ON/OFF switching) idle nodes are two important approaches to reduce the carbon footprint.

Workload consolidation belongs to the task offloading (or computation offloading) problem which reshuffles the distributed subtasks taking the node resource utilization into account and aims to reach a minimal energy consumption goal. Weiwei et al. propose a multi-user multitask computation offloading framework in a renewable mobile edge cloud system to maximize the overall system utility.[10] This framework can map the workload of MD (mobile device) onto multiple WDs (wireless devices) which has an energy harvesting equipment to collect renewable energy from the environment. Pu et al. assume a set of MDs that form a mobile edge cloud and help to execute tasks from each other to minimize the average energy consumption of all MDs.[11] To encourage the cooperation among different MDs, incentive schemes are introduced to prevent one MD from overriding the energy from other MDs. And then, they propose a D2D Crowd framework for 5G mobile edge computing.[12] Based on control assistance from the network operator, a massive crowd of devices at the network edge can beneficially share the resources. Experiments show that more than 50% carbon footprint is reduced. Xiang et al. design efficient Avatars (virtual machine) placement strategy that migrates the Avatars to the proper cloudlets according to their energy demand.[13] Yunbo et al. propose a framework that balances performance and energy cost trade-offs for real-time data analysis.[14] They leverage the green energy to analyze the users' tasks and use the brown energy from the grid as a backup.

However, it is worthwhile to point that all the nodes, in workload consolidation, are still in a working state. In other words, the energy of the node is still dwindling, especially the idle nodes in a massive edge nodes' scenario. Thus, it is essential to turn off the idle nodes.[15]

Allai et al. utilize the Wake-On-LAN technology to design a centralized energy-saving framework.[16] They use a Nd-LoRa(LoRa Endpoint) device to control the edge node state. This device can receive the signal from the core network and make the inactive edge node into a working state. However, the latency is intolerable in edge cloud collaboration,[17] especially when the ISP barrier exists.[18] Meanwhile, the periodic listen-sleep mechanism is another way that can be utilized in ON/OFF control. Based on flexible duty cycling, Parzysz et al. propose a 5G Discontinuous Transmission mode (gDTX).[19] At the start of the gDTX cycle, the base stations are switched on and users can transmit data to each other. At the end of the gDTX cycle, base stations are supposed to sleep. Ning et al. put forward a green and sustainable virtual network embedding framework for cooperative edge computing in wireless-optical broadband access networks.[20] They first divide the nodes into working and standby subparts and then reduce the energy consumption of backup service by just switching k standby nodes on. Zheng et al. integrate with ZigBee to implement an energy-saving system for office building lighting.[21] They implement a video monitoring system that can detect people activities with edge computing to control the switch of the luminaries or switch the preset illumination mode to save energy. There, however, are still shortcomings; these methods do not generalize to the edge computing with massive edge nodes. Instead, we propose a more general decentralized method to solve the energy-saving problem, which can be used in many scenarios only if the edge node can communicate with each other.

3 | ALGORITHM

In this section, we give the basic definitions and the implementation associated with DSE and DSE+.

To be more clear in practice, we formalize the scene to be a square area of $L * L$ where N wireless edge nodes are distributed randomly. To minimize the cost from a system perspective, instead of keeping all the nodes in an active state, our aim is to only invoke the edge nodes in the vicinity of the mobile user who has the demand of computation offloading. This can be used in many scenarios, such as avoiding cold starting in a real-time system and preparing resources before the users offload their computation. We mainly adopt the S-MAC synchronization ideas[22] in the realm of wireless sensor networks (WSNs) to periodically synchronize the pheromone and leave the mechanism explanation of communication out during the pheromone control step. Once the energy of a node is exhausted, it turns into invalid state permanently. It is worth noting; we focus on the distributed edge node state control; the cost of executing an offloaded task is omitted.

Figure 1 depicts the core steps of DSE and DSE+, that is, “node activation,” “warming up,” and “pheromone control.” In the phase of “node activation,” an edge node randomly chooses the state of “wake” or “sleep” according to its “wake probability.” Here, “wake” refers to “wake-up sensing module” because the communication module only works in the “pheromone control” phase. The edge node in the “wake” state turns on its wireless (or network scanning) module and initializes the predefined services (“warming up”) if it detects users in its sensing range (with the detection probability P_d), while the node in the “sleep” state turns off its sensing module

