Complexity

Green Computing in Complex Systems

Lead Guest Editor: Yongsheng Hao
Guest Editors: Chen Wang and Guanfeng Liu

Green Computing in Complex Systems

Complexity

Green Computing in Complex Systems

Lead Guest Editor: Yongsheng Hao
Guest Editors: Chen Wang and Guanfeng Liu

Contents

Hindawi
Complexity
Volume 2021, Article ID 9994680, 6 pages
https://doi.org/10.1155/2021/9994680

Research Article

Energy-Optimal 3D Path Planning for MAV with Motion Uncertainty

Yamin Li,[1,2,3] **Bowen Sun**,[1] **Ping Xia**,[2,3,4] **and Yang Yang**[1]

[1]*School of Computer Science and Information Engineering, Hubei University, Wuhan 430062, China*
[2]*Hubei Key Laboratory of Intelligent Vision Based Monitoring for Hydroelectric Engineering, China Three Gorges University, Yichang 443002, China*
[3]*Yichang Key Laboratory of Intelligent Vision Based Monitoring for Hydroelectric Engineering, China Three Gorges University, Yichang 443002, China*
[4]*College of Computer and Information Technology, China Three Gorges University, Yichang 443002, China*

Correspondence should be addressed to Yang Yang; yangyang@hubu.edu.cn

Received 8 March 2021; Accepted 23 September 2021; Published 18 October 2021

Academic Editor: Guanfeng Liu

Practical applications of microaerial vehicle face significant challenges including imprecise localization, limited on-board energy, and motion uncertainty. This paper focuses on the latter two issues. The core of proposed energy-optimal path planning algorithm is an energy consumption model deriving from real measurements of a specific quadrotor and utilizing a 2D Gaussian distribution function to simulate the uncertainty of random drift. Based on these two models, we formulate the optimal path traversing the 3D map with minimum energy consumption using a heuristic ant colony optimization. Multiple sets of contrast experiments demonstrate the effectiveness and efficiency of the proposed algorithm.

1. Introduction

Microaerial vehicle (MAV) has shown great application potentials in both military and civilian fields such as security surveillance, aerial photography, and medical escort [1–4]. While the flight endurance of MAV is limited relating to its minisize and possible on-board fuel/battery that can be carried. And the MAV may unpredictably deviate from the planned path due to the random drift or aerodynamic interferences. Therefore, the practical applications of MAV face significant challenges including imprecise localization, motion uncertainty, and limited energy. In this paper, we aim to plan an energy-optimal path for MAV under mission with motion uncertainty.

The research of efficient energy path planning for the MAV has received increasing attention. In the studies of traveling salesman problem (TSP), effective routes are chosen to reduce the total path length [5]. Yet, this method does not take into account the orientations of the MAV. Moreover, minimum path length does not always correlate to minimum energy consumption. Franco et al. proposed an energy-aware path planning algorithm based on a real energy measurement of unmanned aerial vehicle (UAV) in different velocities and operating conditions [6]. However, the energy consumption for different flight motions is assumed to be constant, which is unrealistic. A different approach has been taken by Al-Sabban et al. [7], who developed an energy-efficient path planning algorithm by using the available environmental wind energy within the medium where the UAV is operating to extend the flight endurance.

Most of the current path planning methods, such as the above algorithms, only focus on the path planning problems in 2-dimensional (2D) plane and unrealistically assume that the MAV can accurately move to the target point without any motion error according to the control commands [8]. The orientation of the MAV after each move is not considered either. Different from these methods, this paper proposes an energy-optimal 3-dimensional (3D) path planning algorithm that minimizes the energy consumption while also considering the motion uncertainty.

The contributions of this paper are as follows. (1) An energy consumption model is established deriving from real measurements of a specific MAV, characterizing the relationship between energy consumption and particular flying motions. (2) Instead of correcting motion deviations continuously, we exploit the trend of movement drift, and a 2D Gaussian distribution function is employed to simulate the random drift and position bias. (3) A heuristic procedure fused with the Gaussian distribution is proposed to search the energy-optimal path traversing a 3D map using a modified ant colony optimization (ACO) algorithm.

2. System Model

As shown in Figure 1, this paper considers a MAV flying in indoor testbed deployed with 3D visual sensor network (VSN) composed of multiple RGB-D sensors. To completely cover the test space, a number of RGB-D sensors (C_1, C_2,...), which are Microsoft Kinect sensors [9], are installed. The testbed with obstacles is abstracted as a meshed cuboid model, which is the planning space. Evenly divide the space into grids and the complete path can be stored as a list of coordinates p (x, y, z). To avoid the obstacles, only grids above the obstacles are searchable.

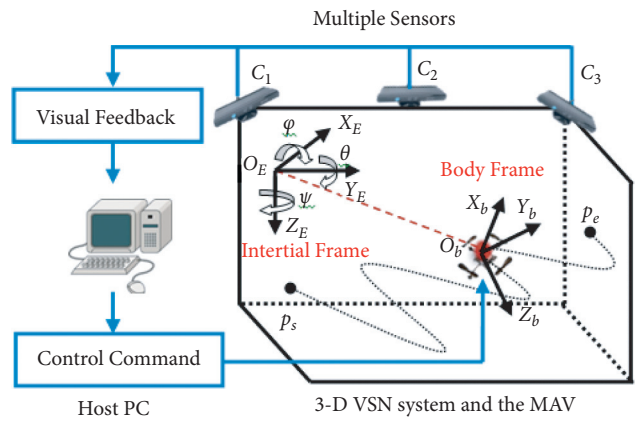


Figure 1: Indoor testbed configuration and the two coordinate systems.

For autonomous flight, the visual feedback concept [10] is employed and two coordinate systems are defined: the inertial frame $O_E X_E Y_E Z_E$ and the body frame $O_b X_b Y_b Z_b$ attached at the center of the quadrotor. The center of mass and the body frame origin are assumed to coincide. The rotation matrix R_b^E from the body frame to the inertial frame is denoted using Euler angles yaw ψ, pitch θ, and roll φ:

$$R_b^E = \begin{bmatrix} \cos\theta\cos\psi & \sin\varphi\sin\theta\cos\psi - \cos\varphi\sin\psi & \cos\varphi\sin\theta\sin\psi + \sin\varphi\sin\psi \\ \cos\theta\sin\psi & \sin\varphi\sin\theta\sin\psi + \cos\varphi\cos\psi & \cos\varphi\sin\theta\sin\psi - \sin\varphi\cos\psi \\ -\sin\theta & \sin\varphi\cos\theta & \cos\varphi\cos\theta \end{bmatrix}. \quad (1)$$

The core problem we address in this paper is to plan an optimal path U_{opt} from the arbitrary starting point p_s to the arbitrary ending point p_e that minimizes the energy consumption.

3. Energy Consumption Estimation

3.1. Energy Consumption Analysis. The energy consumption sources of a MAV mainly include the powertrain, communications, sensors, and control circuits. Table 1 shows a power consumption measurement result of Crazyflie Nano Quadcopter from Bitcraze [11], which is a 19 g mini MAV used in our current system. As can be seen from Table 1, the motor drive consumes more than 85% of its total energy, which is absolutely the main energy consumer.

According to Franco et al. [6], the MAV has a most energy-efficient speed v_0 defined as the speed that minimizes the energy required to cover a given straight path of length. When the MAV is flying straight in a constant speed v, the thrust $\boldsymbol{T}$ is equal to the sum of weight $\boldsymbol{G}$ (payload involved) and the air friction $\boldsymbol{F}_{\text{drag}}$ which approximates a constant, as illustrated in Figure 2. The power consumed by the DC motors $P_M(v)$ under a specific speed condition is also a constant.

Assume that the total weight $\boldsymbol{G}$ and the power consumed by other airborne modules P_0 remain unchanged. We have

$$\mathbf{T}\cos\theta = \mathbf{G}, \quad (2)$$

$$\mathbf{T}\sin\theta = \mathbf{F}_{\text{drag}} = \frac{1}{2}C_d(\theta)v^2 S, \quad (3)$$

$$P_M(v) = \mathbf{T}v\cos\left(\frac{\pi}{2} - \theta\right), \quad (4)$$

where $C_d(\theta) = C_1(1 - \cos^3\theta) + C_2(1 - \sin^3\theta)$ is the drag coefficient, ρ is the air density, and S is the area of the propellers. The energy consumed in the straight flight to cover a distance d at speed v can be computed as

$$E_v = \int_0^{d/v} [P_M(v) + P_0]\mathrm{d}t = [P_M(v) + P_0]\frac{d}{v}. \quad (5)$$

Substituting equations (2)–(4) into equation (5), a function of E_v varies with v is obtained. Then, the most energy-efficient speed v_0 can be found by computing the partial derivative of E_v with respect to v The MAV is controlled to fly at v_0 between each grid.

4. Energy Consumption Model

The flight speed and movements of the MAV determine the energy consumption. Therefore, we characterize the relationship between the energy consumption of a candidate

Table 1: The power consumptions of different modules of Crazyflie.

MCU (%)	Sensors (%)	Wireless communication (%)	Motor drive (%)	Others (%)
9.48	0.46	1.98	87.02	1.06

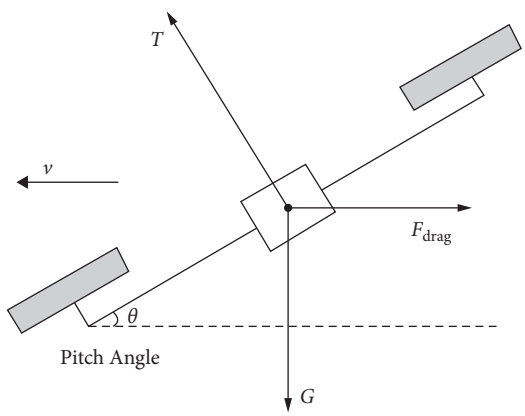


Figure 2: Force analysis of the MAV flying at speed (v).

path U and flight motions of the MAV as an energy consumption model to evaluate U. The energy consumption of U is determined by the following three factors.

(1) Path length: the energy consumption is proportional to the path length. So, the first energy consumption factor s_1 is defined as

$$s_1(U) = d_{\text{proj}}(p_s, p_e), \tag{6}$$

where $d_{\text{proj}}(p_s, p_e)$ is the length of the projection of U on the $O_E X_E Y_E$ plane from p_s to p_e, as illustrated in Figure 3.

(2) Climbing and descending rate: as shown in Figure 3, the climbing and descending rate is defined as the change of the altitude of the MAV on Z_e-dimension over time. The energy consumption increases as the rate increases. Therefore, the second energy consumption factor s_2 is denoted as

$$s_2(U) = \max\left|\frac{\Delta z}{\Delta t}\right| = \max\left|\frac{z(p_{i+1}) - z(p_i)}{t(p_i, p_{i+1})}\right|. \tag{7}$$

(3) Turning angle: a sharp turning is undesirable as turnings often associated with decelerations [12], which are both time and energy consuming. The turning angle is defined as the angle formed by any three consecutive points on the projection of U on the OEXEYE plane, as illustrated in Figure 3. The third energy consumption factor s3 is expressed as

$$s_3(U) = 180^\circ - \min \angle p_{i-1}{}' p_i' p_{i+1}{}'. \tag{8}$$

The energy consumption model $f_{\text{eng}}(U)$ is defined as a weighted sum of the three aforementioned factors and the optimal path U_{opt} should have the energy consumption model being minimized. Such that

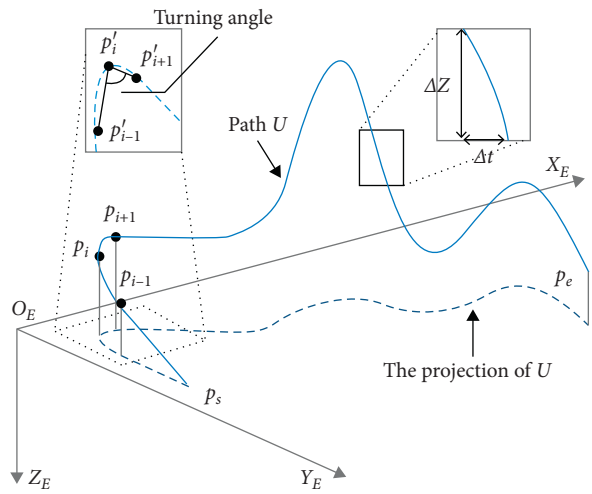


Figure 3: An illustration of (U) and the three desirability factors.

$$U_{\text{opt}} = \arg\min f_{\text{eng}}(U) = \arg\min\left(\frac{\omega_1 s_1 + \omega_2 s_2 + \omega_3 s_3}{\omega_1 + \omega_2 + \omega_3}\right). \tag{9}$$

where ω_1, ω_2, and ω_3 are the energy consumption tuning parameters. The energy consumption model will assist in achieving path with energy optimization.

5. Path Planning with Motion Uncertainty

Instead of correcting the motion deviations constantly in the control phase, we deal with the problem of the motion uncertainty of a flying MAV in advance, that is, in the process of path planning. Different from a traditional ACO algorithm [13] whose artificial ants are only allowed to select the vertexes of the grids as their next hop, in the proposed path planning procedure, all the artificial ants are given "drift" characteristics which allow them to select any point on the circumference of a drift circle with a radius of R centered at the searchable vertex of the grid as their next hop. This method is developed to simulate the motion uncertainty of the flying MAV and also give the ants extraflexibility in making routing decisions.

As illustrated in Figure 4, the main direction of the path planning is set along the X_E-axis. In each iteration, q artificial ants with drift characteristics are released at the starting point p_s. Except for the ending point p_e, these q ants are free to choose any point on the drift circles. The coordinates of current location p_i of the ant are (x_i, y_i, z_i) and there are nine searchable vertexes whose coordinates are (x_{i+1}, y_c, z_c) on the searchable plane $\prod_{i+1}$ paralleling to the $O_E Y_E Z_E$-plane. Here, x_{i+1} is determined by the grid size. The drift circles are on the searchable plane $\prod_{i+1}$ centered at (x_{i+1}, y_c, z_c). The

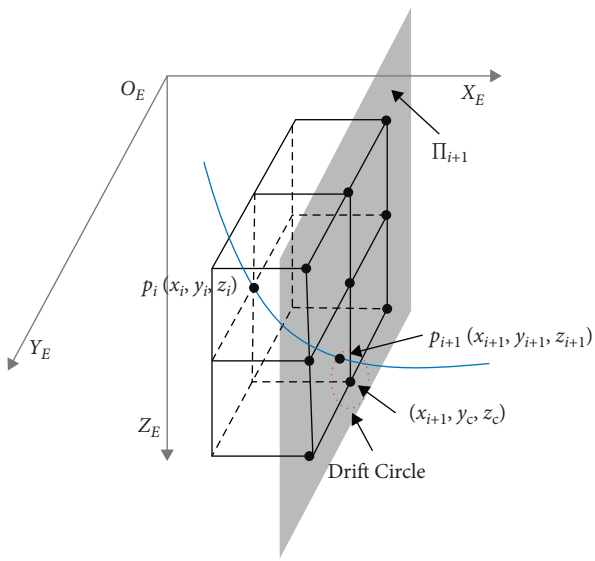


Figure 4: The searchable plane and drift circle.

probability of choosing point $p_{i+1}(x_{i+1}, y_{i+1}, z_{i+1})$ on the drift circles is

$$P(p_{i+1}) = \frac{f_h(x_{i+1}, y_{i+1}, z_{i+1}) f_\tau(x_{i+1}, y_{i+1}, z_{i+1})}{\sum f_h(x_{i+1}, y_{i+1}, z_{i+1}) f_\tau(x_{i+1}, y_{i+1}, z_{i+1})}. \quad (10)$$

The next point of path U is chosen using the Roulette method according to the probabilities of all the searchable points on the drift circle.

Here, $f_h(x_{i+1}, y_{i+1}, z_{i+1})$ is the heuristic function defined as

$$f_h(x_{i+1}, y_{i+1}, z_{i+1}) = S \times (\lambda_1 D + \lambda_2 Q + \lambda_3 Z + \lambda_4 T), \quad (11)$$

where $S = \{0, 1\}$ is the safety factor indicating whether the next point is reachable, D is the distance between p_{i+1} and p_i, Q is the distance between p_{i+1} and p_e, Z is the altitude difference in Z_E-axis between p_{i+1} and p_i, T is the turning angle from p_i to p_{i+1}, and $\lambda_1, \ldots, \lambda_4$ are the coefficients.

$f_\tau(x_{i+1}, y_{i+1}, z_{i+1})$ is the distribution of the pheromone concentration on the circumferences of the drift circles. Once the ant reaches p_e, the complete path is evaluated by the energy consumption model $f_{\text{eng}}(U)$ and the pheromone will be distributed on the drift circles that have been visited, which is in the form of 2D Gaussian distribution function given by

$$f_\tau(x_{i+1}, y, z) = A \exp\left(-\left(\frac{(y - y_c)^2}{2\sigma_y^2} + \frac{(z - z_c)^2}{2\sigma_z^2}\right)\right), \quad (12)$$

where $\sigma_y = \sigma_z$ are the variances along Y_E-axis and Z_E-axis, respectively, since only symmetric 2D Gaussian functions are considered in the proposed method. A is the amplitude of its pheromone concentration whose initial value is inversely proportional to its energy consumption $f_{\text{eng}}(U)$. By the end of this iteration, A will be updated as follows to reinforce the pheromone concentration of the shorter path:

$$A_{n+1} = (1 - \varepsilon) A_n + \varepsilon \frac{K}{f_{\text{eng}}(U_n)}, \quad (13)$$

where n is the number of iterations, K is a coefficient, and $0 < \varepsilon < 1$ is the update factor.

When all the q ants reach p_e, on each searchable vertex, there might be $k (0 \le k \le q)$ 2D Gaussian distribution functions. These k Gaussian functions will superpose and form a joint distribution function on each searchable plane $\prod_{i+1}$ and update the pheromone concentration on the circumferences of the drift circles. In the following iterations, the ants will select their moving direction based on the pheromone residue on the drift circles.

6. Simulations and Results

6.1. Simulation Settings. To evaluate the performance of the proposed method, we compare the proposed energy-optimal path planning method with the traditional ACO-based path planning method. Using only path length $s_1(U)$ as the evaluation factor, the traditional ACO-based path planning algorithm neither employs the energy consumption model $f_{\text{eng}}(U)$ nor has the drift characteristic. The purposed energy-optimal path planning method is evaluated in two cases which have the difference of whether the drift characteristic is fused.

The experiments are applied in the testbed introduced in the section system model and performed in the MATLAB R2012a. The size of the planning space is 6m × 6 m × 3 m, and it is evenly divided into 20 × 20 × 10 grids. The Crazyflie starts from point (0, 2, 4) and ends at point (20, 6, 6) at a speed of 0.5 m/s which is close to the most energy-efferent speed in the current configuration. Parameters used in the simulations are listed in Table 2. It is worth noting that the tuning parameters ω_1, ω_2, and ω_3 of the energy consumption model, the coefficients of the proposed energy-optimal path planning algorithm such as λ_1, λ_2, λ_3, and λ_4 in equation (11), the variances σ_y and σ_z in equation (12), and even the number of the artificial ants can significantly impact the simulation results. Therefore, the parameters' list in Table 2 are reasonably determined by experimental analyses based on uniform design [14], which is used to convert the problem of parameter establishment into the experimental design of multifactor and multilevel and reduces the work load of experiment greatly of simulation. The number of the iteration of each algorithm is 100, and in each iteration, 10 artificial ants are released into the 3D map.

6.2. Performance Analysis. Each algorithm is run for 40 times individually. Figure 5 displays an example of two paths planned by the traditional ACO-based path planning algorithm which is displayed in red line and the proposed energy-optimal path planning method without drift characteristic which is displayed in blue line. As can be observed visually, both the two path have successfully avoided the obstacles in the 3D map. However, the blue path planned by the proposed method is smoother with less climbings, descendings, and sharp turns than the red one planned by the traditional ACO algorithm.

Table 2: Parameters used in the simulation.