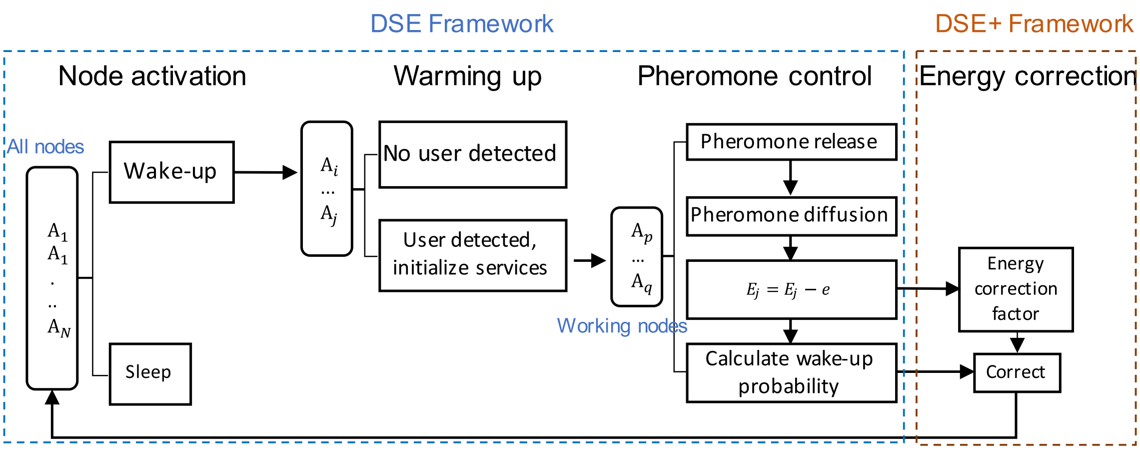


FIGURE 1 Framework of distributed swarm intelligence-based energy-saving, DSE and DSE+

for energy saving. For an edge node that does not sense the nearby users, the "node activation" phase can be periodically scheduled for every T's duration.[1] In the phase of "pheromone control," the nodes receive pheromone for collaboration and only the nodes that sense the user in this round will send pheromone of target information to their neighbors. The nuance of DSE and DSE+ is DSE+ arms with "energy correction" that aims to alleviate the energy skewed distribution problem.

Some basic definitions are listed as follows.

1. R_c is the max communication range of the node, R_s is the sensing radius to sense users, and P_d is the probability of a user is detected by an activated node within R_s.
2. The energy cost of waking up a node, stay in active, and releasing pheromone are the same, which is e.

3.1 | DSE algorithm

In this section, we present the process of DSE framework illustrated in Figure 1.

Node activation. This phase intends to turn the state of a sensing module from off (or sleep) into active. We define each edge node as an ant A_j and write $A=\{A_j|j=1,...,N\}$ for all the nodes in the square area of $L*L$. As with the ant algorithm, I_j represents for pheromone of A_j, initializing from its minimum I_j^{min}. Then, we define the state of a sensing module during the kth ($k \geq 1$) period as follows:

$$Stat(j,k)=\begin{cases}1 & A_j \text{ is on sensing (or searching) state}\\ 0 & otherwise\end{cases} \tag{1}$$

Let $E(j, k)$ denote the total amount of energy for A_j in the kth period. Then, $E(j,k)=E(j,k-1)-e$, if $Stat(j,k)=1$.

Warming up. To alleviate the cold start problem, functions or services that user requested usually need to be preloaded. This process can also consume energy; in this work, we assume this equals to e, the same as "node activation" phase, for simplicity. Let $Appear(k)$ denote whether the user is appeared in the sensing range of a node and $Det(j, k)$ whether the user is detected by node A_j, in kth period.

$$Appear(k)=\begin{cases}1 & \text{User exactly exists in the sensing area in the } k\text{th period}\\ 0 & \text{otherwise}\end{cases} \tag{2}$$

$$Det(j,k)=\begin{cases}1 & Stat(j,k)=1, Appear(k)=1, \text{and } A_j \text{sensed the user}\\ 0 & \text{otherwise}\end{cases} \tag{3}$$

These two indicators also imply that a false negative or a false positive result can be produced due to sensing ability of the node is limited. And this is why we define the P_d as the probability ($0 \leq P_d \leq 1$) of an activated node sensing the user in its R_s range.

Meanwhile, provisioning a reliable service in production always needs backups. And the effect can be assessed as the number of nines in the realm of system reliability. In this paper, reliability can be regarded as the detection probability by edge nodes. Let $R(t)$ denote the reliability of the user is detected in tth period. Write $\overline{B_{j,t}}=Pr\{Det(j,t)=0\}$. Then,

[1]The time synchronization has fully developed in WSNs, such as the sensor-medium access control (S-MAC) protocol. The operating period T usually contains two phases T_1 and T_2, T_1 for "wake/sleep" and T_2 for "synchronization."

$$\begin{aligned} R(t) &= 1 - Pr(\overline{B_{t,1}}, \overline{B_{t,2}}, ..., \overline{B_{t,N}}) \\ &= 1 - Pr(\overline{B_{t,1}})Pr(\overline{B_{t,2}})...Pr(\overline{B_{t,N}}) \\ &= 1 - \prod_{i=1...M} Pr(\overline{B_{t,i}}) \\ &= 1 - \prod_{i=1...M} (1 - P_d) \end{aligned} \tag{4}$$

where M is the number of activated nodes in $t$period. After this phase, $Energy(j,k) = Energy(j,k) - e$.

Pheromone control. This stage aims to calculate the wake-up probability of a node after the warming up phase. In this stage, the node that detected user releases a certain amount of pheromone to its neighbors. Based on the ant algorithm, the wake-up probability can be calculated according to the accumulated pheromone value of the last step.

The neighbors of jth node is defined as follows:

$$neighbor(j) = \{i \| \|Pos(A_j) - Pos(A_i)\| \leq Rc \, and \, i = 1, ..., N\} \tag{5}$$

where $\|Pos(A_j) - Pos(A_i)\|$ is the distance between the jth and the kth nodes. Obviously, we assume that the neighbors of a node are its directly reachable nodes to avoid routing loop.

If the jth node confirms that the user exists, then it will release a unit pheromone increment $\tau(j, k)$ to its neighbors:

$$\tau(j,k) = \begin{cases} 1 & Det(j,k) = 1 \, j = 1, ..., N \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

Calculation of wake-up probability. The pheromone concentration is uniform given a node A_j, and it comes from two sources. One is the newly released pheromone increment $\Delta I_j(k)$ of A_j and/or its neighbors

$$\Delta I_j(k) = \sum_{i \in neighbor(j)} \left(\frac{\tau(i,k)}{|neighbor(i)|} \right) \tag{7}$$

Another source is the residual of previously released pheromone. We assume that the pheromone decreases at a loss ratio $\rho(\rho \in [0, 1])$ over time, then the residual pheromone can be denoted as $(1 - \rho)I_j(k - 1)$.

Then,

$$I_j(k) = \min\{I_j^{max}, \max\{I_j^{min}, \Delta I_j(k) + (1-\rho)I_j(k-1)\}\} \tag{8}$$

where I_j^{max} and I_j^{min} are the maximum and minimum pheromone levels of A_j.

Thus, we get the wake-up probability of A_j in the $(k + 1)$th period:

$$W'(j,k+1) = \frac{I_j(k)}{I_j^{max}}, j = 1, ..., N \tag{9}$$

Algorithm 1 DSE Algorithm: a single node perspective

```
1: Initialize pheromone I_j(1) ← I_j^min, Energy ← E;
2: W'(j,1) ← I_j(1)/I_j^max;
3: for k ← 1 to periods do                                  ▷ Timeline
4:     for A_j ← 1 to N do
5:         r_j ← rand() ∈ [0,1];
6:         if W'(j,k) ≥ r_j then
7:             Stat(j,k) ← 1;                               ▷ Wake up the jth node
8:             Energy(j,k) ← Energy(j,k) − e;
9:             if Det(j,k) = 1 then
10:                Release pheromone τ(j,k) to A_j's neighbors;
11:                Energy(j,k) ← Energy(j,k) − e;
12:            end if
13:        end if
14:        W'(j,k+1) = I_j(k)/I_j^max
15:    end for
16: end for
```

Algorithm 1 depicts the pseudo-code of DSE. We first initialize the pheromone and the energy of each node A_j with I_j^{min} and E (Line 1). Line 2 shows the initial wake-up probabilities for each node A_j. Next, Lines 4–15 show the processes of updating the accumulation of pheromone and wake-up probabilities for each node. If the wake-up probability $W'(j,k)$ of A_j is grater than or equal to r_j in the kth period, A_j is awakened (Lines 6–12).

3.2 | DSE+ algorithm

For hot spots such as the central business district, roadside, and supermarket, the frequently awakened nodes in these areas may cost too much energy than cold areas, which leads to an energy imbalanced problem. How to balance the energy of these nodes within hot areas is important. Plus DSE, we propose an energy correction factor to alleviate this problem.

Energy correction factor reveals the relative relationship between energy of different nodes. Then by the adjustment of wake-up probabilities according to the energy correction factor, lifetime of an edge computing system can be prolonged. Let λ denote the energy correction factor, then

$$\lambda = \frac{Energy(j,k) - \overline{E(j,k)}}{Energy(j,k)} \tag{10}$$

where $\overline{E(j,k)}$ is the average energy of all neighbors of node A_j and we can write it in the form

$$\overline{E(j,k)} = \frac{\sum_{i=1}^{n} Energy_j(i,k)}{M} \tag{11}$$

where M is the number of alive neighbors of A_j.

Then, we get the final calculation of the wake-up probability:

$$W(j,k+1) = max\left\{\frac{{I_j^{min}}^{max}}{I_{\ j}}, min\left\{1, \frac{I_j(k)^{max}}{I_{\ j}} + \lambda\right\}\right\} \tag{12}$$

Algorithm 2 shows the version of DSE plus the energy correction factor λ, namely, DSE+.

Algorithm 2 DSE+ Algorithm: a single node perspective

Initialize pheromone $I_j(1) \leftarrow I_j^{min}, Energy \leftarrow E$;
$W'(j,1) \leftarrow I_j(1)/I_j^{max}$;
for $k \leftarrow 1$ to *periods* **do**
    for $A_j \leftarrow 1$ to N **do**
        $r_j \leftarrow rand() \in [0,1]$;
        if $W'(j,k) \geq r_j$ **then**
            $Stat(j,k) \leftarrow 1$;
            $Energy(j,k) \leftarrow Energy(j,k) - e$;
            if $Det(j,k) = 1$ **then**
                Release pheromone $\tau(j,k)$ to A_j's neighbors;
                $Energy(j,k) \leftarrow Energy(j,k) - e$;
            end if
        end if
    end for
    For each node A_j, calculate λ; ▷ Calculates the energy correction factor
    $W(j,k+1) \leftarrow I_j(k)/I_j^{max} + \lambda$; ▷ Updates the wake-up probability
end for