Parameters	Traditional ACO	Proposed energy-optimal method	
		Without drift characteristic	With drift characteristic
Grid size	$20 \times 20 \times 10$	$20 \times 20 \times 10$	$20 \times 20 \times 10$
Turning parameters (ω_1, ω_2, ω_3)	4, 0, 0	4, 2, 2	4, 2, 2
Coefficients (λ_1, λ_2, λ_3, λ_4)	50, 50, 0, 0	50, 50, 30, 10	50, 50, 30, 10
Radius (R)	N/A	N/A	0.5
Variances (σ_y, σ_z)	N/A	N/A	0.05, 0.05
Coefficients (K)	100	100	100
Update factor (ε)	0.2	0.2	0.2

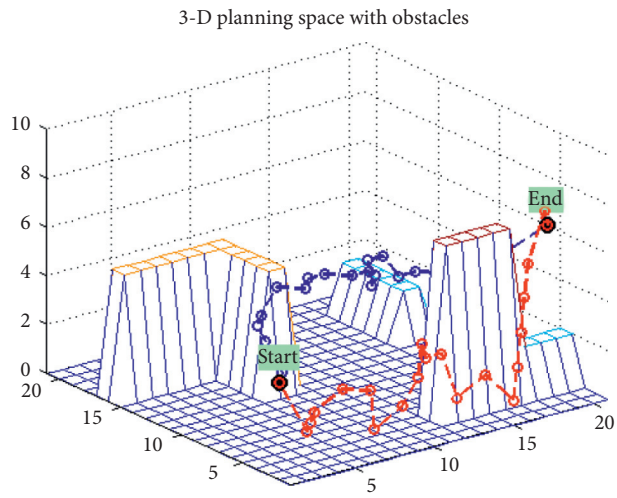


Figure 5: The planned paths of the two methods.

Table 3: Simulation results of the two methods.

Evaluations	Traditional ACO	Proposed energy-optimal method Without drift characteristic	With drift characteristic
Averaged path length/m	10.30	9.64	9.94
Averaged climbing and descending rate/m/s	0.23	0.14	0.16
Averaged minimum turning angle	41.46°	47.39°	43.94°

Table 4: Performances of the proposed algorithm in different grid sizes.

Grid size	$20 \times 20 \times 10$	$40 \times 40 \times 20$
Averaged path length (m)	9.642	9.186
Averaged climbing and descending rate (m/s)	0.138	0.117
Averaged minimum turning angle	47.39°	48.62°
Computing time (s)	3.802	31.122

The average results of the repeated simulations are presented in Table 3. As can be seen, the proposed energy-optimal path planning method shows greatly improved performance over the traditional ACO algorithm in all three evaluation factors, especially the one without simulating the drift characteristic. It outperforms the traditional ACO algorithm by reducing the path length and climbing and descending rate by more than 6.4% and 39.1%, and increasing the minimum turning angle by more than 14.3%.

Such improvements are achieved by employing the energy consumption model $f_{\text{eng}}(U)$ during the optimization process. The one with drift characteristic also has better performance over the traditional ACO algorithm. However, compared with the one without drift characteristic, the planned path slightly degrades, which is reasonable and unavoidable when the ants are given the ability to randomly deviate from the planned path.

6.3. Grid Size Experiments. In addition, we verify the performances of the proposed algorithm when the planning space is divided into $20 \times 20 \times 10$ grids or $40 \times 40 \times 20$ grids.

As displayed in Table 4, the proposed path planning algorithm in finer division space shows improved performance in terms of all three evaluation factors, which means better accuracy and lower power consumption. However, the computing time is many times longer than the former, which means worse real-time performance. Therefore, appropriate grid size should be set to weigh these two factors for practical application.

7. Conclusions

This paper proposes an energy-optimal 3D path planning algorithm for indoor flying MAV which deals with the problem of motion uncertainty in the process of path planning and relaxes the limitation of making routing decisions. Numerical experiments results demonstrate its effectiveness and optimal performances. Our algorithm reduces the path length and climbing and descending rate by more than 6.4% and 39.1%, increasing the minimum turning angle by more than 14.3%. For future work, extension of this method to plan missions for multiple MAVs is considered.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (62002104), the Hubei Province Natural Science Foundation of China (2018CFC900 and 2019CFB191), the 2020 Opening fund for Hubei Key Laboratory of Intelligent Vision Based Monitoring for Hydroelectric Engineering (2020SDSJ06), and the Construction fund for Hubei Key Laboratory of Intelligent Vision Based Monitoring for Hydroelectric Engineering (2019ZYYD007).

References

[1] W. He, T. Meng, X. He, and C. Sun, "Iterative learning control for a flapping wing micro aerial vehicle under distributed disturbances," *IEEE Transactions on Cybernetics*, vol. 49, no. 4, pp. 1524–1535, 2019.

[2] J. Sun, J. Song, H. Chen, X. Huang, and Y. Liu, "Autonomous state estimation and mapping in unknown environments with onboard stereo camera for micro aerial vehicles," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 5746–5756, 2020.

[3] S. Vemprala and S. Saripalli, "Monocular vision based collaborative localization for micro aerial vehicle swarms," in *Proceedings of the 2018 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 315–323, Dallas Marriott, TX, USA, June 2018.

[4] Z. Xiao, X. Dai, H. Jiang et al., "Vehicular task offloading via heat-aware MEC cooperation using game-theoretic method," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 2038–2052, 2020.

[5] J. D. C. Little, K. G. Murty, D. W. Sweeney, and C. Karel, "An algorithm for the traveling salesman problem," *Operations Research*, vol. 11, no. 6, pp. 972–989, 1963.

[6] C. D. Franco and G. Buttazzo, "Energy-aware coverage path planning of UAVs," in *Proceedings of the 2015 IEEE International Conference on Autonomous Robot System and Competitions*, pp. 111–117, Portugal, April 2015.

[7] W. H. Al-Sabban, L. F. Gonzalez, and R. N. Smith, "Wind-energy based path planning for unmanned aerial vehicles using markov decision processes," in *Proceedings of the 2013 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 784–789, Karlsruhe, Germany, May 2013.

[8] H. Jiang, J. Li, P. Zhao, F. Zeng, Z. Xiao, and A. Iyengar, "Location privacy-preserving mechanisms in location-based services," *ACM Computing Surveys*, vol. 54, no. 1, pp. 1–36, 2021.

[9] Meet Kinect for Windows: https://developer.microsoft.com/en-us/windows/kinect.

[10] H.-M. Chuang, D. He, and A. Namiki, "Autonomous target tracking of UAV using high-speed visual feedback," *Applied Sciences*, vol. 9, no. 21, p. 4552, 2019.

[11] The Crazyflie Nano Quadcopter: https://www.bitcraze.io/crazyflie/.

[12] N. Ganganath and C. T. Cheng, "A 2-dimensional ACO-based path planner for off-line robot path planning," in *Proceedings of the 2013 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, pp. 302–307, Beijing, China, October 2013.

[13] M. Dorigo, *Optimization, learning and natural algorithms*, Ph.D. dissertation, 1992.

[14] Y. Q. Huang, C. Y. Liang, and X. D. Zhang, "Parameter establishment of an ant system based on Uniform design," *Control and Decision*, vol. 21, no. 1, pp. 93–96, 2006.

Hindawi
Complexity
Volume 2021, Article ID 7321719, 10 pages
https://doi.org/10.1155/2021/7321719

Research Article

Ultrahigh-Dimensional Model and Optimization Algorithm for Resource Allocation in Large-Scale Intelligent D2D Communication System

Minxin Liang,[1] **Jiandong Liu**,[1] **Jinrui Tang**,[2] **and Ruoli Tang**[3]

[1]*Guangzhou Power Supply Bureau, Guangdong Power Grid Co., Ltd., Guangzhou, China*
[2]*School of Automation, Wuhan University of Technology, Wuhan, China*
[3]*School of Energy and Power Engineering, Wuhan University of Technology, Wuhan, China*

Correspondence should be addressed to Ruoli Tang; ruolitang@hotmail.com

Received 7 June 2021; Revised 19 July 2021; Accepted 27 July 2021; Published 4 August 2021

Academic Editor: Chen Wang

The optimal resource allocation in the large-scale intelligent device-to-device (D2D) communication system is of great importance for improving system spectrum efficiency and ensuring communication quality. In this study, the D2D resource allocation is modelled as an ultrahigh-dimensional optimization (UHDO) problem with thousands of binary dimensionalities. Then, for efficiently optimizing this UHDO problem, the coupling relationships among those dimensionalities are comprehensively analysed, and several efficient variable-grouping strategies are developed, i.e., cellular user grouping (CU-grouping), D2D pair grouping (DP-grouping), and random grouping (R-grouping). In addition, a novel evolutionary algorithm called the cooperatively coevolving particle swarm optimization with variable-grouping (VGCC-PSO) is developed, in which a novel mutation operation is introduced for ensuring fast satisfaction of constraints. Finally, the proposed UHDO-based allocation model and VGCC-PSO algorithm as well as the grouping and mutation strategies are verified by a comprehensive set of case studies. Simulation results show that the developed VGCC-PSO algorithm performs the best in optimizing the UHDO model with up to 6000 dimensionalities. According to our study, the proposed methodology can effectively overcome the "curse of dimensionality" and optimally allocate the resources with high accuracy and robustness.

1. Introduction

Due to the fast development of cellular communication networks, the complexity of net structure and user number explosively increases in recent years. As a result, the device-to-device (D2D) communication technology is developed and plays an increasingly important role in the modern 5G cellular networks [1, 2]. However, due to the fast growing cellular users and high requirement for quality of service (QoS), the lack of spectrum resource becomes one of the main reasons which severely restricts the development of modern communication network [3, 4].

In the D2D communication system, two types of user equipment nearby (namely, D2D-pair and DP) can directly communicate under the control of enhanced-Node B (eNB) [5]. However, the direct communication of DP always requires to reuse the physical resource blocks (PRBs) of the traditional cellular users (CUs). In order to obtain promising performance on spectrum efficiency and communication quality, the aforementioned PRB of CU should be optimally allocated to each DP. As a result, the resource allocation model and optimization algorithm are hot topics in the field of D2D communication in recent years [6–9]. For example, Li et al. developed a nonconvex mixed-integer nonlinear programming (MINLP) problem-based model to minimize the mobile power consumption to obtain efficient resource allocation solution and also ensured the QoS and high communication rate at the same time [10]. Su et al. proposed an approach to maximize the total D2D groups capacity by considering the requirement of QoS and energy causality

constraints. Simulation results verified the effectiveness of their methodology [11]. In order to reduce the cross-tier interference in D2D communication, Khazali et al. proposed a fractional frequency reuse (FFR)-based spectrum partitioning scheme. In Khazali et al.'s study, they also modelled the spectral efficiency as an optimization problem, which can be effectively solved by employing iterative algorithms [12]. Mohamad et al. proposed a dynamic sectorization method in which eNB can vary the number of sectors dynamically and allocated the resource block to D2D users. According to their study, the signal-to-interference-noise-ratio (SINR) and the network overall performance were improved [13]. Amin et al. proposed a resource allocation algorithm based on the so-called Q-learning, in which the multiagent learners from multiple D2D users were created, and the system throughput was determined by the state-learning of Q value list. According to their study, the system throughput was effectively maximized by controlling the D2D users' power, and a fine QoS of cellular users was also ensured [14].

In this study, the resource allocation problem in an intelligent D2D communication system with large number of users is addressed. To be specific, by describing the communication constraints as penalty functions, the aforementioned resource allocation is modelled as a binary optimization problem with ultrahigh dimensionality, called the binary large-scale global optimization (BLSGO) problem in this study. Then, considering the consequent "curse of dimensionality," the coupling relationships among the thousands of dimensionalities are comprehensively analysed and some efficient variable-grouping strategies are developed, i.e., the cellular user grouping (CU-grouping), D2D pair grouping (DP-grouping), and random grouping (R-grouping). In addition, a novel swarm-intelligence-based algorithm, namely, cooperatively coevolving particle swarm optimization with variable-grouping (VGCC-PSO) is developed, in which an efficient mutation operation is also introduced for rapidly escaping the punishment of penalty function and speeding up the convergence process. Finally, the proposed model and optimization methodologies are comprehensively verified by case studies.

The contributions of this paper can be summarized as follows. Firstly, the BLSGO-based resource allocation model for the intelligent D2D communication system is established. Secondly, the variable-grouping strategies including the CU-grouping, DP-grouping, and R-grouping are developed. Finally, a novel VGCC-PSO algorithm is proposed and employed to optimize the aforementioned BLSGO-based model.

The rest of this paper is organized as follows. In Section 2, the ultrahigh dimensional resource allocation model is developed. The corresponding constraints and penalty functions as well as the encoding scheme for defining optimization vector are also discussed in this section. In Section 3, the variable-coupling relationships are comprehensively analysed, and the VGCC-PSO with different variable-grouping strategies and mutation operation is developed. Then, in Section 4, the effectiveness of the proposed variable-grouping strategies and mutation operation is tested. In Section 5, the proposed model and optimization methodologies are verified by a comprehensive set of case studies. Finally, this paper is concluded in Section 6.

2. Ultrahigh-Dimensional Resource Allocation Model

2.1. Resource Allocation in D2D Communication. In an intelligent D2D communication system, the allocation of PRB is of great importance for improving system spectrum efficiency and ensuring communication quality. In order to optimally allocate the CU resources to DP, an efficient offline model for evaluating the cost of each allocation solution is required [15].

For a cellular network of LTE-advance systems, assume the eNB locates at the center of a region, in which all the CUs and DPs are randomly distributed. Denote $C = \{CU_n | n = 1, 2, \ldots, N\}$ as the set of CU, and denote $D = \{DP_m | m = 1, 2, \ldots, M\}$ as the set of DP. The resource allocation principle employed in this study is defined as follows: on the one hand, the PRB of each CU should be reused by only one DP; on the other hand, each DP can reuse more than one CUs' PRB (but at least one). Schematic of the evaluated D2D communication system is illustrated as Figure 1.

In the D2D communication system, an efficient resource allocation solution is to maximize the system energy efficiency by allocating all the CUs' PRB to each DP while satisfying some constraints. The energy efficiency to be maximized can be formulized as follows:

$$\eta_e = \frac{\sum_{m=1}^{M} \sum_{n=1}^{N} x_{m,n} \cdot R_{m,n}}{\sum_{m=1}^{M} \sum_{n=1}^{N} x_{m,n} \cdot P_{m,n} + P_c}, \tag{1}$$

where η_e represents the system energy efficiency; $x_{m,n}$ is the binary variable, $x_{m,n} = 1$ denotes the PRB of CU_n which is reused by DP_m, $x_{m,n} = 0$ denotes the opposite, $R_{m,n}$, which is formulized as equation (2), represents the transmission speed of DP_m when reusing the PRB of CU_n, $P_{m,n}$ represents the transmission power of DP_m when reusing the PRB of CU_n; and P_c represents the circuit power consumption of DP_m.

$$R_{m,n} = \log_2 \left(1 + \frac{P_{m,n} \cdot H_m}{P_n \cdot H_{n,m} + n_0} \right), \tag{2}$$

where H_m represents the channel gain from DP transmitter DT_m to DP receiver DR_m; P_n represents the transmission power of CU_n; $H_{m,n}$ represents the channel gain from DT_m to CU_n; and n_0 represents the channel noise power under the effect of white Gaussian noise.

According to reference [16], the maximization of energy efficiency η_e is equal to the minimization of the following equation:

$$f_{\min} = \sum_{m=1}^{M} \sum_{n=1}^{N} x_{m,n} \cdot \frac{H_{n,m} \cdot H_{m,n}}{H_m \cdot H_n}, \tag{3}$$

where H_n represents the channel gain from CU_n to eNB and $H_{n,m}$ represents the channel gain from CU_n to DR_m. The channel gains here (H_n, H_m, $H_{n,m}$, and $H_{m,n}$) are all calculated as $H = 10^{(-\mathrm{PL}-\mathrm{SHD})/10}$, where PL represents the path

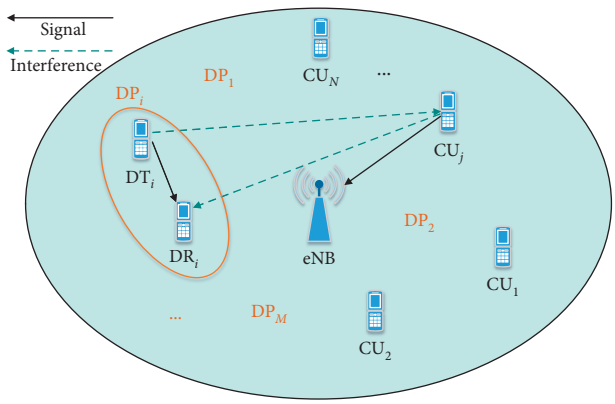


Figure 1: Schematic of the D2D communication system.

loss and SHD represents the lognormal fading between each transmitter and receiver.

As mentioned above, the PRB of each CU should be reused by only one DP and each DP should be allocated with one or more CUs' PRB. As a result, the overall allocation model can be formulized as the following constrained optimization problem:

$$\min_{x_{m,n}} f = \sum_{m=1}^{M}\sum_{n=1}^{N} x_{m,n}\cdot\frac{H_{n,m}\cdot H_{m,n}}{H_m\cdot H_n}$$
$$s.t.\begin{cases} x_{m,n}\in\{0,1\} \\ \sum_{m=1}^{M} x_{m,n} = 1, \quad (n=1,2,\ldots,N) \\ \sum_{n=1}^{N} x_{m,n}\geq 1, \quad (m=1,2,\ldots,M). \end{cases} \tag{4}$$

2.2. Encoding Scheme. As shown in equation (4), the parameters to be optimized for obtaining optimal resource allocation are the $N\times M$ binary variables $x_{m,n}$ ($m=1, 2, \ldots, M$; $n=1, 2, \ldots, N$). In this study, the direct encoding scheme is employed, i.e., all the $N\times M$ binary variables $x_{m,n}$ are directly employed to combine the optimization vector $\vec{x}$, which is formulized as follows:

$$\vec{x} = \left(x_{1,1}, x_{1,2}, \ldots, x_{1,N}, \ldots, x_{M,1}, x_{M,2}, \ldots, x_{M,N}\right). \tag{5}$$

Note that each $x_{m,n}$ ($m=1, 2, \ldots, M$; $n=1, 2, \ldots, N$) in equation (5) is a binary variable, which indicates whether CU$_n$ is reused by DP$_m$. Obviously, dimensionality of the optimization vector $\vec{x}$ is equal to $N\times M$. This implies that the problem dimensionality (or complexity) will become extremely high when N and M are large. For example, assume that a D2D communication system contains 100 CU (i.e., $N=100$) and 20 DP (i.e., $M=20$). Then, dimensionality of the model will become $N\times M=2000$. Obviously, complexity of this 2000-dimensional problem is extremely high because of the "curse of dimensionality." As a result, an effective optimization algorithm is required to solve this BLSGO problem.

Note that in this study, the continuous evolutionary algorithm, namely, VGCC-PSO is developed and employed to optimize the variables listed in equation (5). As a result, all the binary variables $x_{m,n}$ are bounded within the interval [0, 100]. The encoding scheme is defined as follows: for the variable $x_{m,n}$, encode $x_{m,n}$ to 0 when it belongs to [0, 50) in VGCC-PSO; otherwise, when $x_{m,n}$ belongs to [50, 100] in VGCC-PSO, encode it to 1.

2.3. Ultrahigh-Dimensional Model for D2D Resource Allocation. In order to optimize the variables in equation (5) using swarm-intelligence-based algorithms, the constraints listed in equation (4) are transformed into penalty function in our model. That is to say, the constrained problem shown in equation (4) is transformed into an unconstrained problem by defining the following penalty function:

$$f_p = \lambda\cdot\left(N_1+N_2\right), \tag{6}$$

where λ represents the penalty factor which is used to control the penalty intensity; N_1 denotes the number of CU which does not satisfy the second constraint in equation (4), i.e., $\sum_{m=1}^{M} x_{m,n} = 1$, ($n=1, 2, \ldots, N$); and N_2 denotes the number of DP which does not satisfy the third constraint in equation (4), i.e., $\sum_{n=1}^{N} x_{m,n}\geq 1$, ($m=1, 2, \ldots, M$). Note that for a certain solution $\vec{x}$, the values of N_1 and N_2 can be calculated by decoding each dimensionality of $\vec{x}$.

By introducing the penalty function, the overall resource allocation model can be formulized as

$$\min_{x_{m,n}} f = \sum_{m=1}^{M}\sum_{n=1}^{N} x_{m,n}\cdot\frac{H_{n,m}\cdot H_{m,n}}{H_m\cdot H_n} + \lambda\cdot\left(N_1+N_2\right). \tag{7}$$

3. Optimization Methodology

In the field of numerical optimization, different kinds of optimization methodologies are developed and employed in solving real-world engineering problems, e.g., the linear programming methods [17], neural network methods [18], evolutionary algorithms [19], and so on. In this study, the cooperatively coevolving algorithms are developed for solving the aforementioned UHDO-based resource allocation model.