3.3 | Time complexity analysis

In DSE and DSE+, the event that an ant randomly walks to search the food within its search zone represents the random user detection by a node within its sensing zone (the discs area centered at the node with the diameter being the sensing range R_s). Thus, the ant in our model is not strictly equal to the ACO. Actually, providing services in the wireless network scenario can be referred to as the network coverage problem (or the dominating set problem), which is proved to be NP-complete.[23] Instead, we solve this problem by just considering the neighbors of the node, which is a local view angle. When a node senses the user, it diffuses pheromone to all its neighbors. Thus, suppose the average number of neighbors of the sensor is N, we can get the time complexity $\mathcal{O}(N)$. The reader who has an interest in number N can check[22] for further reading.

4 | SIMULATION RESULT ANALYSIS

In this section, we evaluate the performance of our proposed algorithms through simulations. We carry out Monte Carlo simulations of our algorithms in MATLAB, and the following is a detailed description of the simulation.

4.1 | Simulation environment setting

We set the area to be a square area of 200 m $\times$ 200 m, where 500 edge nodes are randomly scattered. The communication range $R_c = 30$ m and sensing radius $P_s = 0.9$. Initially, we set the energy value of every node equal to 1000, and when the energy runs out, it stops working immediately. And the whole system is out of service if the users cannot be detected for a continuous 10 steps. For simplicity, let $e = 1$. We assume a user carrying equipment moving along the circle with a radius of 80 m in a uniform circular motion and 50 steps for each round. The design environment is shown in Figure 2.

4.2 | Performance indicators explanation and setting

In order to better illustrate the performance of DSE and DSE+, we give the following indicators.

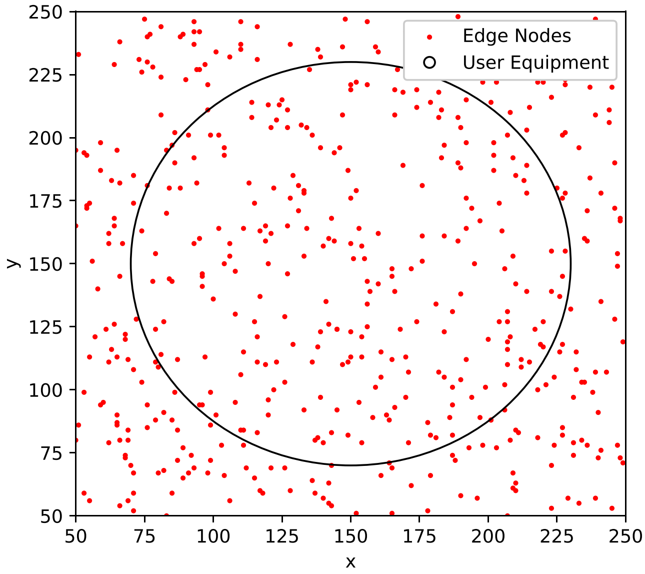


FIGURE 2 Uniform circular motion of the user

Reliability. It is obvious we can get high reliability with more activated edge nodes. In this simulation, we let P_d the probability of a user that is detected by an activated node, equal to 0.9. For example, according to Equation (4), if two nodes awaken simultaneously, the reliability is $1-(1-0.9)^2=0.99$.

Total number of node failures. This indicator refers to the number of node failures at a certain step. A slow number of failures growth means the system can provide the user a long time service, which also implies that the energy consumption is under balance between the hot and the cold areas.

The step at detection of the user. We assume that detection of the user is the step when two or more nodes have discovered the user. This index can measure the delay of detection, and fewer steps indicate short delays.

Gray scale map. The gray scale map shows the detection effect of the algorithm on the sensing area. Black area represents the blind area of detecting, and higher brightness means more awakened nodes.

Network coverage. Network coverage represents the ratio of the total of covered pixels to the total of all pixels in the gray scale map. This ratio shows the performance in service provisioning of the whole network. Specifically, it is calculated as follows:

$$Cov=\frac{K_d}{K_{num}} \tag{13}$$

where K_d is the total number of the covered pixels by awakened nodes and K_{num} is the total number of all pixels.

4.3 | Simulation analysis

We first compare DSE and DSE+ with a well-known decentralized node random activation (RA) method[24] on the node activation angle. We implement RA with three possibilities 1, 0.5, and 0.077; for the sake of the simplicity, denote these three implementations as RA1, RA0.5, and RA0.077. Be aware of that 0.077, not 0, is the smallest probability of activating a node. And RA1 means sensing modules of all the nodes are in active, which is unpleasant from the sustainable perspective.