3.1. Cooperatively Coevolving. The cooperatively coevolving (CC) is a general algorithm framework proposed for solving the high-dimensional optimization problem [20–22]. In basic CC, the D-dimensional problem is decomposed into several subproblems based on the philosophy of "divide and conquer." Each of these low-dimensional subproblems is solved by a certain algorithm in turn. Then, a D-dimensional individual, namely, context vector is defined to connect these subproblems and ensure the coevolving process. The CC framework has been integrated with different evolutionary algorithms and obtained promising performance on solving high-dimensional problems [20, 23, 24]. Principle of the basic CC framework can be illustrated as the following steps:

(1) For a D-dimensional problem P, initialize the D-dimensional population with N_P individuals. Then, decompose the original problem P into K subproblems SP_i ($i = 1, 2, \ldots, K$), i.e., $P = [SP_1, SP_2, \ldots, SP_K]$. Note that the dimensionality of each subproblem SP_i is equal to D/K. For a D-dimensional individual x, $x = (x^1, x^2, \ldots, x^K)$, where x^i represents the corresponding variables that belong to the ith subproblem.

(2) Define the context vector as the current global best individual y. Then, the ith subproblem in CC is defined as

$$\min f^i(x, y), \quad x \in R^S, \tag{8}$$

where $f^i(x, y) = f(y^1, \ldots, y^{i-1}, x, y^{i+1}, \ldots, y^K)$ and R^S represents the solution space. Start an evolution circle, in which all the subproblems are optimized with a certain algorithm. The context vector y is updated in every iteration.

(3) Proceed another cycle if the stopping criteria are not satisfied; otherwise, stop the cooperative coevolution.

Note that, in CC framework, in order to decrease the complexity of high-dimensional problem, the original problem is decomposed into several less difficult subproblems to be solved separately. According to reference [25], the basic CC framework is effective only if any two subproblems have no interaction. In another word, the variable-grouping strategy (means the subordinate relationship between variables and subproblems) significantly affects the performance of CC.

3.2. Variable-Grouping Strategy. In order to effectively optimize the ultrahigh-dimensional problem using CC, the coupled (or called nonseparable) variables should be grouped into the same subproblem and coevolved for enough iterations [25]. With regard to the resource allocation model as listed in equation (4), all the optimization variables $x_{m,n}$ ($m = 1, 2, \ldots, M$; $n = 1, 2, \ldots, N$) are grouped using the following strategies.

3.2.1. Random Grouping (R-Grouping). In R-grouping, all the variables are randomly disorganized and grouped into different subproblems. To be specific, flow of R-grouping mechanism is as the following steps:

(i) Firstly, orders of the entire $D = N \times M$ dimensionalities in the original model are randomly disorganized.

(ii) Secondly, these disorganized dimensionalities are decomposed into $K = D/s$ sub-problems. Obviously, each subproblem has s dimensionalities, where the group size s is randomly generated within a predefined set S.

(iii) Finally, the group size s is dynamically changed during the coevolving process as the following principle: for each coevolving iteration, randomly selected a new s in S if the global optimum is not updated; otherwise, keep the current s value unchanged.

Schematic of the R-grouping mechanism is illustrated in Figure 2.

3.2.2. Cellular User Grouping (CU-Grouping). As discussed in Section 2.2, each optimization variable represents the reusing relationship between a certain CU_n and a certain DP_m. In CU-grouping, the variables reflecting the relationships between one certain CU_n and every DP are grouped into a subproblem and are employed to coevolve for enough iterations. To be specific, the variables for CU_1, i.e., $x_{1,1}, x_{2,1}, \ldots, x_{M,1}$, are regarded as the first subproblem, then the variables for CU_2, i.e., $x_{1,2}, x_{2,2}, \ldots, x_{M,2}$, are regarded as the second subproblem, and so on. Note that in CU-grouping, as each subproblem (or called group) has M dimensionalities, i.e., $s = M$, the number of subproblems K is equal to $D/s = N$.

Schematic of the CU-grouping mechanism is illustrated in Figure 3.

3.2.3. D2D Pair Grouping (DP-Grouping). Similarly, in DP-grouping, the variables reflecting the relationships between every CU_n and one certain DP are grouped into a subproblem. To be specific, the variables for DP_1, i.e., $x_{1,1}, x_{1,2}, \ldots, x_{1,N}$, are regarded as the first subproblem, then the variables for DP_2, i.e., $x_{2,1}, x_{2,2}, \ldots, x_{2,N}$, are regarded as the second subproblem, and so on. Note that in DP-grouping, as each subproblem has N dimensionalities, i.e., $s = N$, the number of subproblems K is equal to $D/s = M$.

Schematic of the DP-grouping mechanism is illustrated in Figure 4.

3.3. Mutation Operation. According to the encoding scheme developed in Section 2.2, it can be easily concluded that the binary variable $x_{m,n}$ is not related to its specific value within the solution space but is directly decided by whether it is greater than the boundary 50. As a result, the encoding scheme will significantly increase the solution space and complicate the original model. In order to overcome this problem, a novel mutation operation is developed and imposed on all the context vectors of VGCC-PSO.

As discussed in Section 2, the PRB of each CU should be reused by only one DP, and each DP can reuse more than one CUs' PRB (but at least one). That is to say, in a feasible solution, there is at most one variable in $x_{1,n}, x_{2,n}, \ldots, x_{M,n}$ ($n = 1, 2, \ldots, N$) which is greater than 0 at each time (denoted as Constraint I). In addition, there is at least one variable in $x_{m,1}, x_{m,2}, \ldots, x_{m,N}$ ($m = 1, 2, \ldots, M$) which is greater than 0 at each time (denoted as Constraint II). As Constraints I and II are closely related to the entire variables, most of the solutions in solution space will be infeasible because of these constraints. In order to reduce the model complexity caused by these so many infeasible solutions, the feasible solutions satisfying Constraint I and Constraint II are directly

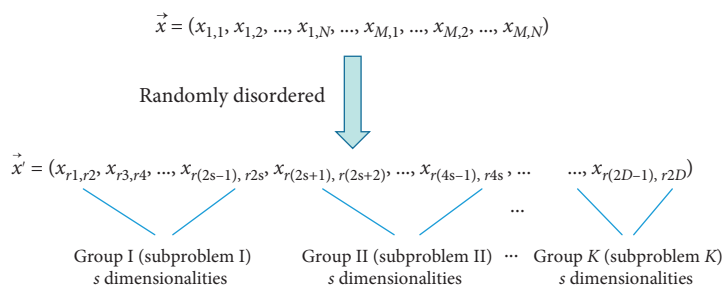


Figure 2: Schematic of R-grouping.

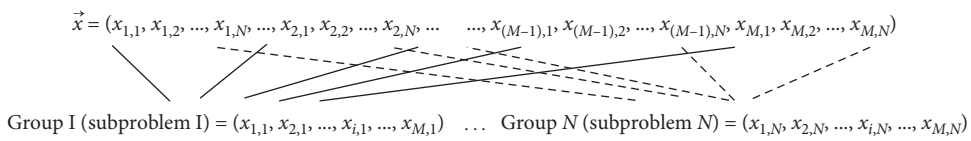


Figure 3: Schematic of CU-grouping.

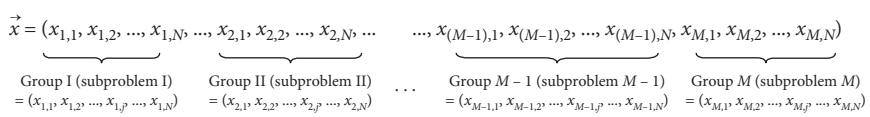


Figure 4: Schematic of DP-grouping.

employed as the context vector and inserted into VGCC-PSO population. The proposed mutation operation for context vector is illustrated as the following steps:

Step 1. Define parameters P_{m1} and P_{m2} satisfying $P_{m1} \le P_{m2}$ and $P_{m1}, P_{m2} \in [0, 1]$ to control the mutation probabilities.

Step 2. In each iteration, for each of the context vectors in VGCC-PSO, say the ith context vector in the tth iteration $CV_i(t)$, randomly generate a mutation variable $P_i(t)$ with the interval [0, 1]. Then, mutate $CV_i(t)$ according to the following principles:

(i) If $P_i(t) < P_{m1}$, keep $CV_i(t)$ unchanged

(ii) If $P_{m1} \le P_i(t) < P_{m2}$, each of the components $[x_{1,n}, x_{2,n}, \ldots, x_{M,n}]$ $(n = 1, 2, \ldots, N)$ is randomly mutated to $[r_0, r_0, \ldots, r_0, r_1]$, $[r_0, \ldots, r_0, r_1, r_0]$, $\ldots$, $[r_0, r_1, r_0, \ldots, r_0]$ and $[r_1, r_0, r_0, \ldots, r_0]$, in which each r_0 is randomly generated within [0, 50) and each r_1 is randomly generated within [50, 100]

(iii) Otherwise (i.e., $P_i(t) \ge P_{m2}$), check each of the components $[x_{m,1}, x_{m,2}, \ldots, x_{m,N}]$ $(m = 1, 2, \ldots, M)$ if all the variables are lower than 50, i.e., $x_{m,1}, x_{m,2}, \ldots, x_{m,N}$ are encoded to 0, then randomly set one variable (e.g., $x_{m,n}$) to r_1

Step 3. Denote the mutated context vector $CV_i(t)$ as $CV_{i\text{-mut}}(t)$. Update $CV_i(t)$ using $CV_{i\text{-mut}}(t)$ if better.

Note that the proposed mutation mechanism is only imposed on the context vectors rather than the entire individuals in current population. The reason is that, on the one hand, the mutated context vectors can better guide the other individuals to rapidly satisfy the constraints and thus significantly reduce the scope of solution space. On the other hand, the population diversity will be not destroyed in mutation process.

3.4. VGCC-PSO Algorithm. In this study, a novel evolutionary algorithm, namely, VGCC-PSO is developed for optimizing the BLSGO-based allocation model. In VGCC-PSO, the aforementioned variable-grouping strategies and mutation operation are integrated for overcoming the ultrahigh dimensionality characteristic of the model.

In VGCC-PSO, the basic CC framework described in Section 3.1 is imposed on the PSO algorithm. Then, the R-grouping, CU-grouping, and DP-grouping mechanisms are randomly selected in each iteration as the following steps:

Step 1. Set the selection probabilities P_R (for R-grouping), P_{CU} (for CU-grouping), and P_{DP} (for DP-grouping), satisfying

$$\begin{cases} P_R, P_{\text{CU}}, P_{\text{DP}} \in [0, 1], \\ P_R + P_{\text{CU}} + P_{\text{DP}} = 1. \end{cases} \tag{9}$$

Step 2. In each iteration, randomly generate a variable $P_g \in [0, 1]$. Then, select a grouping strategy according to the following equation:

$$\begin{cases} \text{select } R\text{-grouping}, & \text{if} \quad 0 \le p_g < P_R, \\ \text{select CU-grouping}, & \text{if} \quad P_R \le p_g < P_R + P_{\text{CU}}, \\ \text{select DP-grouping}, & \text{if} \quad P_R + P_{\text{CU}} \le p_g \le 1. \end{cases} \tag{10}$$

In addition, in VGCC-PSO, the mutation operation described in Section 3.3 is imposed on each of the context vectors in each iteration. Pseudo code of VGCC-PSO is given in Algorithm 1.

4. Verification for Variable-Grouping Strategies and Mutation Operation

In this section, efficiency of the proposed variable-grouping strategies and mutation operation is verified by simulation experiments. In the following simulation, the numbers of CU and DP are set to 80 and 20, respectively. The location of each CU, DT, and DR is randomly generated within a hexagon region with the radius of 700 m. As suggested in reference [5], the communication model and parameters employed in the following simulation are chosen in accordance with 3GPP LTE regulation for the OFDMA system. In addition, the path loss model is defined as follows:

$$\begin{cases} \text{eNB-UE: PL} = 33.65 + 23.47 \log 10(d[m]), \\ \text{UE-UE: PL} = 36.67 + 19.54 \log 10(d[m]). \end{cases} \tag{11}$$

According to the encoding scheme described in Section 2.2, the model dimensionality D in this case is equal to $N \times M = 1600$. In this section, the VGCC-PSO algorithm is employed for optimizing this 1600-dimensional problem. In order to verify effectiveness of the proposed variable-grouping strategy and mutation operation, the basic PSO without CC framework, variable-grouping strategy, and mutation operation (denoted as PSO), the CC-based PSO (i.e., without variable-grouping strategy and mutation operation, denoted as CCPSO), the CC-based PSO only integrated with mutation operation (i.e., without variable-grouping strategy, denoted as $\text{CCPSO}_{\text{mut}}$), and the CC-based PSO only integrated with variable-grouping strategy (i.e., without mutation operation, denoted as CCPSO_{vg}) are employed for comparison.

For all the compared algorithms, the dynamic group size S in R-grouping is set as $S = \{10, 20, 50, 100, 200\}$. According to our numerical experiments, the selection probabilities for variable-grouping strategies P_R, P_{CU}, and P_{DP} are suggested to be set as follows: $P_R = 0.4$, $P_{\text{CU}} = 0.3$, and $P_{\text{DP}} = 0.3$. The penalty factor λ in equation (6) is set to 10000. The probabilities in mutation operation P_{m1} and P_{m2} are set to 0.3 and 0.6, respectively. The maximum number of fitness evaluations ($FE_{\max}$) is set to 1×10^6. The population size is set to 50, and the number of context vectors is set to 5. Results of the simulation experiments are compared in Figure 5 and Table 1.

According to the simulation results, the proposed VGCC-PSO algorithm performs the best and obtains the minimum fitness function value of 12.7027. Compared with the CC-based algorithms, the basic PSO without CC framework fails to optimize the 1600-dimensional problem and obtains the worst performance of 631.2110, which is significantly larger than its competitors. This implies that by decomposing the original problem into several low-dimensional subproblems, the CC framework is very efficient on overcoming the ultrahigh dimensional characteristic.

By integrating the CC framework, the result obtained by CCPSO is significantly better than that of PSO but worse than the best performer VGCC-PSO. Obviously, the gap between CCPSO and VGCC-PSO shows the efficacy of the developed variable-grouping strategies and mutation operation. To be specific, by integrating the variable-grouping strategies, the CCPSO_{vg} obtains the final result of 17.4469, which significantly outperforms 49.3261 obtained by CCPSO. Rationale behind the achievement is that by integrating the CU-grouping and DP-grouping strategies, the coupled variables are given higher probability to be grouped into the same subproblem and coevolved for enough iterations.

By integrating the mutation operation, $\text{CCPSO}_{\text{mut}}$ obtains the final result of 27.9705, which outperforms 49.3261 obtained by CCPSO. In addition, according to the convergence graph shown in Figure 5, the mutation operation can help the algorithm to fast satisfy the constraints and avoid the punishment brought by penalty function. As discussed above, the developed variable-grouping strategies and mutation operation are efficient on improving the performance of the evolutionary algorithm on solving ultrahigh dimensional problem. To be specific, the variable-grouping strategies can improve the global exploration ability and optimization accuracy, while the mutation operation can significantly accelerate the convergence or satisfaction speed of constraints.

5. Simulation Experiments and Analysis

In this section, the performance of VGCC-PSO is empirically evaluated on a comprehensive set of case studies. Parameters setting of VGCC-PSO is the same with Section 4. In addition, some state-of-the-art evolutionary algorithms are employed for comparison, including the CPSO-S_{K} [26], CPSO-$\text{S}_{\text{K-rg-aw}}$ [27], CCPSO2 [28], and CCDE [29]. Parameter settings of these algorithms are following their original studies. The detailed model parameters for each case are listed in Table 2.

Simulation results of the case studies are listed in Table 3, in which the best performance is set in bold. The convergence graphs for each case are plotted in Figure 6.

As shown in Table 3, the proposed VGCC-PSO obtains the best performance for all the cases. To be specific, for the low-dimensional models (i.e., Cases 1 and 2), the outperformance of VGCC-PSO is not significant compared with that of CCDE, CCPSO2, and CPSO-$\text{S}_{\text{K-rg-aw}}$. However, when scale of the D2D communication system becomes large (i.e., the numbers of CU and DP increase to several decades and the model dimensionality increases to more than 1000 in Cases 3 to 6), VGCC-PSO can obtain its efficiency and significantly outperforms the competitors because of the integration of variable-grouping mechanism and mutation operation. For example, in Cases 3 and 4, VGCC-PSO obtains the final results of 18.8794 and 13.1455 for the 1200-dimensional and 2000-dimensional problems, respectively. However, the results obtained by other algorithms are 35.0793 and 23.2162 for CCDE, 30.0329 and 27.0283 for CCPSO2, and 130.2828 and 115.9234 for CPSO-

Algorithm: VGCC-PSO
Initialize a D-dimensional population with N_P particles. Initialize p context vectors with the best p particles.
repeat
    Randomly generate a variable P_g, then select a variable-grouping strategy using equation (10).
    Decompose the original optimization vector into K subproblems according to the selected variable-grouping strategy.
    Denote the jth subproblem as P_j.
    for *each subproblem* P_j **do**
        Coevolve the corresponding dimensionalities of P_j using the CC-based PSO as discussed in our previous work [22].
    end
    Update the personal best of each particle, and update the context vectors according to reference [22].
    for *each context vector* CV_i **do**
        Mutate CV_i to $CV_{i\text{-mut}}$ according to the principles developed in Section 3.3.
        if $f(CV_{i\text{-mut}}) < f(CV_i)$ **then**
            Update CV_i using $CV_{i\text{-}mut}$.
    end
    Update the global best with the best context vector.
until the stopping criteria are satisfied

Algorithm 1: Pseudo code of VGCC-PSO.

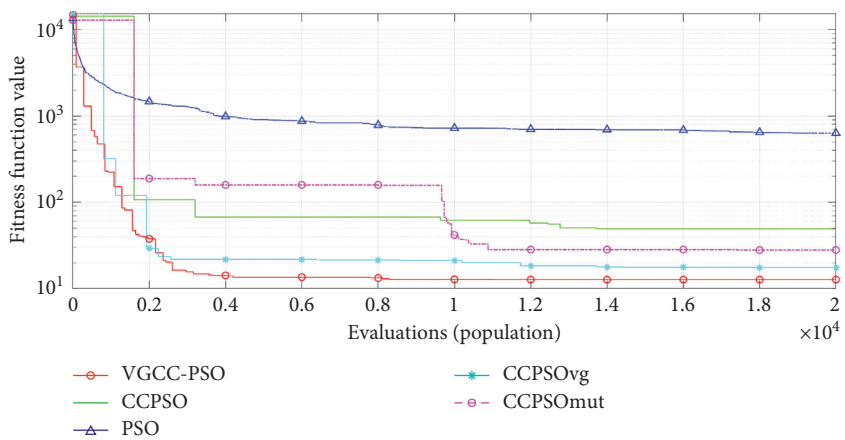


Figure 5: Convergence graphs for optimizing the 1600-dimensional model.

Table 1: Optimization results of the compared algorithms.

Algorithm	Optimization result
VGCC-PSO	12.7027
CCPSO	49.3261
PSO	631.2110
$CCPSO_{vg}$	17.4469
$CCPSO_{mut}$	27.9705

Table 2: Parameter settings of resource allocation models.

Case number	Number of CU	Number of DP	Model dimensionality	Dynamic group size in R-grouping
Case 1	30	8	240	{5, 10, 12, 20, 40}
Case 2	50	10	500	{5, 10, 20, 25, 50}
Case 3	60	20	1200	{10, 20, 50, 100, 200}
Case 4	80	25	2000	{20, 50, 100, 200, 400}
Case 5	100	40	4000	{20, 50, 100, 200, 500}
Case 6	120	50	6000	{50, 100, 200, 600, 1000}

Table 3: Simulation results for Case 1 to 6.