Figure 3 depicts the total number of nodes being activated during the 50 steps of the user movement. This observation gives a picture about the *node activation phase*. A lower number of activated nodes will have an impact on the service provisioning. We can see that both the results of DSE and DSE+ are approximately equal, and the number of the activated nodes of DSE and DSE+ are higher than RA0.077. This is because all the nodes are with the same probability at the beginning before the user being detected. After the user being detected, the pheromone of the nodes near the user is permeated, which leads to a higher wake-up probability. Most importantly, DSE and DSE+ are better

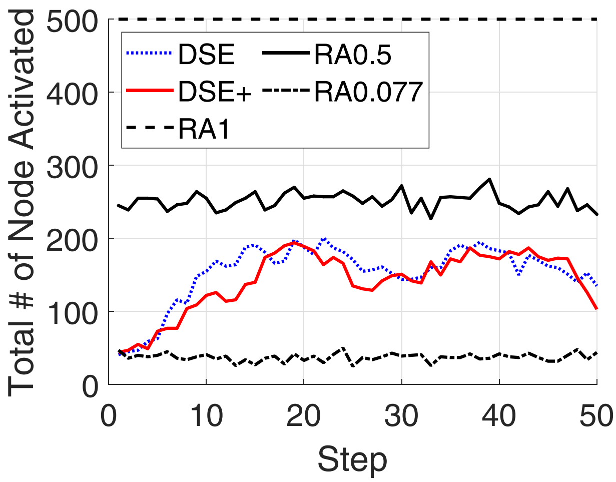


FIGURE 3 Total number of node activated w.r.t. movement of the user

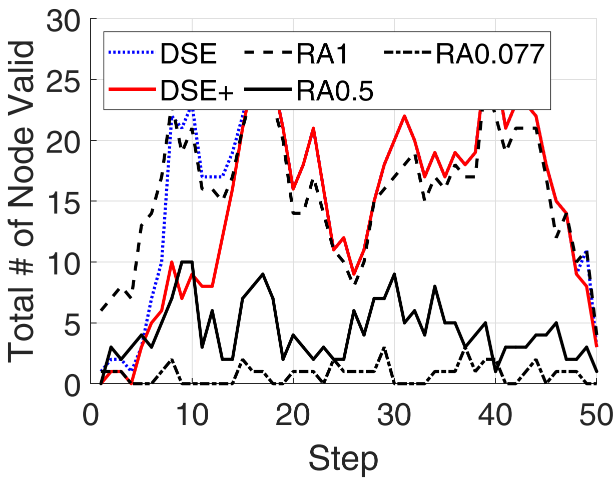


FIGURE 4 Total number of node valid w.r.t. movement of the user

than RA0.5, approximately a half of RA0.5 and better than RA1 at that. This implies that the number of the activated nodes for DSE and DSE+ is smaller than RA1 and RA0.5 especially in the areas without user.

Figure 4 depicts the total number of nodes from the activated nodes that have sensed the user. A higher number means a higher reliability can be promised. All these valid nodes can turn into the *warming up phase* to provide a more reliable service, which can be referred to the standby state in reliability theory. We can see that DSE and DSE+ are nearly equal, approximately the same with RA1, and better than RA0.5 and RA0.077. Putting together with Figure 3, this result implies that DSE and DSE+ can get a result like RA1 and also keeps the number of nodes being activated at a lower level.

Then, to evaluate the accuracy of DSE and DSE+, we record the average reliability of the nodes with the varying number of rounds; Figure 7 illustrates the results. We can see that the reliability of DSE+ is almost equal to the DSE from the first to the fifth round, which is close to 1. In the earlier rounds, there is almost no difference between two algorithms because the energy for each node is affordable, and the number of working nodes are sufficient to detect the user. But after a few rounds, reliability in DSE is significantly higher compared with DSE+. The reason is that DSE+ leverages the energy correction to make the whole system working well even after several rounds. In statistics, the user moves more 1.39 rounds (about 69 steps) than that in the DSE algorithm on average over 100 times of experiments. And the lifetime of the whole system backed up with DSE improved about 12.6% compared with the DSE+ algorithm.

Figure 5 depicts the total number of node failures with varying the user step. Before step 200 ($200/50 = Round4$), there are almost no node failures since their energy is still sufficient. And 200 steps later, an increase can be seen in the

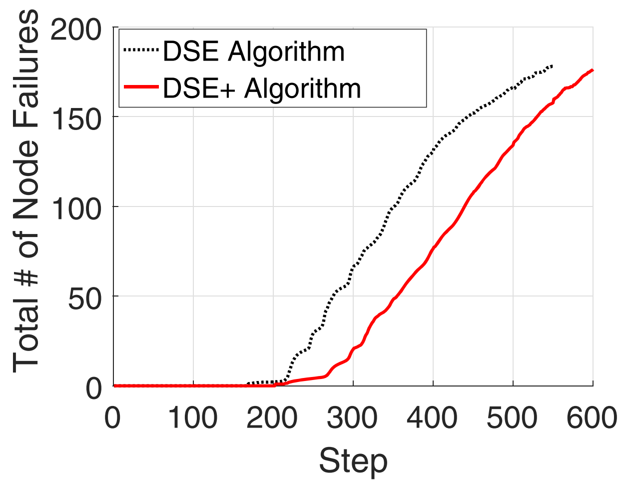


FIGURE 5 Total number of node failures w.r.t. movement of the user

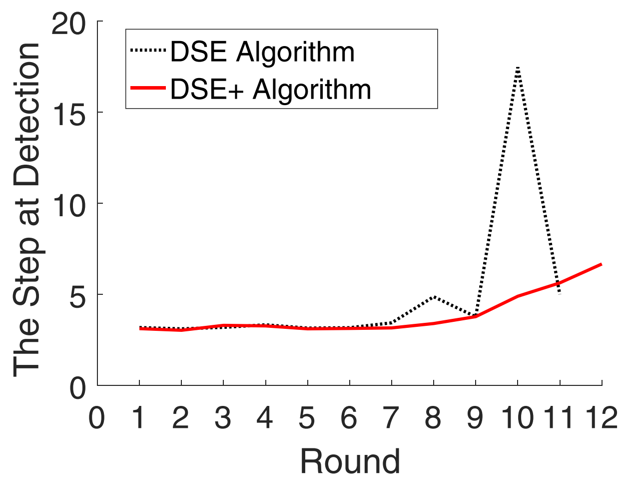


FIGURE 6 Final step at detection of the user w.r.t. number of round

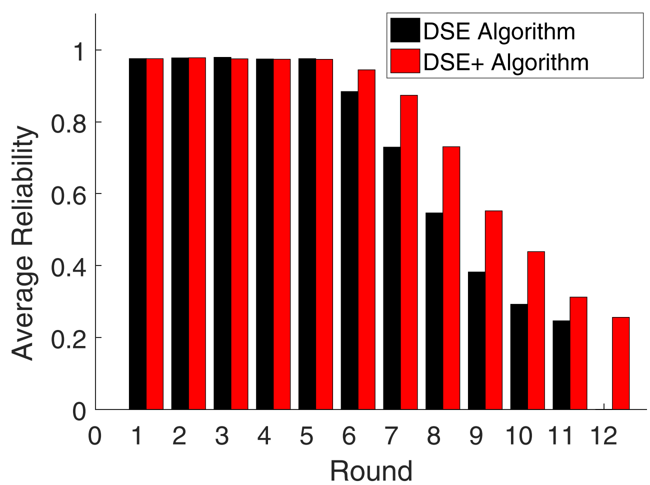
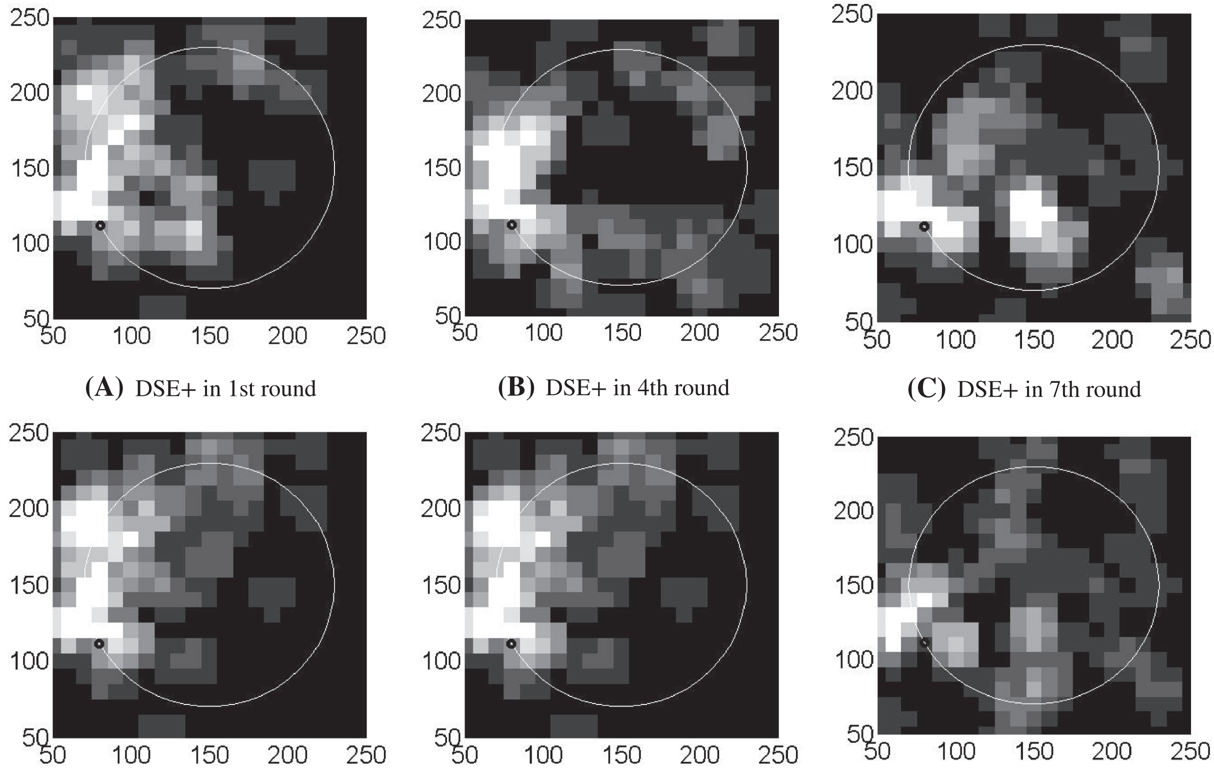


FIGURE 7 Average reliability w.r.t. number of round

total number of nodes failures due to depletion of energy in more and more nodes. But even under such condition, it is apparent that the total number of nodes failures in DSE+ is always smaller than that in the DSE algorithm. Thus, DSE + can keep the system away from out of service more efficiently.