Algorithm	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
VGCC-PSO	**10.5182**	**10.0771**	**18.8794**	**13.1455**	**30.0673**	**48.7768**
CCDE	22.9087	21.6069	35.0793	23.2162	42.0547	87.0291
CCPSO2	24.2608	34.4517	30.0329	27.0283	96.0441	77.4215
CPSO-S_K	146.3635	264.6194	613.8953	431.4242	2098.5629	1602.9015
CPSO-$S_{K\text{-rg-aw}}$	24.1810	62.8635	130.2828	115.9234	315.2044	321.8840

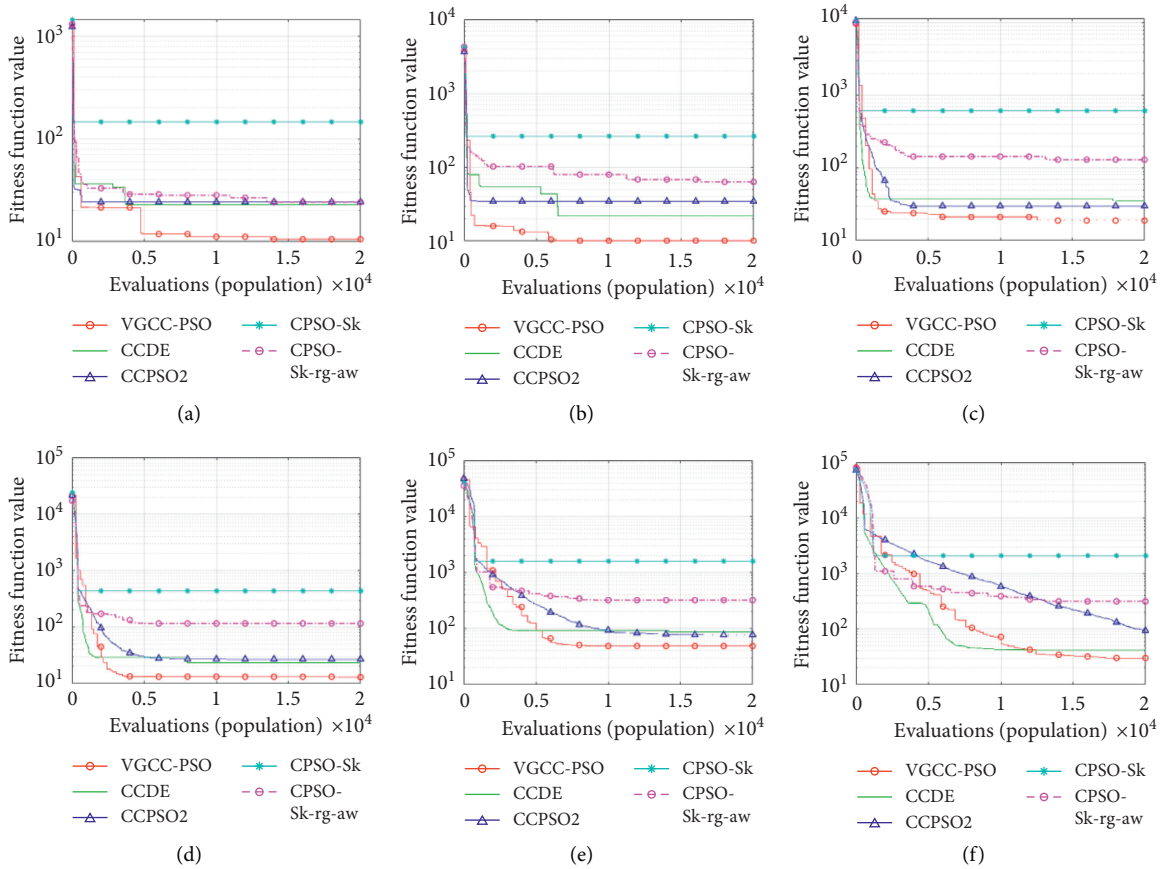


Figure 6: Convergence graphs for Case 1 to 6: (a) Case 1; (b) Case 2; (c) Case 3; (d) Case 4; (e) Case 5; (f) Case 6.

$S_{K\text{-rg-aw}}$. Note that the results obtained by CPSO-S_K for Cases 3 and 4 are 613.8953 and 431.4242, which implies that CPSO-S_K losses its efficacy and fails to optimize these ultrahigh dimensional problems.

In Cases 5 and 6, the model dimensionality increases to 4000 and 6000, and VGCC-PSO is still able to effectively optimize the model and obtains the final results of 30.0673 and 48.7768, which significantly outperforms 42.0547 and 87.0291 by CCDE, 96.0441 and 77.4215 by CCPSO2, 315.2044 and 321.8840 by CPSO-$S_{K\text{-rg-aw}}$, and 2098.5629 and 1602.9015 by CPSO-S_K. Note that the final results listed in Table 3 are all lower than the penalty factor λ which is set to 10000 in the simulations. This indicates that because of the application of CC framework, all the algorithms can satisfy the constraints and avoid punishment for each of the ultrahigh-dimensional cases.

6. Conclusion

In this study, the ultrahigh dimensional model for resource allocation in a large-scale intelligent D2D communication system is established, and a novel optimization methodology, namely, VGCC-PSO is also developed for optimizing the BLSGO-based model.

For a large-scale D2D communication system with N CUs and M DPs, by defining the binary encoding scheme and penalty function, the resource allocation problem is modelled as an unconstrained optimization problem with ultrahigh dimensionalities of $N \times M$. In order to effectively optimize the ultrahigh dimensional model, the CC framework is applied to decompose the original problem and coevolve each subproblem according to the philosophy of "divide and conquer."

For further improving the optimization performance, some efficient variable-grouping strategies like the R-grouping, CU-grouping, and DP-grouping are developed to rearrange and co-optimize the large number of optimization variables. In addition, a novel mutation operation is also developed to accelerate the convergence speed. Simulation results show the effectiveness of these algorithm mechanisms; the integration of variable-grouping strategies can improve global exploration ability and final optimization accuracy, while the mutation operation can significantly accelerate the satisfaction of constraints.

Finally, by integrating the CC framework, variable-grouping strategies, and mutation operation, the proposed VGCC-PSO algorithm is empirically evaluated on a comprehensive set of case studies, and some state-of-the-art algorithms are also employed for comparison. Simulation results show that VGCC-PSO performs the best in optimizing the ultrahigh-dimensional model with up to 6000 dimensionalities. In a word, the proposed methodology can effectively overcome the "curse of dimensionality" and optimally allocate the resources in the large-scale intelligent D2D communication system with high accuracy and robustness.

Data Availability

The prior studies and data are cited at relevant places within the text as references [17–19, 21, 22, 25].

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the Science and Technology Project of China Southern Power Grid Co., Ltd (GZHKJXM20190097).

References

[1] I. Loannou, V. Vassiliou, C. Christophorou, and A. Pitsillides, "Distributed aritificial intelligence solution for D2D communication in 5G networks," *IEEE Systems Journal*, vol. 14, no. 3, pp. 4232–4241, 2020.

[2] W. K. Lai, C.-S. Shieh, F.-S. Chou, C.-Y. Hsu, and M.-H. Shen, "Handover management for D2D communication in 5G networks," *Applied Sciences*, vol. 10, no. 12, p. 4409, 2020.

[3] M. Hayati, H. Kalbkhani, and M. G. Shayesteh, "Energy-efficient relay selection and power allocation for multi-source multicast network-coded D2D communications," *AEU—International Journal of Electronics and Communications*, vol. 128, Article ID 153522, 2021.

[4] Y. P. Llerena and P. R. L. Gondim, "Social-aware spectrum sharing for D2D communication by artificial bee colony optimization," *Computer Networks*, vol. 183, Article ID 107581, 2020.

[5] F. Jiang, B. Wang, C. Sun, Y. Liu, and R. Wang, "Mode selection and resource allocation for device-to-device communications in 5G cellular networks," *China Communications*, vol. 13, no. 6, pp. 32–47, 2016.

[6] S. Jayakumar and S. Nandakumar, "A review on resource allocation techniques in D2D communication for 5G and B5G technology," *Peer-to-Peer Networking and Applications*, vol. 14, no. 1, pp. 243–269, 2020.

[7] Y. Yang, L. Feng, C. Zhang, Q. Ou, and W. Li, "Resource allocation for virtual reality content sharing based on 5G D2D multicast communication," *EURASIP Journal on Wireless Communications and Networking*, vol. 2020, no. 1, p. 112, 2020.

[8] G. Hou and L. Chen, "D2D communication mode selection and resource allocation in 5G wireless networks," *Computer Communications*, vol. 155, pp. 244–251, 2020.

[9] M. Liu and L. Zhang, "Graph colour-based resource allocation for relay-assisted D2D underlay communications," *IET Communications*, vol. 14, no. 16, pp. 2701–2708, 2020.

[10] R. Li, P. Hong, K. Xue, M. Zhang, and T. Yang, "Energy-efficient resource allocation for high-rate underlay D2D communications with statistical CSI: a one-to-many strategy," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 4, pp. 4006–4018, 2020.

[11] N. Su, Q. Zhu, and Y. Wang, "Resource allocation algorithm for NOMA-enhanced D2D communications with energy harvesting," *Mobile Information Systems*, vol. 2020, Article ID 4062487, 11 pages, 2020.

[12] A. Khazali, S. Sobhi-Givi, H. Kalbkhani, and M. G. Shayesteh, "Energy-spectral efficient resource allocation and power control in heterogeneous networks with D2D communication," *Wireless Networks*, vol. 26, no. 1, pp. 253–267, 2020.

[13] N. M. V. Mohamad, P. Ambastha, S. Gautam, R. Jain, H. Subramaniyam, and L. Muthukaruppan, "Dynamic sectorization and parallel processing for device-to-device (D2D) resource allocation in 5G and B5G cellular network," *Peer-to-Peer Networking and Applications*, vol. 14, no. 1, pp. 296–304, 2020.

[14] A. Amin, X.-H. Liu, I. Khan, P. Uthansaku, M. Forsat, and S. Sajad Mirjavadi, "A robust resource allocation scheme for device-to-device communications based on Q-learning," *Computers, Materials & Continua*, vol. 65, no. 2, pp. 1487–1505, 2020.

[15] J. Dai, J. Liu, Y. Shi, S. Zhang, and J. Ma, "Analytical modeling of resource allocation in D2D overlaying multihop multichannel uplink cellular networks," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 8, pp. 6633–6644, 2017.

[16] X. W. Li and W. K. Liu, "Particle swarm optimization based energy efficiency maximizing strategy in device-to-device (D2D) communications," *Telecommunication Engineering*, vol. 57, no. 10, pp. 1171–1176, 2017.

[17] R. Tang, Z. Wu, and X. Li, "Optimal operation of photovoltaic/battery/diesel/cold-ironing hybrid energy system for maritime application," *Energy*, vol. 162, pp. 697–714, 2018.

[18] R. Tang, Q. Lin, J. Zhou et al., "Suppression strategy of short-term and long-term environmental disturbances for maritime photovoltaic system," *Applied Energy*, vol. 259, Article ID 114183, 2020.

[19] R. Tang, X. Li, and J. Lai, "A novel optimal energy-management strategy for a maritime hybrid energy system based on large-scale global optimization," *Applied Energy*, vol. 228, pp. 254–264, 2018.

[20] R. S. Lan, Y. Zhu, H. M. Lu, Z. L. Tang, Z. B. Liu, and X. N. Luo, "Large-scale optimisation via cooperatively coevolving competition swarm optimiser," *Enterprise Information Systems*, vol. 14, no. 9-10, pp. 1439–1456, 2020.

[21] R.-l. Tang and X. Li, "Adaptive multi-context cooperatively coevolving in differential evolution," *Applied Intelligence*, vol. 48, no. 9, pp. 2719–2729, 2018.

[22] R. L. Tang, Z. Wu, and Y. J. Fang, "Adaptive multi-context cooperatively coevolving particle swarm optimization for large-scale problems," *Soft Computing*, vol. 21, no. 6, pp. 4735–4754, 2017.

[23] F.-S. Hsieh and Y.-H. Guo, "A discrete cooperatively coevolving particle swarm optimization algorithm for combinatorial double auctions," *Applied Intelligence*, vol. 49, no. 11, pp. 3845–3863, 2019.

[24] S. M. J. A. Tabatabaee and H. Zamiri-Jafarian, "Prototype filter design for FBMC systems via evolutionary PSO algorithm in highly doubly dispersive channels," *Transactions on Emerging Telecommunications Technologies*, vol. 28, no. 4, p. e3048, 2017.

[25] R. Tang, Q. An, F. Xu et al., "Optimal operation of hybrid energy system for intelligent ship: an ultrahigh-dimensional model and control method," *Energy*, vol. 211, Article ID 119077, 2020.

[26] F. vandenBergh and A. P. Engelbrecht, "A cooperative approach to particle swarm optimization," *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 3, pp. 225–239, 2004.

[27] X. D. Li and X. Yao, "Tackling high dimensional nonseparable optimization problems by cooperatively coevolving particle swarms," in *Proceedings of the IEEE Congress on Evolutionary Computation*, vol. 1–5, pp. 1546–1553, Trondheim, Norway, May 2009.

[28] X. D. Li and X. Yao, "Cooperatively coevolving particle swarms for large scale optimization," *IEEE Transactions on Evolutionary Computation*, vol. 16, no. 2, pp. 210–224, 2012.

[29] C. H. Chen and W. H. Chen, "Cooperatively coevolving differential evolution for compensatory neural fuzzy networks," in *Proceedings of the International Conference on Fuzzy Theory and Its Applications*, pp. 264–267, Taipei, Taiwan, December 2013.

Hindawi
Complexity
Volume 2021, Article ID 5661292, 12 pages
https://doi.org/10.1155/2021/5661292

Research Article

Distributed Typhoon Track Prediction Based on Complex Features and Multitask Learning

Yongjiao Sun,[1] Yaning Song,[1] Baiyou Qiao,[1] and Boyang Li[2]

[1]*School of Computer Science and Engineering, Northeastern University, Shenyang 110819, China*
[2]*School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China*

Correspondence should be addressed to Yaning Song; 1901780@stu.neu.edu.cn

Received 30 April 2021; Accepted 4 July 2021; Published 13 July 2021

Academic Editor: Guanfeng Liu

Typhoons are common natural phenomena that often have disastrous aftermaths, particularly in coastal areas. Consequently, typhoon track prediction has always been an important research topic. It chiefly involves predicting the movement of a typhoon according to its history. However, the formation and movement of typhoons is a complex process, which in turn makes accurate prediction more complicated; the potential location of typhoons is related to both historical and future factors. Existing works do not fully consider these factors; thus, there is significant room for improving the accuracy of predictions. To this end, we presented a novel typhoon track prediction framework comprising complex historical features—climatic, geographical, and physical features—as well as a deep-learning network based on multitask learning. We implemented the framework in a distributed system, thereby improving the training efficiency of the network. We verified the efficiency of the proposed framework on real datasets.

1. Introduction

Typhoons are tropical cyclones that occur in the Western Pacific and adjacent waters and are common climate phenomena. Given that typhoons have significant destructive power and often imperil the coastal areas where they make landfall, the nature of these typhoons has long been an important research topic [1–3].

Typhoon track prediction is a typical problem in typhoon research. Traditionally, typhoon paths are often predicted through such methods as force analysis and mathematical statistics [4–7]. In recent years, however, with the development of artificial intelligence, more researchers are using deep-learning technology to predict the movement of typhoons. For example, some studies have utilized cloud maps to locate typhoons and predict their movement via convolutional neural networks (CNNs) and generative adversarial networks (GANs) [8,9]. Given that typhoon track is a continuous process, many studies also use recurrent neural networks (RNNs) and long short-term memory (LSTMs) to process the track sequence [10]. The formation and movement of typhoons is a very complex process that is affected by historical as well as future factors. Although this problem has been widely studied, some limitations remain and hinder the accurate prediction of the paths typhoons take.

Typhoons have complex historical features. Existing studies have evaluated the history of typhoons with respect to geopotential height, wind field, and atmospheric pressure; however, these studies did not comprehensively analyse the features of previous typhoons. Therefore, by analysing historical data, we identified additional pertinent features and categorized them into climatic, geographical, and physical features. Further, we considered some new features—such as geostrophic force—for the purposes of this study. The factors that affect typhoon movement from many aspects were categorized under multimodal features.

Although existing works apply deep learning to evaluate typhoons, most only consider the track of a typhoon as an isolated target and ignore the multiple factors that influence this track. Likewise, although a few studies have predicted typhoon tracks via a multifaceted approach, their analyses of typhoon features are too simplistic. Therefore, we combined

the complex features of typhoons, processed the features through different learning frameworks, and incorporated multitask learning to further improve the accuracy of typhoon track prediction.

However, the expansion of data and model parameters is accompanied by an increase in computational power and duration of model training. In this regard, using distributed and parallel training methods such as SparkMLlib (http://spark.apache.org/mllib/) can significantly improve the efficiency of model training. Therefore, to improve the training efficiency of the framework proposed in this paper, we implemented it based on Ray (https://ray.io/), which is an emerging distributed AI platform.

The contributions of this paper are as follows:

(1) We propose a typhoon track prediction framework that considers both historical features and the interaction of multiple factors.

(2) We extracted the complex features—climatic, geographical, and physical—that affect the movement of typhoons. We employed deep-learning networks and a multitask learning method to improve the accuracy of typhoon track prediction.

(3) We utilized distributed implementation to improve the training efficiency of the network.

(4) We used real-life datasets to conduct the experiments and verify the effectiveness of the proposed framework.

The remainder of this paper is organized as follows: Section 2 introduces related works on typhoon track prediction. Section 3 covers the problem definition and related technologies. Section 4 introduces the proposed track prediction framework, including feature selection and network structure. We then verify the efficiency of the proposed framework through experiments in Section 5 and finally summarize this paper in Section 6.

2. Related Works

2.1. Traditional Methods. Traditional methods of typhoon track prediction include numerical, statistical, regression, and integrated models. Weber [7] proposed a numerical model (STEPS) to analyse the annual performance of the numerical orbit-prediction model; the model involves a very complex atmospheric-dynamics formula and requires strong computational power to successfully predict a typhoon's path. Demaria et al. [4] proposed a statistical model (SHIPS) that modifies the predictor according to the new prediction factors of every new year to make the model more suitable for observing typhoon movement. Compared with STEPS, SHIPS has a lower computational complexity; nonetheless, its accuracy is also relatively low. Goerss and Krishnamurti et al. [5,6] demonstrated that the integrated model comprising multiple models was more accurate as opposed to individual models. Although traditional models play a crucial role in forecasting typhoon tracks, they still have many shortcomings. With the increase in meteorological detection instruments, more meteorological spatiotemporal data (big data) will be produced. However, traditional models are inevitably becoming outmoded. It is difficult for them to capture nonlinear typhoon models from these huge datasets, which significantly reduce the accuracy of prediction.

2.2. Deep-Learning Methods. In recent years, deep learning and parameter optimization [11] have rapidly developed and provided more powerful methods for typhoon track prediction. Neural networks have the advantages of nonlinearity and nonlocality. They can utilize big data to train the network and hence determine the mapping relationships between input and output; this essentially makes the predictions more accurate.

CNN-based methods: Wang et al. [9] used 2250 infrared satellite images to train the CNN network. The average angular error of typhoon track prediction was thus reduced to 27.8 degrees, indicating the great potential of CNN in typhoon path prediction. Giffard-Roisin et al. [12] proposed a fusion neural network comprising a neural network using past trajectory data and a CNN involving the reanalysis of atmospheric wind-field images.

GAN-based methods: Rüttgers et al. [8] used GAN in conjunction with satellite images and meteorological data to forecast the central location of typhoons. It has been proven that GAN utilizes many features that otherwise cannot be used by traditional models, thus preventing the otherwise inevitable errors associated with some traditional models.

RNN- and LSTM-based methods: Moradi Kordmahalleh et al. [13] used sparse RNNs with flexible topology in which a genetic algorithm (GA) was used to optimize the weight connection. Alemany et al. [14] proposed a fully connected RNN in the grid system; the proposed approach can be used to model the complex and nonlinear temporal behavior of typhoons. Further, it can accumulate the historical information of the nonlinear dynamics of the atmospheric system by updating the weight matrix, hence improving the accuracy of typhoon track prediction. Chandra and Dayal and Chandra et al. [15,16] also proved that RNNs are suitable for typhoon track prediction. Lian et al. [17] proposed a novel data-driven deep-learning model composed of a multidimensional feature-selection layer, a convolution layer, and a gating-cycle unit layer. It uses spatial locations and a variety of meteorological features to predict typhoon trajectories. Compared with CNNs and RNNs without a feature-selection layer, the novel model has higher accuracy. Using records from 1949 to 2012 as the training data, Gao et al. [10] proposed a typhoon track prediction method based on LSTM; the research shows that the model can predict the typhoon track 6–24 hours in advance with better accuracy. Kim et al. [18] proposed a large number of temporal and spatial prediction models based on the ConvLSTM model.

Multitask learning-based methods: Chandra [19] proposed a coevolutionary multitask learning algorithm that combines the functions of modularization and multitask learning. This approach coordinates multitask learning, dynamic programming, and coevolution algorithms. Furthermore, it can train neural networks via feature sharing and modular knowledge representation. It can also be used to predict typhoon intensity, with limited input [20]. This shows that, compared with traditional models, the algorithm not only solves the problem of dynamic time series but also improves the prediction accuracy. Mukherjee and Mitra [21] proposed a joint learning model that can learn the distance and direction of typhoons simultaneously via two different structures with multiple LSTMs and multiple fully connected layers; initial layer parameters are shared according to past typhoon track data. The research results show that the model can predict direction and distance (i.e., displacement) simultaneously.

3. Preliminaries

In this section, we first introduce the relevant technologies utilized in our framework and then proceed to define our problem.

3.1. CNN and ResNet. CNN is a type of deep-learning model that has been successfully implemented in image recognition [22]. The convolution layer is one of the core structures of CNNs. The input of the convolution layer includes one or more matrices of the same size, each of which is called a channel. Each convolution layer uses common parameters known as convolution kernels. For 2D input, the function of the convolution layer is to weigh the corresponding submatrices according to the size of kernels; thus, the convolution layer output is generated.

Another important structure is the pooling layer, which aims to reduce the parameters of the model and strengthen the network while improving the computing speed. The strategies of the pooling layer include the maximum and average pooling.

ResNet is a CNN model widely used for feature extraction [23]. To solve the migration problem in deep networks, ResNet proposes residual learning. ResNet replaces the feature $H(x)$ obtained by convolution layers with the residual $H(x) - x$ of feature and input. In contrast with ordinary CNN, ResNet adds a shortcut mechanism between every two layers to realize residual learning.

3.2. LSTM. LSTM is a special type of RNN that is deliberately designed to avoid long-term dependence. It introduces a gate to solve gradient disappearance or explosion [24]. LSTM contains four important structures, namely, the forget gate, the input gate, the update stage, and the output gate. As shown in Figure 1, this framework operates as follows:

(1) The function of the forget gate is implemented by sigmoid to determine which information needs to be forgotten according to the input x_t and the output h_{t-1} of the previous cell.

(2) The input gate determines the information that will be stored in the current cell.

(3) The update stage updates C_t of the current cell.

(4) The output gate outputs the final information to the next cell.

3.3. Multitask Learning. In single-task learning (involved in the previous models), the model learns only one task at a time. For complex problems, single-task learning decomposes the problems into multiple independent subproblems for separate training and then combines them. However, in practical applications, these subproblems frequently contain correlation information that is often ignored by the single-task learning method.

In this regard, the goal of multitask learning is to integrate multiple related tasks through shared representations [25]. It entails hard and soft parameter sharing. Hard parameter sharing shares some parameters among all tasks and only uses the tasks' unique parameters at a specific layer. In soft parameter sharing, each task has unique parameters. Finally, the similarity is expressed by adding constraints to the differences between parameters of different tasks.

3.4. Problem Definition. The problem of typhoon track prediction can be expressed in terms of the features of a given typhoon at several past instances or moments; the goal is to predict the locations at certain times or instances in the future. The past-feature sequence of the typhoon is denoted as $S(s_1, s_2, \ldots, s_t)$, where s_i represents the features of the typhoon at time i and t is the length of the sequence. The track of the typhoon at the future moment is $T[(x_{t+1}, y_{t+1}), (x_{t+2}, y_{t+2}), \ldots, (x_{t+n}, y_{t+n})]$, where (x_j, y_j) is the geographical coordinate (latitude and longitude) at time t_j. The goal of this study is to establish the mapping model $M: T \longrightarrow S$ and hence calculate the future trajectory sequence T through the historical sequence S.

4. Framework

Figure 2 illustrates the structure of our proposed framework. It entails feature selection, weighted fusion, and multitask prediction. In this section, we will introduce all the parts individually.

4.1. Features. There are three types of features in our framework, namely, climatic, geographical, and characteristic features. The climatic features include sea surface temperature, geopotential height, and specific humidity. Geographical features include geostrophic forces. The characteristic features are the speed and position of the typhoon.

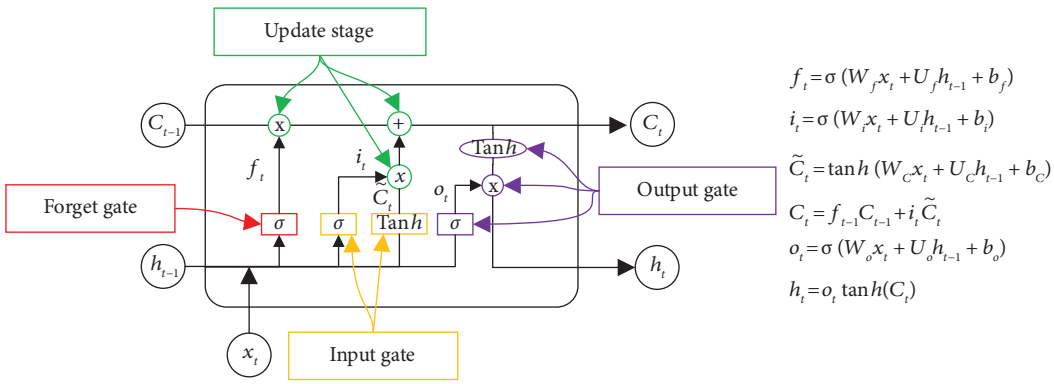


Figure 1: Structure of LSTM.

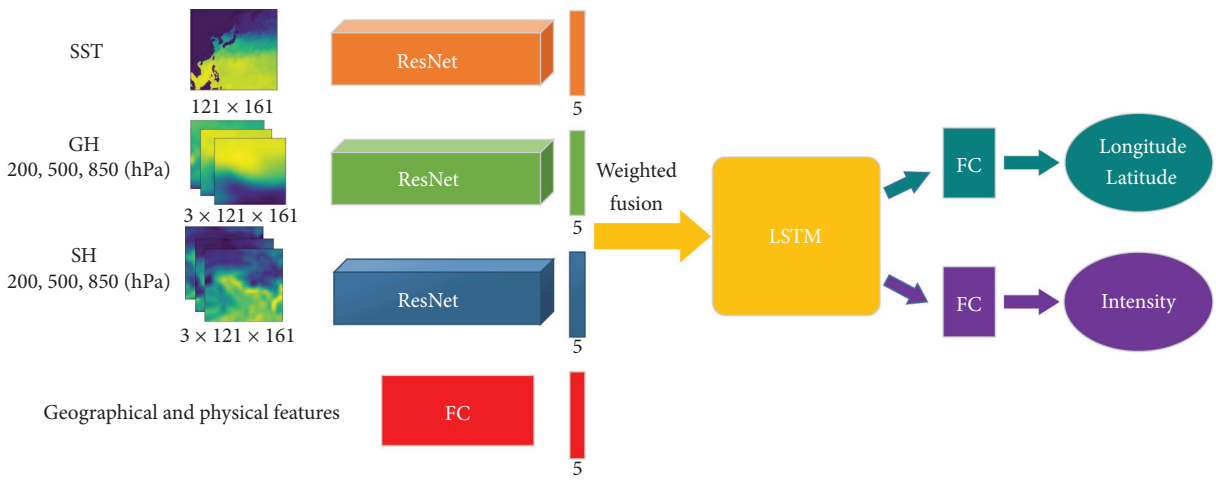


Figure 2: The structure of our framework.

4.1.1. Climatic Features. By studying the influence of climate on typhoons, we selected three main factors as climatic features in this study.

Sea surface temperature (SST): SST is one of the most important factors in meteorological research. In general, SST decreases when latitude increases. SST plays a pivotal role in the formation and movement of typhoons; typhoons are formed above the sea surface where SST is higher than 26.5°C and the intensity of the typhoon increases through continuous absorption of energy. SST is also one of the main factors influencing the direction of motion and landing location of typhoons. In this study, we mainly considered the region within 0°N and 60°N latitudes and 100°E and 180°E longitudes. The SST in this area was regularly collected by the sensor. As shown in Figure 3, we used a matrix of 121 rows and 161 columns to represent SST, in which the SST near the equator is above 30°C, whereas the SST at higher latitudes is approximately 0°C. We also distinguished land from sea; the darkest shades in Figure 3 are land.

Geopotential height (GH): GH is an imaginary height in meteorology, expressed in terms of the work done against gravity by an object of unit mass rising from sea level to a certain height. GH also plays an important role in maintaining the intensity and motion of typhoons. For example, the large geopotential height gradient between the Western Pacific subtropical high and typhoon determines the direction of movement of typhoon Ambi to a certain extent [13]. We studied GH in the same region described previously. In contrast to

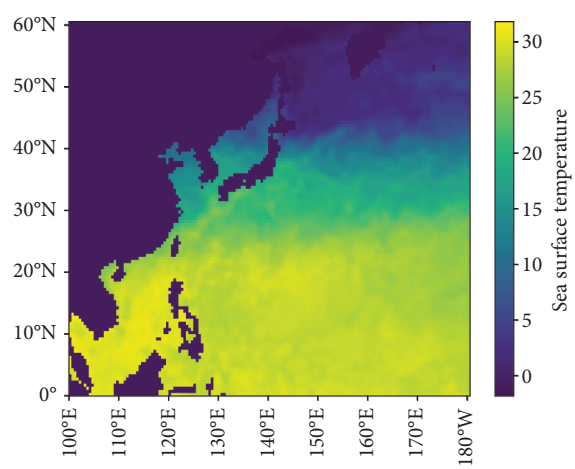


Figure 3: An example of SST.

SST, we choose three different GHs under different hPa. Figures 4(a)–4(c) show examples of GH, which is also represented by matrices with 121 rows and 161 columns. It is evident from these charts that GH increases with latitude.

Specific humidity (SH): SH refers to the ratio of the mass of water vapor in the atmosphere to the total mass of air. There is a strong relationship between typhoons and vertical air motion, and SH is usually used when discussing the vertical motion. Therefore, we introduced SH as a distinct climatic feature. Figures 4(d)–4(f) show 3 SH data charts under different hPa. We can observe that SH in the south is higher than that in the north.

4.1.2. Geographical Feature. (1) Geostrophic force (GF): GF, also known as the Coriolis force, was derived to describe the force exerted on moving objects on the surface of the Earth as a result of the Earth's rotation. Owing to the existence of GF, a rotating flow of air is formed, and eventually, a typhoon is formed under the combined action of various factors. The typhoon is also affected by GF during its movement. In the northern hemisphere, the GF of the typhoon is to the right, which determines the typhoon's direction of movement to a certain extent. GF can be expressed as

$$F = 2mv\omega \sin \theta, \tag{1}$$

where m is the mass of the object, v is the velocity of the object, ω is the angular velocity of the Earth's rotation, and θ is the latitude of the object before it begins to move. Given that the mass of typhoons is difficult to estimate, we use the geostrophic force gradient to represent the influence of GF on typhoons, denoted as

$$\frac{\partial F}{\partial m} = 2v\omega \sin \theta. \tag{2}$$

4.1.3. Physical Features. We use physical characteristics to describe the time series and tracks of typhoons.

Location and direction: Given that the track of a typhoon is a series of coordinates, we used the latitudes and longitudes (lat, lon) or offsets $(\Delta\text{lat}, \Delta\text{lon})$ to describe the location and direction of motion of typhoons. Since typhoon data were collected every 6 hours, we calculated the movement and direction of the typhoon every 6 hours.

Speed: The typhoon data are coarse. Therefore, we used the average of the velocities of the typhoon at two consecutive moments to describe the moving velocity of the typhoon.

Intensity: The intensity of a typhoon is determined by its wind speed. Existing studies have validated the relationship between the central pressure of a typhoon and the maximum wind speed [26]. Therefore, we used the maximum central pressure to express the intensity characteristics of a typhoon.

4.2. Network. Owing to the different modes of features, we used different networks to process the features and then used feature fusion for learning. The entire network architecture is illustrated in Figure 2.

4.2.1. Feature Extraction. We used climatic, geographical, and physical features. Some of these were two-dimensional matrices, whereas some were one-dimensional vectors. Consequently, we used different networks for different features.

For climatic features, all inputs were two-dimensional images. We therefore used three ResNets to process the images. The ResNets employed in our framework have 18 hidden layers [23] as shown in Figure 5. GH and SH have trichannel inputs, whereas SST has single-channel input. The first layer is a convolution layer. The size of the convolution kernel is 7×7 and the stride is $(2, 2)$. Based on the size of the input, we set padding as $(3, 3)$. Batch normalization (BN) and rectified linear units (ReLU) were also used in the convolution layer. After the convolution operation, the network performs a maximum-pooling operation. There are four residual blocks after the first layer. Each residual block is repeated twice. To simplify the representation, the repeated parts have been replaced by ellipses. Each residual block contains two convolution layers. Each layer contains a convolution kernel, batch normalization, and ReLU. The size of the convolution kernel is 3×3, the stride is $(1, 1)$, and the padding is $(1, 1)$. The output dimensions of each residual block are 64, 128, 256, and 512. After the last residual block, the network performs an average-pooling operation. The last layer of the network is a fully connected network with 5-dimensional output.

As for the geographical and physical features, we used a fully connected network and obtained a 5-dimensional vector as the output. For feature fusion, we adopted a weight module. The weight of each feature can be regarded as the correlation between the feature and track of the typhoon. Through weighted feature fusion, for each moment, we obtained a 20-dimensional feature vector, which then became the input of the predictor.

4.2.2. Multitask Prediction. Because LSTM has a considerable advantage in the processing of sequence data, we used the classic LSTM as the predictor. The dimension of the input was $t \times 20$, where t is the length of the sequence, as introduced in Section 2. The training process is shown in Figure 6. First, we used zero-state initialization to calibrate the weight, h_0, and C_0. For each cell of the LSTM, the input is the i-th 20-dimensional feature vector. It should be noted that all LSTM cells share these parameters.

The LSTM output is divided into two tasks. The main task involves locating the typhoon at the next moment, and the auxiliary task involves determining the central pressure of the typhoon (i.e., the intensity of the typhoon). We used the L_2 norm as the loss function of the two tasks.

For the main task, the loss is the difference in distance between the real location and the predicted location of the typhoon, as follows:

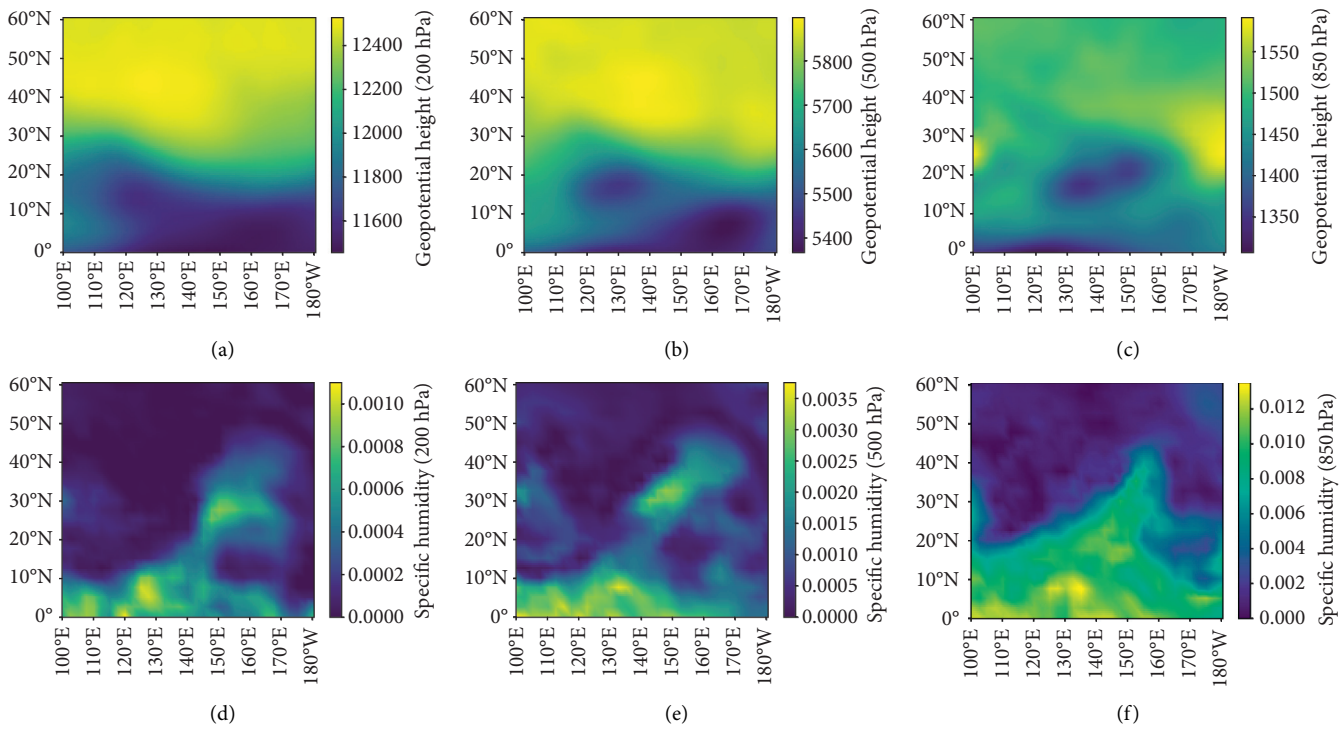


Figure 4: Example of GH and SH. (a) GH at 200 hPa. (b) GH at 500 hPa (c) GH at 850 hPa. (d) SH at 200 hPa. (e) SH at 500 hPa. (f) SH at 850 hPa.

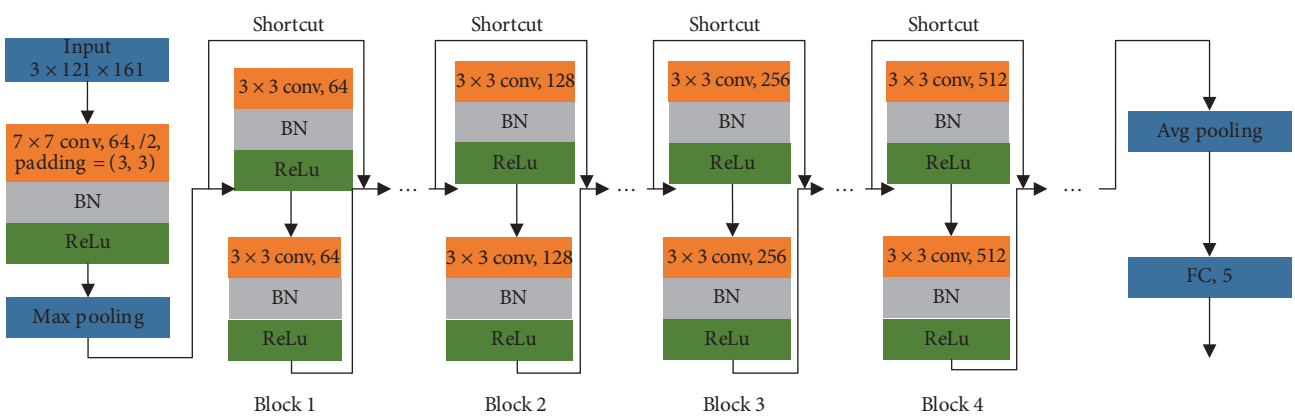


Figure 5: Details of ResNet in our framework.

$$L_{\text{main}} = \sqrt{(x - \hat{x})^2 + (y - \hat{y})^2}, \tag{3}$$

where (x, y) is the location of the typhoon at the next moment and $(\hat{x}, \hat{y})$ is the output of the predictor. The longitude and latitude offset can also be used as input, and the corresponding loss will become the difference in the offset. For the auxiliary task, the loss function is denoted as

$$L_{\text{auxiliary}} = \sqrt{(p - \hat{p})^2}, \tag{4}$$

where p is the central pressure of the typhoon and $\hat{p}$ is the prediction result.