In order to evaluate the efficiency of DSE and DSE+, we record the final step of detecting the user in Figure 6. Obviously, the user has moved roughly the same steps when the task is processed in DSE and the DSE+ algorithms. In the last several rounds, the sensitivity of DSE+ is higher than DSE algorithm because of the energy correction (Figure 7).



(A) DSE+ in 1st round **(B)** DSE+ in 4th round **(C)** DSE+ in 7th round

(D) DSE in 1st round **(E)** DSE in 4th round **(F)** DSE in 7th round

FIGURE 8 Gray scale map

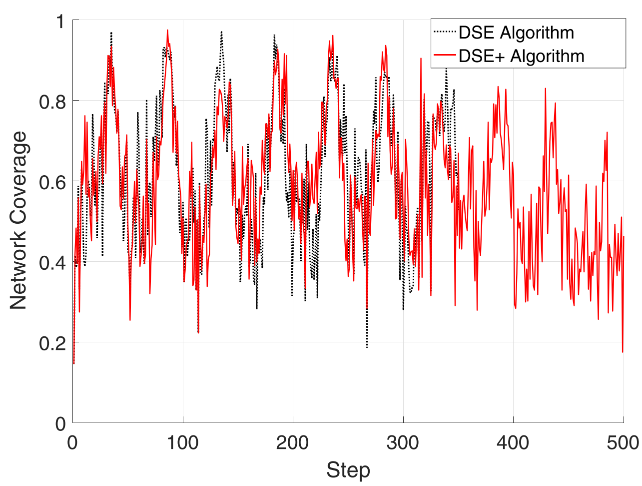


FIGURE 9 Network coverage w.r.t. number of steps

Next, we draw the performance picture of DSE and DSE+ in gray scale map (Figure 8). We record data at the 30th step in 1st, 4th, and 7th rounds and make comparison. Figure 8A–C describes the DSE+ algorithm; Figure 8D–F describes the DSE algorithm. We can see that gray scale maps are similar in both algorithms, which means that reliability is higher in the vicinity of the user while lower in other regions. Meanwhile, nodes activated by DSE in fourth round are more than DSE+, which implies that the energy correction of DSE+ is working and makes awaken nodes as few as possible. In the seventh round, on the contrary, awaken nodes in DSE+ are more than DSE, which proves that the energy correction λ prolong the system life. So swarm intelligence feature is well maintained by DSE+.

Figure 9 depicts the network coverage of DSE and DSE+. We can see that performance of DSE and DSE+ is roughly similar in terms of network coverage in first 350 steps. However, after 350 steps, DSE+ still maintains its good performance, whereas the system backed up by DSE is out of service. This evidence indicates that DSE+ is able to function effectively in a distributed manner.

5 | CONCLUSION

In edge computing, how to save energy among massive sustainable edge nodes is still needed to be settled especially in a distributed manner. Meanwhile, unbalanced energy consumption is a disadvantage in most of the biologically inspired methods. To solve them, inspired by the ant optimization, we propose two algorithms DSE and DSE+ to wake-up nodes. We first put forward a way to calculate the wake-up probability of a node. Then we use a correction factor to alleviate the impact of energy skewed distribution problem. This factor is used to increase the wake-up probability of the edge nodes with relatively high energy. Simulations demonstrate that DSE+ solves the problem of unbalanced energy consumption effectively. Moreover, it improves service reliability in the later stages and prolongs the service time of edge nodes.

Compared with the existing solutions, DSE and DSE+ have many advantages. First, they allow for distributed implementation without requiring a centralized control by the coordinator. Thus, the problems caused by master failures can be avoided and the additional cost of master election is saved. Second, the pheromone is temporally and spatially accumulated for ON/OFF control so that the proposed methods are robust to false alarms, which are inevitable in decentralized computing due to the complex environment and limited resource of each node. Third, the proposed methods do not require node localization. For network uncertainty, the proposed methods are suitable for facing the massive edge node scenarios such as a school and business center.

Future work aims at how to efficiently provision service among multi-users.

ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China under grant no. 72004147, 62173158 and 61803391, the Scientific Technological Innovation Foundation of Shunde Graduate School, USTB under grant no. BK19CF010 and BK20BF012 and the National Key Research and Development Program of China under grant no. 2016YFC0901303.