Therefore, the total loss of our framework is as follows:

$$L_{\text{total}} = \alpha L_{\text{main}} + (1 - \alpha) L_{\text{auxiliary}}. \tag{5}$$

In this loss function, α is a hyperparameter.

4.3. Distributed Implementation. To ensure that the proposed framework can handle big data and consequently improve the efficiency of training, we implemented a distributed framework based on Ray. Ray is a very popular distributed AI platform implemented via Python. This facilitates the rapidly distributed computing of the Python

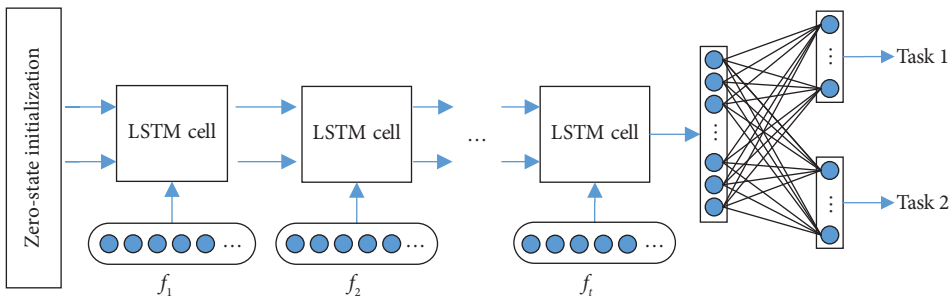


Figure 6: The details of multitask prediction.

code. In the implementation, each network structure (such as convolution layer, pooling layer, and FC layer) is implemented as a class, also known as an actor in Ray. Multiple actors construct the entire network through the data flow. In the calculation process, each calculation node starts multiple workers as the basis of calculation. Each actor is assigned to the corresponding worker for execution. In the training process, the data flows through gRPC and shared memory to the corresponding worker for calculation. For example, in each ResNet, after the calculation of the current layer is completed, the data will flow to the worker of the next layer. There is no data dependence between multiple ResNets; therefore, parallel training can be realized.

5. Experiments

5.1. Setup. We use a real dataset to verify the effectiveness of our framework. The dataset is the Western Pacific Typhoon track data from the JTWC (be Typhoon Warning Center, the Joint Typhoon Warning Center). The dataset contains typhoon tracks from January 1, 2001, to December 31, 2005. The attitude is from 0°N to 60°N and the longitude is from 100°E to 180°E. Statistics of the experimental setup are shown in Table 1.

We use the metric of distance error (same as L_{main}) to verify the effectiveness of our framework. We first verify the benefits of multitask learning technology to this framework. Next, we use different weights to discuss the relationship between features and results. The framework is implemented by Python 3, and the experiments are conducted on a cluster in which each node has Intel Purley 4110 CPUs and Tesla P100 GPUs.

5.2. Results. In this section, we will introduce the experimental results in the real-life dataset. We report and analyse the results by changing the parameters. Then, we choose some real typhoon tracks to show our prediction results.

Distance error with respect to multitask and single-task learning: firstly, we compare the results of multitask learning (MTL) and single-task learning (STL), as shown in Figure 7. We can obverse that MTL can get better results than STL in most cases. In the 6 h prediction results, MTL is similar to STL. However, in other cases, MTL can achieve about 20% performance improvement. It proves that it is feasible to improve the effect of track prediction by auxiliary tasks. What is more, the best results in 6 h, 24 h, 48 h, and 72 h are about 40 km, 70 km, 220 km and 380 km which are better than most existing models. It also proves the effectiveness of our framework.

Distance error with respect to $|\mathbf{T}||\mathbf{T}|$: secondly, we report the distance error with different size of input $|T|$. The results are also shown in Figure 7. We find that $|T|$ has a great influence on our framework in different cases. The optimal value is 3, 7, 4, and 5 in 6 h, 24 h, 48 h, and 72 h. As $|T|$ becomes larger or smaller, the distance error gradually increases. In the later experiments, we selected the best value of $|T|$ in each case to verify the effect of feature weight on the distance error.

Then, we study the relationship between features and prediction results.

Distance error with respect to $\mathbf{w_{SST}w_{SST}}$: to study the effect of SST, we keep w_{GH} and w_{SH} unchanged and then adjust the value of w_{SST} from 0.1 to 1.0. The results are shown in Figure 8. We can obverse that SST will greatly affect the results. The best choice is to reduce w_{SST} as small as possible.

Distance error with respect to $\mathbf{w_{GH}w_{GH}}$: to study the relationship between GH and prediction results, we keep w_{SST} and w_{SH} unchanged and then adjust the value of w_{GH} from 0.1 to 1.0. As shown in Figure 9, we can get the best results when w_{GH} is set as 0.8. The difference between the best result and the worst result in 6 h, 24 h, and 48 h is about 30 km to 100 km. In 72 h, the difference could be more than 300 km. An appropriate w_{GH} can improve the results by 30% to 40%. The experimental results show that there is a strong correlation between GH and prediction results.

Distance error with respect to $\mathbf{w_{SH}w_{SH}}$: to study the relationship between SH and prediction results, we keep w_{SST} and w_{GH} unchanged and then adjust the value of w_{SH} from 0.1 to 1.0. The results are shown in Figure 10. To get better results, w_{SH} is smaller than w_{GH}. In 6 h and 24 h cases, we can get the best results when w_{SH} is set as 0.1. In 48 h and 72 h cases, it is better to set w_{SH} as 0.3. An appropriate w_{SH} can improve the result by 40% to 50%. The experimental results show that SH is also related to the prediction results, but the correlation is less than GH.

Table 1: Statistics of the experimental setup.

Region		Date range	Dimension of features		
Attitude	Longitude		SST	GH/SH	Others
0°N to 60°N	100°E to 180°E	January 1, 2001, to December 31, 2005	121 × 161	3 × 121 × 161	20 × 1

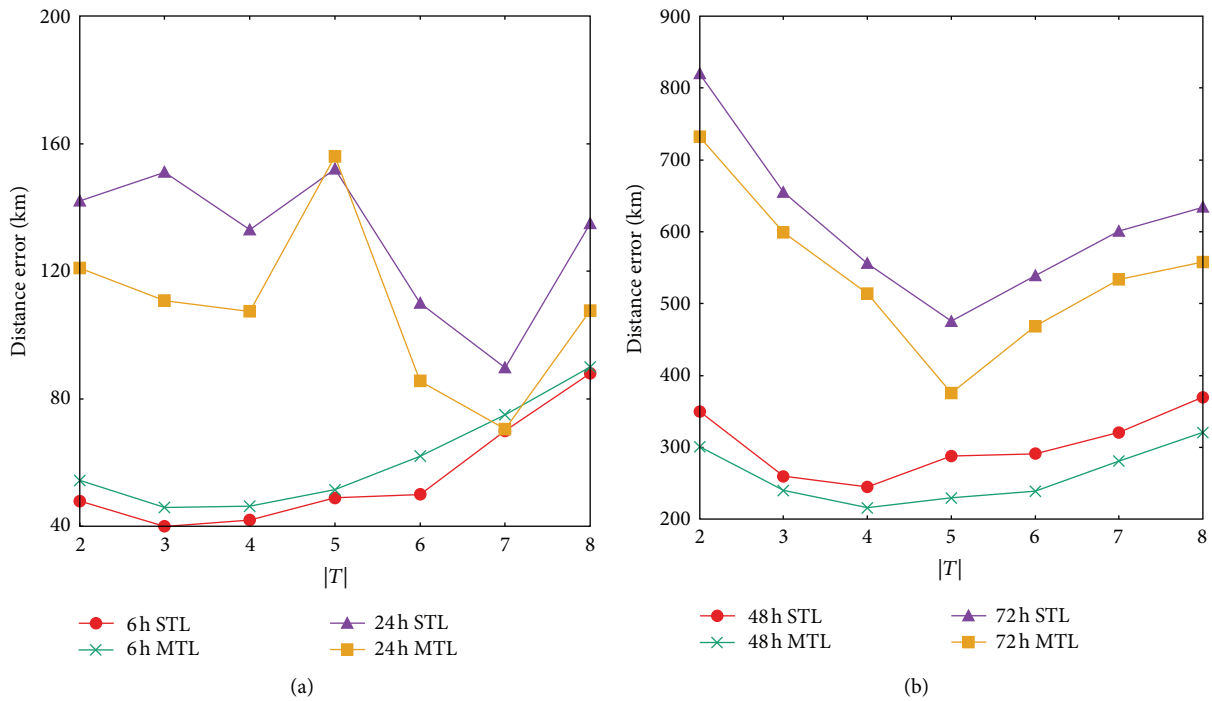


Figure 7: Results of varying $|T|$. (a) Results of 6 h and 24 h. (b) Results of 48 and 72 h.

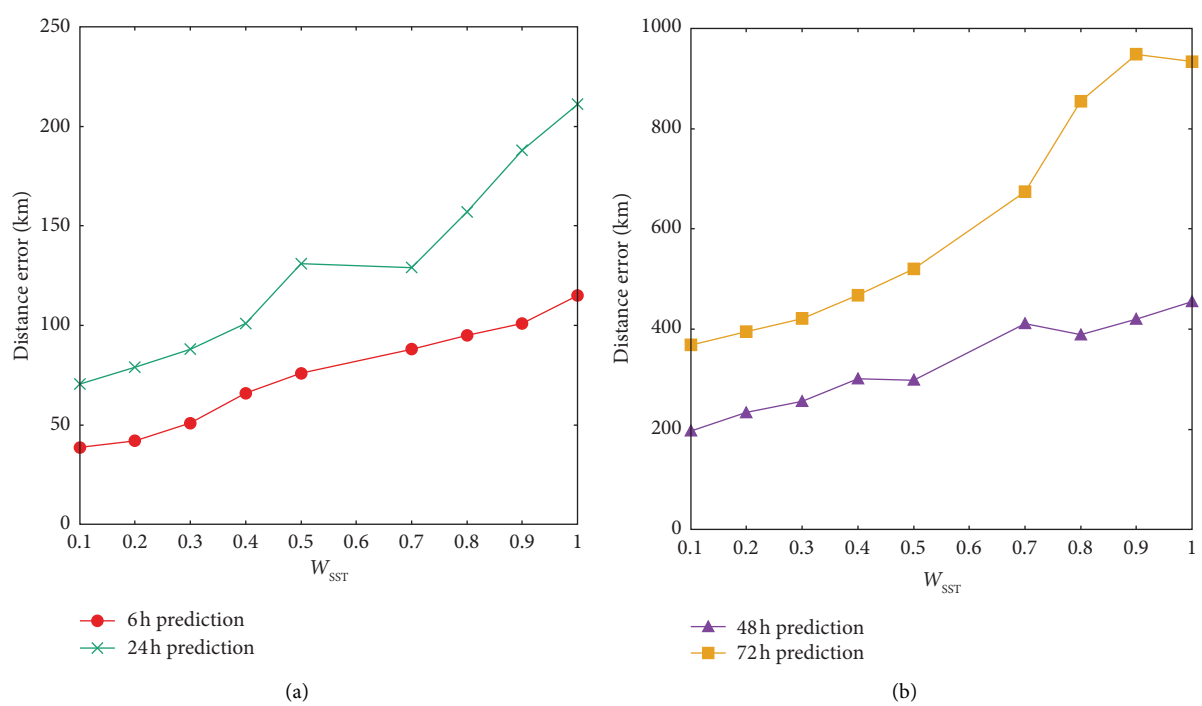


Figure 8: Results of varying weight of w_{SST}. (a) Results of 6 h and 24 h. (b) Results of 48 and 72 h.

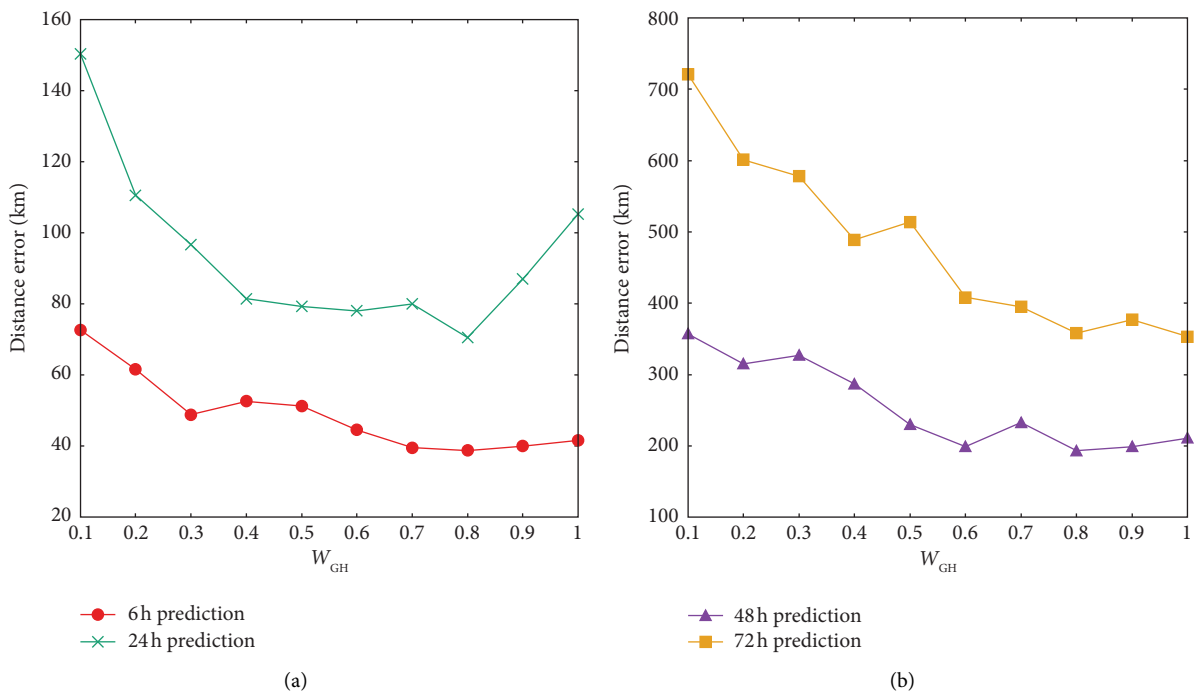


FIGURE 9: Results of varying weight of w_{GH}. (a) Results of 6 h and 24 h. (b) Results of 48 and 72 h.

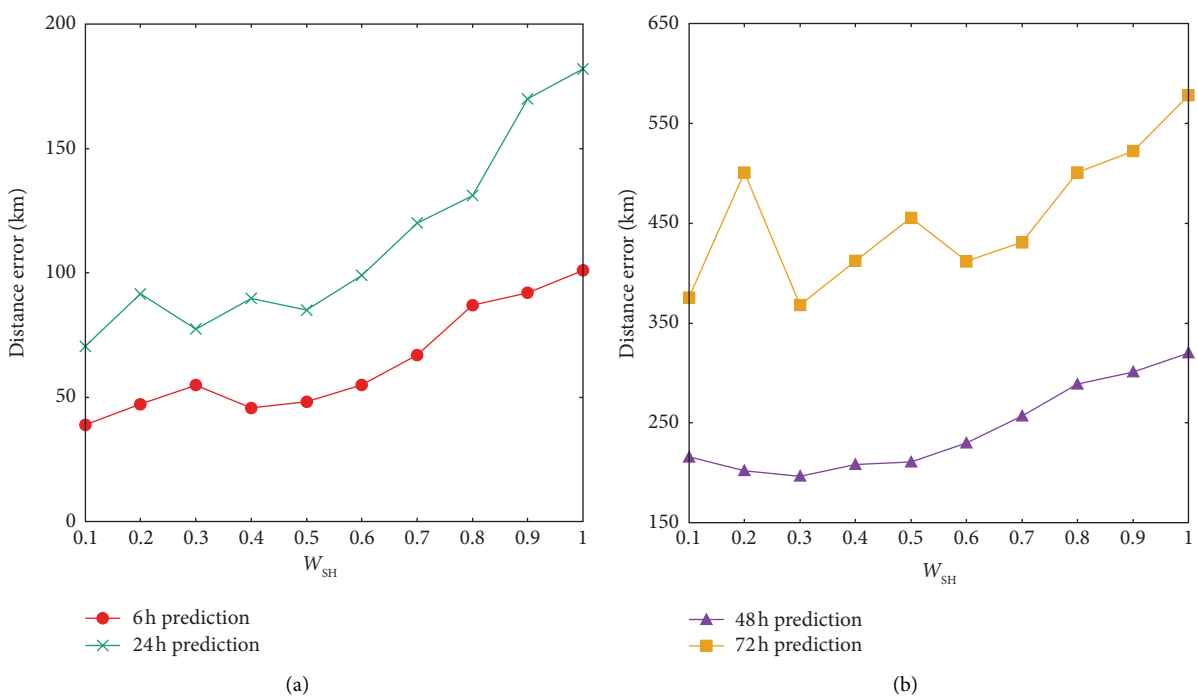


FIGURE 10: Results of varying weight of w_{SH}. (a) Results of 6 h and 24 h. (b) Results of 48 and 72 h.

Case study: We use some real typhoons to compare the real tracks and the prediction results. We select Saola, Damrey, and Longwang that are formed in 2005; the real tracks and 6 h prediction results are shown in Figures 11–13. Typhoon Saola was formed on September 20th; the average distance error of 6 h prediction results is 40.33 km. Typhoon Damrey was formed on September 21, the average distance error is 40.59 km, the minimum error is 8.9 km, and the maximum error is 60.33 km. Typhoon Longwang was formed on September 26; the average distance error is 46.51 km.

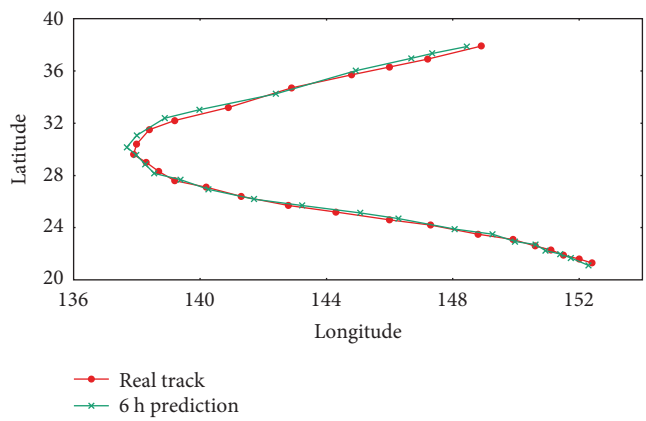


Figure 11: 6 h prediction results of Saola.

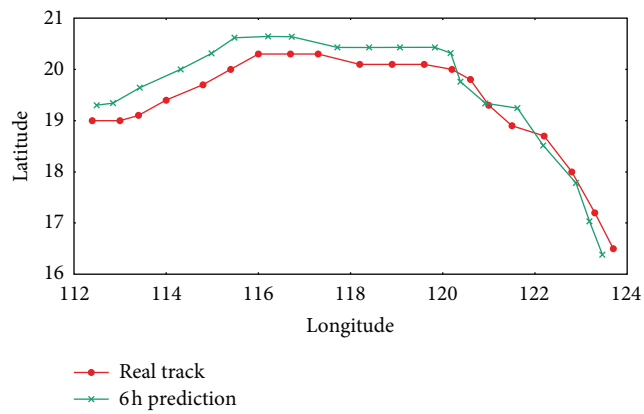


Figure 12: 6 h prediction results of Damrey.

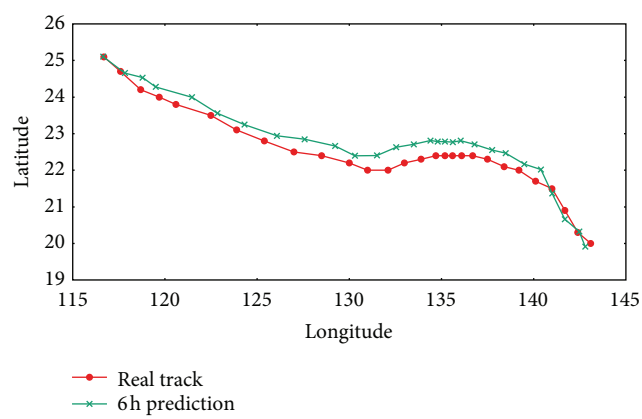


Figure 13: 6 h prediction results of Longwang.

Comparison with existing works: Finally, we compare our framework with several existing works [8,10,12,27]. According to the previous introduction, Rüttgers et al. [8] introduced a GAN-based model, used satellite images as the input, and predicted locations after 6 hours. Gao et al. [10] introduced an LSTM-based model. The work by Giffard-Roisin et al. [12] was based on CNN and feature fusion. Lv et al. [27] used the least square method and FC network to predict the locations. We still use distance error to verify the effectiveness and the results are shown in Table 2. Compared with these works, our framework can achieve high prediction results, especially in 48 h and 72 h cases. In 72 h results, our framework improves the accuracy by 60%.

Table 2: Results compared with the existing works.

	6 h	24 h	48 h	72 h
Our framework	**38.75**	**69.54**	**196.61**	**368.1**
Gao et al. [10]	45.95	105.68	332.54	974.50
Giffard-Roisin et al. [12]	—	136.1	—	—
Rüttgers et al. [8]	95.6	—	—	—
Lv et al. [27]	—	158.34	361.76	—

5.3. Summary. In this section, we verify the effect of different parameters on the performance of our framework in the real dataset. In general, our framework can achieve good results based on multitask and feature weighting. We find that GH has a strong correlation with the movement of typhoons, followed by SH, and SST has the weakest correlation. Through the training results, the optimal prediction results can be obtained by selecting the appropriate parameters for different scenes.

6. Conclusion

In this paper, we proposed a typhoon track prediction framework based on multitask learning and feature weighting. We analysed the correlation between the climatic, geographical, and physical features and typhoon movement through the method of feature weighting. We designed a network based on ResNet and LSTM and used a multitask learning method to improve the prediction accuracy. We implemented the network in a distributed platform. Finally, we conducted experiments on real datasets to prove the effectiveness of the framework. In future works, we will analyse more features and use the attention mechanism to automatically process the weight of features.

Data Availability

The data are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Acknowledgments

The work was supported by the National Key R&D Program of China (Grant no. 2016YFC1401902), the National Natural Science Foundation of China (Grant no. 61972077), and the LiaoNing Revitalization Talents Program (Grant no. XLYC2007079).

References

[1] W. Liu, K. Fujii, Y. Maruyama, and F. Yamazaki, "Inundation assessment of the 2019 typhoon hagibis in Japan using multi-temporal sentinel-1 intensity images," *Remote Sensing*, vol. 13, no. 4, p. 639, 2021.

[2] J. Cai, Y. Zhang, R. J. Doviak, Y. Shrestha, and P. W. Chan, "Diagnosis and classification of typhoon-associated low-altitude turbulence using HKO-TDWR radar observations and machine learning," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 6, pp. 3633–3648, 2019.

[3] J. Li, Q. Zheng, M. Li, Q. Li, and L. Xie, "Spatiotemporal distributions of ocean color elements in response to tropical cyclone: a case study of typhoon mangkhut (2018) past over the northern south China sea," *Remote Sensing*, vol. 13, no. 4, p. 687, 2021.

[4] M. Demaria, M. Mainelli, L. K. Shay, J. A. Knaff, and J. Kaplan, "Further improvements to the statistical hurricane intensity prediction scheme (SHIPS)," *Weather and Forecasting*, vol. 20, no. 4, pp. 531–543, 2005.

[5] J. S. Goerss, "Tropical cyclone track forecasts using an ensemble of dynamical models," *Monthly Weather Review*, vol. 128, no. 4, pp. 1187–1193, 2000.

[6] T. N. Krishnamurti, C. M. Kishtawal, Z. Zhang et al., "Multimodel ensemble forecasts for weather and seasonal climate," *Journal of Climate*, vol. 13, no. 23, pp. 4196–4216, 2000.

[7] H. C. Weber, "Hurricane track prediction using a statistical ensemble of numerical models," *Monthly Weather Review*, vol. 131, no. 5, pp. 749–770, 2003.

[8] M. Rüttgers, S. Lee, S. Jeon, and D You, "Prediction of a typhoon track using a generative adversarial network and satellite images," *Scientific Reports*, vol. 9, no. 1, pp. 6057–6115, 2019.

[9] C. Wang, Q. Xu, X. Li et al., "CNN-based tropical cyclone track forecasting from satellite infrared images," in *Proceedings of the IEEE International Geoscience and Remote Sensing Symposium*, pp. 5811–5814, Waikoloa, HI, USA, September 2020.

[10] S. Gao, P. Zhao, B. Pan et al., "A nowcasting model for the prediction of typhoon tracks based on a long short term memory neural network," *Acta Oceanologica Sinica*, vol. 37, no. 5, pp. 8–12, 2018.

[11] J. Chen, M. Zhong, J. Li, D. Wang, T. Qian, and H. Tu, "Effective deep attributed network representation learning with topology adapted smoothing," *IEEE Transactions on Cybernetics*, 2021.

[12] S. Giffard-Roisin, M. Yang, G. Charpiat, C. Kumler Bonfanti, B. Kégl, and C. Monteleoni, "Tropical cyclone track forecasting using fused deep learning from aligned reanalysis data," *Frontiers in Big Data*, vol. 3, p. 1, 2020.

[13] M. Moradi Kordmahalleh, M. Gorji Sefidmazgi, and A. Homaifar, "A sparse recurrent neural network for trajectory prediction of atlantic hurricanes," in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 957–964, Lille, France, July 2016.

[14] S. Alemany, J. Beltran, A. Perez et al., "Predicting hurricane trajectories using a recurrent neural network," in *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, pp. 468–475, Honolulu, HI, USA, January 2019.

[15] R. Chandra and K. Dayal, "Cooperative neuro-evolution of Elman recurrent networks for tropical cyclone wind-intensity prediction in the south pacific region," in *Proceedings of the*

IEEE Congress on Evolutionary Computation (CEC), pp. 1784–1791, Sendai, Japan, May 2015.

[16] R. Chandra, K. Dayal, and N. Rollings, "Application of cooperative neuro-evolution of Elman recurrent networks for a two-dimensional cyclone track prediction for the South Pacific region," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, Killarney, Ireland, July 2015.

[17] J. Lian, P. Dong, Y. Zhang, J. Pan, and K. Liu, "A novel data-driven tropical cyclone track prediction model based on CNN and GRU with multi-dimensional feature selection," *IEEE Access*, vol. 8, pp. 97114–97128, 2020.

[18] S. Kim, H. Kim, J. Lee et al., "Deep-hurricane-tracker: tracking and forecasting extreme climate events," in *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, pp. 1761–1769, Waikoloa, HI, USA, January 2019.

[19] R. Chandra, "Dynamic cyclone wind-intensity prediction using co-evolutionary multi-task learning," in *Proceedings of the International Conference on Neural Information Processing*, pp. 618–627, Guangzhou, China, November 2017.

[20] R. Chandra, Y.-S. Ong, and C.-K. Goh, "Co-evolutionary multi-task learning for dynamic time series prediction," *Applied Soft Computing*, vol. 70, pp. 576–589, 2018.

[21] A. Mukherjee and P. Mitra, "Joint learning for cyclone track nowcasting," in *Proceedings of the ECML/PKDD, CEUR Workshop*, Ghent, Belgium, September 2020.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, pp. 1097–1105, 2012.

[23] K. He, X. Zhang, S. Ren et al., "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, USA, June 2016.

[24] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[25] K.-H. Thung and C.-Y. Wee, "A brief review on multi-task learning," *Multimedia Tools and Applications*, vol. 77, no. 22, pp. 29705–29725, 2018.

[26] K. Chen, "Calculation of the maximum wind speed of typhoon in the western pacific," *Marine Science Bulletin*, 1985.

[27] Q. P. Lv, J. Luo, K. Zhu et al., "Experiments on predicting tracks of tropical cyclones based on artificial neural network," *Guangdong Meteorology*, pp. 19–22, 2009.

Hindawi
Complexity
Volume 2021, Article ID 5580631, 16 pages
https://doi.org/10.1155/2021/5580631

Research Article

Analysis and Design of the Battery Initial Energy Level with Task Scheduling for Energy-Harvesting Embedded Systems

Xingyu Miao, Jiayuan Wei, and Yongqi Ge

School of Information Engineering, Ningxia University, Yinchuan 750021, China

Correspondence should be addressed to Yongqi Ge; geyongqi@nxu.edu.cn

Received 5 February 2021; Revised 18 March 2021; Accepted 30 March 2021; Published 16 April 2021

Academic Editor: Yongsheng Hao

When the energy-harvesting embedded system (EHES) is running, its available energy (harvesting energy and battery storage energy) seems to be sufficient overall. However, in the process of EHES task execution, an energy shortage may occur in the busy period such that system tasks cannot be scheduled. We call this issue the energy deception (ED) of the EHES. Aiming to address the ED issue, we design an appropriate initial energy level of the battery. In this paper, we propose three algorithms to judge the feasibility of the task set and calculate the appropriate initial energy level of the battery. The holistic energy evaluation (HEE) algorithm makes a preliminary judgment of the task set feasibility according to available energy and consumption energy. A worst-case response time-based initial energy level of the battery (WCRT-IELB) algorithm and an accurate cycle-initial energy level of the battery (AC-IELB) algorithm can calculate the proper initial battery capacity. We use the YARTISS tool to simulate the above three algorithms. We conducted 250 experiments on As Late As Possible (ALAP) and As Soon As Possible (ASAP) scheduling with the maximum battery capacities of 50, 100, 200, 300, and 400. The experimental results show that setting a reasonable initial energy level of the battery can effectively improve the feasibility of the task set. Among the 250 task sets, the HEE algorithm filtered 2.8% of them as infeasible task sets. When the battery capacity is set to 400, the WCRT-BIEL algorithm increases the success rates of the ALAP and ASAP by 17.2% and 26.8%, respectively. The AC-BIEL algorithm increases the success rates of the ALAP and ASAP by 18% and 26.8%, respectively.

1. Introduction

In recent years, embedded systems are fast becoming a key proportion of computer science and technology in different domains, such as driverless vehicles, medical implants, weather monitoring sensors, wearable devices, and so on [1–6]. Most embedded devices are battery-powered. The battery life determines the embedded system running time. However, the battery capacity is limited. Some embedded systems that are deployed in distant areas require long-term operation [7, 8]. In this case, these systems require periodic battery replacement to maintain running time. Battery replacement, however, is very difficult in general. Energy harvesting provides new insights into this issue. This technology harvests ambient energy and converts it into electrical energy for direct use in an embedded system or stores it in a storage module (i.e., battery) for future uses. The benefit of this approach is that it can increase system running time and eliminate the demand of battery replacement [9]. We refer to an embedded system that uses energy-harvesting technology as an energy-harvesting embedded system (EHES). The major problem in EHES is to ensure that the task calculation of the embedded system can obtain enough energy. The EHES generally consists of three parts, as shown in Figure 1, in which the energy source (such as sunlight, wind power, and vibration energy) is converted into electrical energy by the energy harvester and passed to the energy storage device (battery) for storage. Because of the unpredictability of the energy source (such as a period of overcast rain or the unavailability of solar power during the night), the task of harvesting energy is uncertain. Thus, the converted electrical energy cannot power the system stably.

In this work, we consider the issue of ED arising from scheduling algorithms in EHES. The main reason for ED is

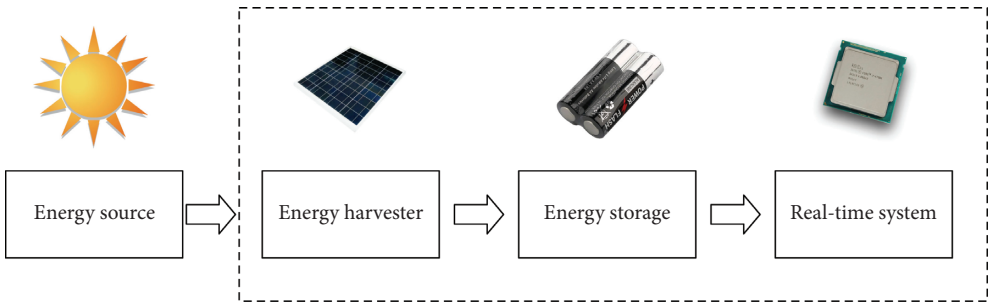


Figure 1: Energy-harvesting embedded system architecture.

that when the system carries out a hyperperiod task, the energy consumed is less than the energy stored in the battery plus the harvested energy, and the system will not stop because of the lack of energy; however, in this hyperperiod, the task may be executed very frequently during a certain period of time; at this time, the energy consumption is relatively large, and the harvested energy and the energy stored in the battery are insufficient to support this consumption, causing the system to stop running. This is very contradictory to the above situation. Through our research, we found that setting the proper initial energy level of the battery can effectively eliminate the ED issue. Setting the battery's initial energy can effectively eliminate the lack of energy in the process of task execution. We proposed an energy level judgment algorithm and two battery initial level calculation algorithms to solve this issue.

The contributions of this paper are as follows:

(1) In this work, we conduct the system schedulability analysis and find the necessary condition for the viability of the EHES, designing a task scheduling pre-judgment method based on task set attributes and energy production power for EHES, which can filter out those task sets that are not globally feasible. Among the 250 task sets, we filtered 2.8% of them as infeasible task sets.

(2) An analysis of the initial battery energy level issue of the battery is given in this work. The battery must have initial energy for some tasks to be schedulable. Based on the worst-case response time (WCRT) model, the WCRT-IELB algorithm is proposed, and the schedulability of the scheduling algorithm is effectively improved by setting the initial energy level. Simulations show that the introduction of the WCRT-IELB algorithm for ALAP and ASAP is better than that without the introduction of the WCRT-IELB algorithm, and when the battery capacity is set to 400, the success rate is increased by 17.2% and 26.8%, respectively.

(3) An online AC-IELB algorithm is proposed, which is more accurate than the initial energy calculation method based on WCRT. Experiments show that the success rate of the AC-IELB algorithm based on ALAP and ASAP is increased by 28%, and 26.8%, respectively, compared with the original algorithm when the battery capacity is set to 400.

(4) The experimental results show that the AC-IELB algorithm has higher successful rate than the WCRT-IELB algorithm. When the battery capacity is set to 400, the success rates of ALAP are increased by 0.8%.

The remainder of this paper is organized as follows. The related works are summarized in Section 2. In Section 3, we review the general model. Section 4 explains the research motivation, and we conduct system schedulability analysis in Section 5. Section 6 presents three algorithms and describes their rules in detail. Simulation results and discussions follow in Section 7. Finally, we conclude this work and provide some directions for future works in Section 8.

2. Related Work

In the past two decades, researchers began to address issues of minimizing energy consumption in scheduling. For EHES, many studies have focused on reducing energy consumption or optimizing battery storage efficiency under the premise of ideal battery capacity while ignoring the influence of battery capacity on the scheduling algorithm. In general, researchers focus on the following three points. There are many power management technologies; for instance, dynamic power management (DPM) [10–12] and dynamic voltage and frequency scaling (DVFS) [13–15] techniques are currently two well-known technologies. DPM selectively shuts down idle components in the process of system operation to achieve the purpose of energy savings. DVFS saves energy by reducing the CPU frequency and extending task execution times. When there is not enough energy to execute the task, these two technologies cannot be used, and they are difficult to use in energy-harvesting systems. The approach proposed by Balsamo et al. [16] has successfully solved this problem.

The strategy of battery energy storage and consumption is designed to extend battery life and achieve the purpose of extending system life. Most of the works use ideal battery and/or supercapacitor models; the current works consider more accurate battery and/or supercapacitor models. At the same time, it causes some very complex issues about prediction of harvestable energy and the battery and/or supercapacitor status. According to the current state of the system, the Highest/Lowest-Power-First (HLPF) real-time task scheduling algorithm proposed by Hasanlooa et al. will store electrical energy in the system as much as possible to avoid waste, thereby increasing the life of the system. And

they proposed hybrid energy storage system (HESS) component scheduling [17] which has a similar function. Furthermore, in [18], Kwak et al. researched the impact of task scheduling on battery aging, and the main principle of minimizing battery aging was proposed based on results.

The feasible scheduling algorithm is designed to extend the system running time. Allavena and Mossé [19] first focused on embedded systems with battery charging and deadline constraints. They proposed a simple and effective task scheduling method in a frame-based system of maximum and minimum energy constraints. However, this method needs to be carried out under a very strict task model in which all tasks have the same period and implicit period. Then, Moser et al. [20] proposed an algorithm called the Lazy Scheduling Algorithm (LSA), which relies on the energy consumption of the task to change the CPU frequency, thereby adjusting the WCRT. However, the result of this work heavily relies on assumptions and energy consumption directly related to WCRT, which is unrealistic for embedded systems [21]. Abdeddaïm et al. adopted the energy harvesting to address the energy and time constraint in the operation of embedded systems. They proposed two classic scheduling strategies, ASAP [22] and ALAP [23]. The ALAP delays the execution time of the task as much as possible and compresses the slack time as much as possible, enabling the systems to supplement the battery energy to the greatest extent. The ASAP algorithm judges whether the current energy level is sufficient to execute a time unit and if it can be executed, it will execute immediately. Otherwise, the systems will suspend a time unit to replenish energy and then judge. Abdeddaïm et al. [22] proved that the ASAP algorithm is optimal on a nonconcrete task set (a nonconcrete task set is a set of real-time tasks whose offset is only known at runtime). However, the ASAP algorithm will cause frequent switching of battery charge and discharge modes in order to perform tasks as much as possible, which will reduce battery life and thereby reduce system life. This is unrealistic. Afterward, Abdeddaïm et al. [24] extended the ASAP algorithm by incorporating the idea of clairvoyance, proposing an algorithm called FPCASAP. The purpose is to find the optimal algorithm for the concrete task set. However, so far, the algorithm has not been proven to be optimal for the concrete task set. Moreover, the LSA considers the battery capacity as an ideal situation when designing the battery model and sets an initial battery capacity that is always equal to the maximum battery capacity, which is unrealistic [25]. Abdeddaïm et al. [22, 26] evaluated two upper limits of the battery capacity of the fixed-priority scheduling algorithm by two tests. In general, it is difficult to calculate the minimum battery capacity performed by the system due to the consideration of environmental factors and the scheduling algorithm. In the first test, they considered that the ASAP algorithm can accurately obtain this value, since the energy replenished each time during the operation of the ASAP algorithm only needs to be equal to the energy consumed by the single time unit. In this case, the minimum battery capacity that is feasible to maintain the task set is the maximum energy consumption, that is, the maximum instantaneous energy consumption.

It is only necessary to ensure that the maximum battery capacity is not less than maximum instantaneous energy consumption to ensure the task set feasibility. For the second test, they consider that the maximum capacity of the battery is at least equal to the energy consumed by all tasks in the longest busy period of the priority n task. Such maximum battery capacity is equivalent to having unlimited battery capacity in terms of task execution; however, whether considering ASAP, ALAP, or FPCASAP, their battery capacity is unlimited, which is not realistic. Designing the proper battery capacity and initial battery capacity will increase the schedulability of the scheduling algorithm. Ghadaksaz et al. [27] first proposed the calculation method for the battery capacity of the EDF-ASAP algorithm, and simulation results verify that the proper battery capacity is an important issue affecting system task scheduling. To the best of our knowledge, there is no previous work in this era that gives computation methods for battery initial level. In this work, we propose two methods to compute battery initial levels (WCRT-IELB and AC-IELB).

3. Model

The EHES generally consists of two parts: the energy system and the real-time system. Correspondingly, we assume that our model also has two parts: the energy model and the task model. In this work, we consider every time interval as one time unit, while the energy unit depends on the real situation such as the type of energy, the rate of energy conversion, and the rate of energy harvesting.

3.1. Energy Model. In this work, the energy model of EHES consists of the energy production model and energy storage model. The available energy of EHES consists of harvesting energy and battery storage energy.

3.1.1. Energy Production Model. We suppose that ambient energy can be collected by a harvesting model to produce energy and convert it into electrical power with an instantaneous charging rate, denoted as $R_p(t)$. $R_p(t)$ is a function of time. The energy harvested during the time interval $[t_1, t_2)$ is denoted as $E_p(t_1, t_2) = \int_{t_1}^{t_2} R_p(t)\mathrm{d}t$. Based on [20, 22, 23], we make $R_p(t)$ to be a constant function and denote it as R. The energy harvesting during the time interval $[t_1, t_2)$ is denoted as $E_p(t_1, t_2) = (t_2 - t_1) \times R$ in the following. In addition, since the method proposed in this work is a general method, we only consider the consumption of electrical energy after energy conversion. Therefore, if the new energy is applied to the method proposed in this work, only this energy production model needs to be replaced.