DATA AVAILABILITY STATEMENT

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

ORCID

Jianpeng Qi https://orcid.org/0000-0001-6150-9773

REFERENCES

1. Chen D, Tower C & Ye W 5G power: Creating a green grid that slashes costs, emissions and energy use. https://www.huawei.com/en/publications/communicate/89/5g-power-green-grid-slashes-costs-emissions-energy-use; 2020.
2. Hao K. Training a single ai model can emit as much carbon as five cars in their lifetimes. MIT Technol Rev. 2019. https://www.technologyreview.com/2019/06/06/239031/training-a-single-ai-model-can-emit-as-much-carbon-as-five-cars-in-their-lifetimes
3. Jiang C, Fan T, Gao H, et al. Energy aware edge computing: A survey. *Comput Commun*. 2020;151:556-580.
4. Hu P, Dhelim S, Ning H, Qiu T. Survey on fog computing: Architecture, key technologies, applications and open issues. *J Netw Comput Appl*. 2017;98:27-42. https://www.sciencedirect.com/science/article/pii/S1084804517302953

5. Carvalho G, Cabral B, Pereira V, Bernardino J. Computation offloading in edge computing environments using artificial intelligence techniques. *Eng Appl Artif Intell*. 2020;95:103840. https://www.sciencedirect.com/science/article/pii/S0952197620302050
6. Aral A, Brandic I. Dependency mining for service resilience at the edge. In *2018 IEEE/ACM Symposium on Edge Computing (SEC)*. Seattle, WA, USA: IEEE; 2018:228-242.
7. Elazhary H. Internet of things (IOT), mobile cloud, cloudlet, mobile IOT, IOT cloud, fog, mobile edge, and edge emerging computing paradigms: Disambiguation and research directions. *J Netw Comput Appl*. 2019;128:105-140. https://www.sciencedirect.com/science/article/pii/S1084804518303497
8. Dorigo M, Birattari M, Stutzle T. Ant colony optimization. *IEEE Comput Intell Mag*. 2006;1(4):28-39.
9. Shokouhifar M. FH-ACO: Fuzzy heuristic-based ant colony optimization for joint virtual network function placement and routing. *Appl Soft Comput*. 2021;107:107401. https://doi.org/10.1016/j.asoc.2021.107401
10. Chen W, Wang D, Li K. Multi-user multi-task computation offloading in green mobile edge cloud computing. *IEEE Trans Serv Comput*. 2019;12(5):726-738.
11. Pu L, Chen X, Xu J, Fu X. D2D fogging: An energy-efficient and incentive-aware task offloading framework via network-assisted D2D collaboration. *IEEE J Sel Areas Commun*. 2016;34(12):3887-3901.
12. Chen X, Pu L, Gao L, Wu W, Wu D. Exploiting massive D2D collaboration for energy-efficient mobile edge computing. *IEEE Wirel Commun*. 2017;24(4):64-71.
13. Sun X, Ansari N. Green cloudlet network: A sustainable platform for mobile cloud computing. *IEEE Trans Cloud Comput*. 2020;8(1):180-192.
14. Li Y, Orgerie A-C, Rodero I, Parashar M, Menaud J-M. Leveraging renewable energy in edge clouds for data stream analysis in IOT. In *2017 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*. Madrid, Spain: IEEE; 2017:186-195.
15. Feng M, Mao S, Jiang T. Base station on-off switching in 5G wireless networks: Approaches and challenges. *IEEE Wirel Commun*. 2017;24(4):46-54.
16. Allai I, Senouci S-M, Penhoat J, Gourhant Y. A new sustainable mechanism to wake-up bast stations in mobile networks. In *2016 Global Information Infrastructure and Networking Symposium (GIIS)*. Porto, Portugal: IEEE; 2016:1-6.
17. Nguyen C, Mehta A, Klein C, Elmroth E. Why cloud applications are not ready for the edge (yet). In *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*. New York, NY, USA: ACM Press; 2019:250-263.
18. Xiang C, Wang X, Chen Q, et al. No-jump-into-latency in China's internet! In *Proceedings of the International Symposium on Quality of Service - IWQoS '19*. New York, NY, USA: ACM Press; 2019:1-10.
19. Parzysz F, Gourhant Y. Drastic energy reduction with GDTX in low cost 5G networks. *IEEE Access*. 2018;6:58171-58181.
20. Ning Z, Kong X, Xia F, Hou W, Wang X. Green and sustainable cloud of things: Enabling collaborative edge computing. *IEEE Commun Mag*. 2019;57(1):72-78.
21. Zheng Y, Qu J, Yu H. Research on energy-saving power system of office building lighting based on internet of things. *J Eng*. 2020;2020(2):52-57. https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/joe.2019.0825
22. Liang Y, Cao J, Zhang L, Wang R, Pan Q. A biologically inspired sensor wakeup control method for wireless sensor networks. *IEEE Trans Syst Man Cybern Part C (Appl Rev)*. 2010;40(5):525-538.
23. Hedetniemi S, Laskar R. Bibliography on domination in graphs and some basic definitions of domination parameters. *Discrete Math*. 1990;86(1):257-277. https://doi.org/10.1016/0012-365X(90)90365-O
24. Gaotao S, Minghong L. Stochastic sleeping for energy-conserving in large wireless sensor networks. *J Comput Res Develop*. 2006;43(4):579.

How to cite this article: Qi J, Ren S, Wang B, Wang R. A distributed swarm intelligence-based energy-saving method among massive edge nodes. *Int J Commun Syst*. 2021;34(17):e4974. doi:10.1002/dac.4974