3.1.2. Energy Storage Model. A battery is generally used as an energy storage device in a real-time embedded system. We suppose storage energy cannot be more than battery maximum capacity $C_{\max}$and use an ideal storage model that stores as much energy as is harvested, ignoring all losses. The energy charge of the energy storage unit at time t is expressed

as $E_s(t)$, and then $C_{\max} \geq E_s(t) \geq 0$ at any time t, where $E_s(0)$ is the initial energy level. The energy of the energy storage unit in the time interval $[t_1, t_2)$ is denoted as $E_s(t_1, t_2) = E_s(t_2) - E_s(t_1)$. When $E_s(t_1, t_2)$ is positive, it indicates that the energy storage unit is in the charging mode during the time interval $[t_1, t_2)$. In contrast, when $E_s(t_1, t_2)$ is negative, it indicates that the energy storage unit is in the discharge mode during the time interval $[t_1, t_2)$.

3.2. Task Model. In this work, the models and analysis methods used in this article are all oriented to fixed real-time embedded systems. In general, to ensure real-time performance, this system will not add new tasks; therefore, we consider a real-time system $P = \{\tau_1, \tau_2, \ldots, \tau_n\}$ of n independent tasks. Tasks in their task sets are periodic tasks. The task is a 5-tuple $(P_i, C_i, D_i, T_i, E_i)$, in which P_i is the task priority (in this work, P_1 is expressed as the maximum priority), C_i is the worst-case execution time, D_i is the relative task deadline, T_i is the task period, and E_i is the worst-case energy consumption (WCEEC). A periodic task τ_i generates an infinite number of real-time jobs, and each job consumes E_i energy units while executing C_i. The deadline of each period's task is constrained or implicit (i.e.$D_i \leq T_i$). The periodic task set is priority-ordered, the task τ_1 being the task with the highest priority task. In the time interval $[t_1, t_2]$, task consumption is denoted as $E_w(t_1, t_2)$. If the task can be scheduled in the time interval $[t_1, t_2]$, the task energy consumption $E_w(t_1, t_2)$ satisfies the following formula:

$$E_w(t_1, t_2) \leq E_s(t_1) + E_p(t_1, t_2). \quad (1)$$

4. Research Motivation

In this work, we focus on the ED issue of scheduling for EHES. When traditional time-constrained fixed-priority pre-emptive scheduling is directly leveraged by EHES, it may cause the originally feasible task set to become infeasible. We consider that after the system completes a hyperperiod task, the energy level variation has the following two cases:

(i) The replenished energy is lower than the total energy consumed. In this case, every time a hyperperiod passes, the stored energy will decrease until the task sequence is not feasible, causing the system to stop running.

(ii) The replenished energy is greater than or equal to the total energy consumed. In this case, every time a hyperperiod passes, storage energy will increase until reaching the maximum storage value of the storage unit.

However, in some situations, an ED issue may occur. For instance, we assume a task set includes two tasks $\tau_1(P_1 = 1, C_1 = 1, D_1 = 4, T_1 = 4, E_1 = 2)$ and $\tau_2(P_2 = 2, C_2 = 2, D_2 = 8, T_2 = 8, E_2 = 4)$, which are executed in a hyperperiod as shown in Figure 2. The task consumption power is $C_{pi} = E_i/C_i$ (where $C_{p1} = 2/1 = 2$, $C_{p2} = 4/2 = 2$), and the system power consumption is ignored in the idle state,

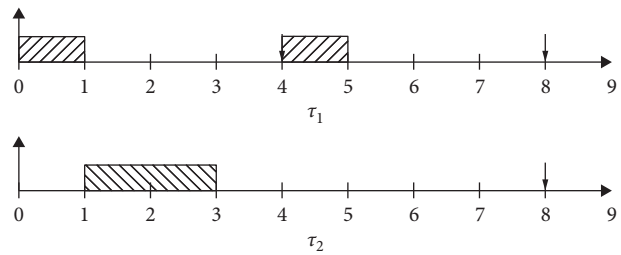


Figure 2: Task τ_1 and task τ_2 executed in a hyperperiod.

such that the energy production power $R = 1$. We can calculate that the total consumed energy (one hyperperiod) $E_t = 2 + 4 + 2 = 8$, and the production energy $E_p(0, 8) = (8 - 0) \times 1 = 8$. We can see that $E_t = E_p$, which appears as if the system will not stop due to insufficient energy. However, as shown in Figure 3, when we set the battery initial value $E_s(0) = 1$, at the time of $t = 2$, the system stops running because energy is exhausted. When we set the battery initial value $E_s(0) = 3$, the system can perform a complete hyperperiod task. Conclusively, although the total energy consumption is equal to the total production energy in some cases, it may still be insufficient energy due to overly frequent task executions in a busy period, or the energy consumption rate of a task is much greater than the energy generation rate. When the above problems occur, the system will stop running.

We found that setting the initial energy of the battery can effectively solve the above problems. Therefore, we propose HEE, WCRT-IELB, and AC-IELB algorithms in Section 6. The HEE algorithm is used to filter task sets that are not feasible under global energy. WCRT-IELB and AC-IELB algorithms are used to calculate the initial energy level of the battery to solve the task scheduling failure caused by the local energy shortage.

5. System Schedulability Analysis

The aim of this section is to characterize the system schedulability. Each task is mapped to a task process when implementing the scheduling. We assumed $\psi(i)$ is a mapping of the task τ_i. This mapping task process is offline and maintains constant during running. A scheduling implementation can be defined as a 2-tuple $I = (\Pi, \Psi)$ for a given system $S = \{\tau_1, \tau_2, \ldots, \tau_n\}$, where

(i) Π: $S \longrightarrow [1, \ldots, n]$ is a priority assignment for the tasks.

(ii) Ψ: $S \longrightarrow [\psi(1), \ldots, \psi(n)]$ is a mapping of tasks into processes.

A scheduling implementation I for a system S is feasible if the WCRT R_i for all tasks under the implementation I is no more than their deadlines D_i. The schedulability is given by equation (2) [28]. Then, a system S is said to be schedulable if there is a feasible implementation for it. In EHES, the WCRT is determined by the worst time requirement and the worst energy requirement together. The calculation is shown below.

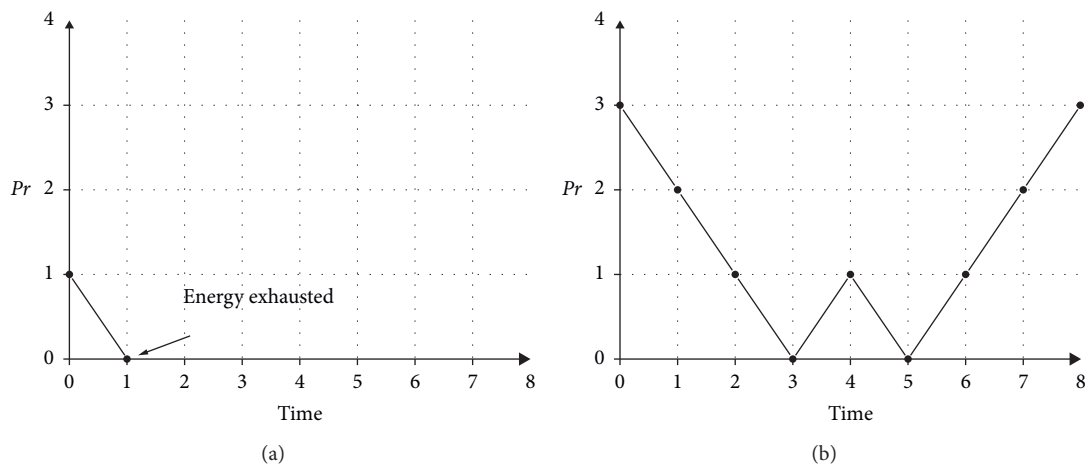


Figure 3: Energy consumption of task τ_1 and task τ_2 executed in a hyperperiod.

$$\text{Feasible}(I,P) \stackrel{\text{def}}{=} (\forall_i \in [1,\ldots,n]) R_i(I) \le D_i. \tag{2}$$

Definition 1. The time demand of the task τ_i in the time interval $[0,t]$ is denoted as $wp_i(t)$ in the worst case, which is the execution time of τ_i and the execution time of all tasks whose priority is higher than τ_i. It can be obtained by the following formula [22]:

$$wp_i(t) = \sum_{j \le i} \lceil \frac{t}{T_j} \rceil \times C_j. \tag{3}$$

Definition 2. The energy demand of the task τ_i in the time interval $[0,t]$ is denoted as $we_i(t)$ in the worst case, which is the energy of executing τ_i and the energy of executing all tasks whose priority is higher than τ_i. It can be obtained by the following formula:

$$we_i(t) = \sum_{1 \le j \le i} \lceil \frac{t}{T_j} \rceil \times E_j. \tag{4}$$

Definition 3. The WCRT of the task τ_i in the time interval $[0,t]$ on EHES is determined together by the time demand and energy demand of the task, and the biggest demand between them is the WCRT denoted as $w_i(t)$. It can be obtained by the following formula:

$$w_i(t) = R_i(I) = \max\left(wp_i(t), \lceil \frac{we_i(t)}{R} \rceil \right). \tag{5}$$

Theorem 1. *If after setting the initial battery level and each task of the task set P meets the demand $w_i(t) \le D_i$ in the worst case, the task set is feasible, then the system is schedulable.*

Proof of Theorem 1. We consider the initial battery level $E_s(0) \le C_{\max}$; then, energy demand in the worst case is

$$we_i(t) = \sum_{1 \le j \le i} \lceil \frac{t}{T_j} \rceil \times E_j - E_s(0). \tag{6}$$

According to Definition 3, we know that

$$w_i(t) = \max\left(wp_i(t), \lceil \frac{we_i(t)}{R} \rceil \right), \tag{7}$$

and $\lceil we_i(t)/R \rceil \ge wp_i(t)$ because in our model, $E_s(0) \ge 0, E_i \ge C_i \times R$. This reveals the fact that in our model, we must have replenishment periods that increase task response time (only aim at scheduling algorithms of considering energy, while WCRT of the traditional fixed-priority scheduling algorithm only considers time).

Assume $E_s(0) = 0$. Then,

$$t_s = w_i(t) = \lceil \frac{\sum_{1 \le j \le i} \lceil t/T_j \rceil \times E_j}{R} \rceil. \tag{8}$$

Similarly, assume $E_s(0) > 0$. Then,

$$t_{s1} = w_i(t) = \lceil \frac{\sum_{1 \le j \le i} \lceil t/T_j \rceil \times E_j - E_s(0)}{R} \rceil. \tag{9}$$

We obtain

$$t_s \ge t_{s1}. \tag{10}$$

While $t_s > D_i$ indicating that the task missed the deadline, that is, the system is unschedulable, otherwise, the system is schedulable ($t_s \le D_i$). Therefore, $t_s \ge t_{s1} \ge D_i$ indicates that system is unschedulable while $t_s \ge D_i \ge t_{s1}$ indicates that the system is schedulable. □

Theorem 2. *In the worst case, the energy demand by the EHES is greater than or equal to 0 which is a necessary and insufficient condition for the system to be schedulable. It can be expressed by the following formula :*

$$E(n) = n \times \left(hp \times R - \sum_{1 \le j \le i} \lceil \frac{hp}{T_j} \rceil \times E_j \right) + E(0), \tag{11}$$

where hp denotes the hyperperiod and n denotes the number of the hyperperiod.

Proof of Theorem 2

(1) Necessary condition: according to the above description, if the system is schedulable and all tasks must be completed before the deadline in the worst case, the system will not miss the deadline due to insufficient energy, that is, $E(n) \geq 0$, and the necessity is proved.

(2) Insufficient condition: when the production energy is less than the total consumption energy, it is not difficult to see that $E(n)$ is a monotonically decreasing function and that the battery level reduces as n increases. EHES has sufficient energy in the first few hyperperiods; however, as n increases, more tasks miss the deadline due to insufficient energy. Furthermore, even when $E(n) \geq 0$, the ED issue mentioned in the research motivation will stop the system, and the system is unschedulable. The insufficient condition is proved. □

6. Algorithms

In this section, we propose three algorithms to address the ED issue and improve the success rate of the task sets scheduled. The HEE algorithm aims at filtering the first case mentioned in the research motivation, while the WCRT-IELB and AC-IELB algorithms aim at adopting the initial energy value to eliminate the second case mentioned in the research motivation, namely, the ED issue.

6.1. HEE Algorithm. HEE, which is shown in Algorithm 1, is a general judgment that can make a preliminary judgment on the task set, which the main relies on equation 11 to calculate. Lines 4–9 show the total energy consumption accumulated over a hyperperiod. Line 10 compares the total energy consumped and the total energy producted; if it returns true, then it indicates the total energy producted is equal or lower than the total energy consumed. This case does not completely guarantee that the task set has enough energy. However, if it returns false, it indicates that the set of tasks is infeasible.

Because HEE is a preliminary judgment, we require WCRT-IELB to judge further. Although HEE cannot accurately determine whether task sets can be scheduled, it can exclude the majority of cases that cannot be scheduled and improve the operation efficiency of the following two algorithms.

6.2. WCRT-IELB Algorithm. WCRT-IELB, which is shown in Algorithm 2, first calculates the WCRT of the set task by formula (5) [22] online 3 and then calculates the execution times of each task during the WCRT and the total energy consumption of each task (lines 5–10). Finally, the total energy consumption of each task is accumulated. The total energy consumption and production energy are compared in line 11; if the production energy is lower than the total consumption energy, the absolute value of the difference between production and consumption is returned. This absolute value is the initial value required by the battery, and this value is not more than the battery maximum capacity.

However, this value is still not the most appropriate in some extreme cases. We assume that a task set includes two tasks τ_1 $(P_1 = 1, C_1 = 1, D_1 = 4, T_1 = 4, E_1 = 3)$ and τ_2 $(P_2 = 2, C_2 = 2, D_2 = 8, T_2 = 8, E_2 = 4)$ and energy production power $P_r = 1$. We can calculate that the WCRT is 3, where τ_1 and τ_2 are executed once, and during this interval of time, the total energy consumption is $3 + 4 = 7$ and the energy production is $1 \times 3 = 3$. Therefore, the initial value calculated by WCRT-IELB is $|3 - 7| = 4$. However, when this initial value is set, τ_1 will still stop running due to insufficient energy when it is executed in the second period (available energy $E_a = 4 + 5 = 9$ is less than consumption energy $E_c = 3 + 4 + 3 = 10$). Therefore, we propose a more accurate AC-IELB algorithm.

6.3. AC-IELB Algorithm. AC-IELB, which is shown in Algorithm 3, determines how to accurately calculate the initial value of the battery when a set of tasks is ready to run. AC-IELB first chooses the highest priority task and then calculates the task energy consumption at the time unit (E_i/C_i) and compares it with the current energy level $(E(t))$ plus production energy (R) at the time unit.

The AC-IELB can be divided into three cases. In the first case (lines 8–10), the available energy of the system is greater than energy consumption, and the system can perform tasks. In the second case (lines 11–14), the available energy is lower than the energy consumption, and the system does not have enough energy to perform tasks. AC-IELB will calculate the difference between the available energy and energy consumption and accumulate this difference to the initial battery level value, and AC-IELB will reset the initial battery level value and run again. In the third case (lines 15–18), the energy consumption at the time unit is equal to the energy production rate, and the current energy level is 0. At the next moment, we cannot guarantee that the system first consumes energy, produces energy, or both; therefore, we accumulate an additional single unit of energy to ensure the normal operation of the system. Then, the first task in the task set is deleted and the execution is repeated until all tasks in the task set have been executed. AC-IELB can address the cases where the initial value calculated by WCRT-IELB is not appropriate.

We consider a task set as shown in Table 1. In time intervals $[0, 55]$, the scheduling result is shown in Figure 4. In this example, we set the energy production power $R = 15$, the battery initial value $E(0) = 20$, and the battery maximum capacity $C_M = 300$. Using the ALAP scheduling algorithm, the scheduling result is shown in Figure 4(a). At time $t = 9$, there is a shortage of available energy; therefore, the task τ_4 stops executing. The initial value of the battery $E(0) = 12$ (the WCRT is 20) is calculated by WCRT-IELB, and the initial value of the battery is reset to run again. The

```
Input: A ⟵ set of n active tasks at time t
Output: true or false
function GLOBAL CALCULATION (A)
  hp ⟵ calculating hyperperiod of the task set A
  sum ⟵ 0
  for i = 1; i ≤ n; i++do
    take the i^th task of A as τ_i
    E_i ⟵ energy cost of τ_i
    T_i ⟵ period of τ_i
    sum ⟵ E_i×⌈hp/T_i⌉ + sum
  end for
  if (sum−hp×R) < 0 then
    return false
  else
    return true
  end if
end function
```

Algorithm 1: Holistic energy evaluation.

```
Input: A ⟵ set of n active tasks at time t
Output: true or false
function WORSTCASECALCUATION (A)
  τ_l ⟵ the lowest priority task of A
  wt ⟵ WorstCaseResponseTime (τ_l)
  sum ⟵ 0
  for i = 1; i < n; i++ do
    task the i^th task of A as τ_i
    Ei ⟵ energy cost of τ_i
    T_i ⟵ period of τ_i
    sum ⟵ E_i×⌈wt/T_i⌉ + sum
  end for
  if (wt×R−sum) < 0 then
    if |wt × R − sum| ≥ C_max then
      return C_max
    else
      return |wt × R − sum|
    end if
  end if
end function
function SCHEDULABILITYJUDGMENT
  iv_i ⟵ WorstCaseCalcuation (A)
  set iv_i and task set A and execute schedule algorithm
  if Scheduling algorithm is schedulable then
    return true
  else
    return false
  end if
end function
```

Algorithm 2: WCRT-based initial energy level of battery.

scheduling result is shown in Figure 4(b). When the system runs on time $t = 8$, the task τ_1 stops executing due to a shortage of available energy. Until the most accurate AC-IELB is used to calculate the battery initial value $E(0) = 158$, the task set can be scheduled in the time interval $[0, 55]$. The scheduling result is shown in Figure 4(c).

```
Input: A ⟵ set of active tasks at time t, sum ⟵ 0
Output: ture of false
function CalculateInitialValue(A; sum)
  t ⟵ 0
  loop
    τ_i ⟵ the first task of A
    E_i ⟵ remaining energy cost of the τ_i at time t
    C_i ⟵ remaining execution time of the τ_i at time t
    if A ≠ ϕ then
      if E_i/C_i < E(t) + R then
        t ⟵ t + 1
      end if
      if E_i/C_i > E(t) + R then
        sum ⟵ |(E_i/C_i) − E(t) − R|+sum
        CalculateInitialValue(A, sum)
      end if
    if E_i/C_i = R & E(t) = 0 then
        sum ⟵ sum + 1
        CalculateInitialValue(A, sum)
      end if
    end if
    A ⟵ remove the first task of A
  end loop
  if result ≥ C_max then
    return C_max
  else
    return result
  end if
end function
function SCHEDULABILITYJUDGMENT
  iv_i ⟵ CalculateInitialValue(A, sum)
  set the iv_i and task set A, and execute schedule algorithm
  if Scheduling algorithm is schedulable then
    return true
  else
    return false
  end if
end function
```

Algorithm 3: Accurate cycle-initial energy level of battery.

Table 1: Task set τ_t.

-	C_i	E_i	T_i	D_i	P_i
τ_1	3	150	36	36	1
τ_2	1	100	10	10	2
τ_3	2	20	24	24	3
τ_4	1	18	30	30	4

7. Simulation and Evaluation

In this section, we describe the design and implementation of the experiment from the simulation tool, input data, simulation duration, evaluation metrics, and result analysis.

7.1. Simulation Tool. In this work, to evaluate the effectiveness of the battery initial value for scheduling algorithms, we randomly generated a large number of periodic task sets and verified them with the ALAP and ASAP. We used YARTISS [29, 30] as the simulation experiment environment and conducted secondary development on it. It provides a simulation framework and implements many scheduling algorithms that can simulate different task sets under different energy parameters for EHES.

7.2. Input Data. YARTISS uses an adapted version of the UUniFast-Discard algorithm [31] coupled with a limitation of the hyperperiod technique to generate task sets. We use this function to generate 250 task sets. Each task set contains 4 tasks, and the range of [0, 200] is selected for each attribute of the task. Task sets are time feasible.

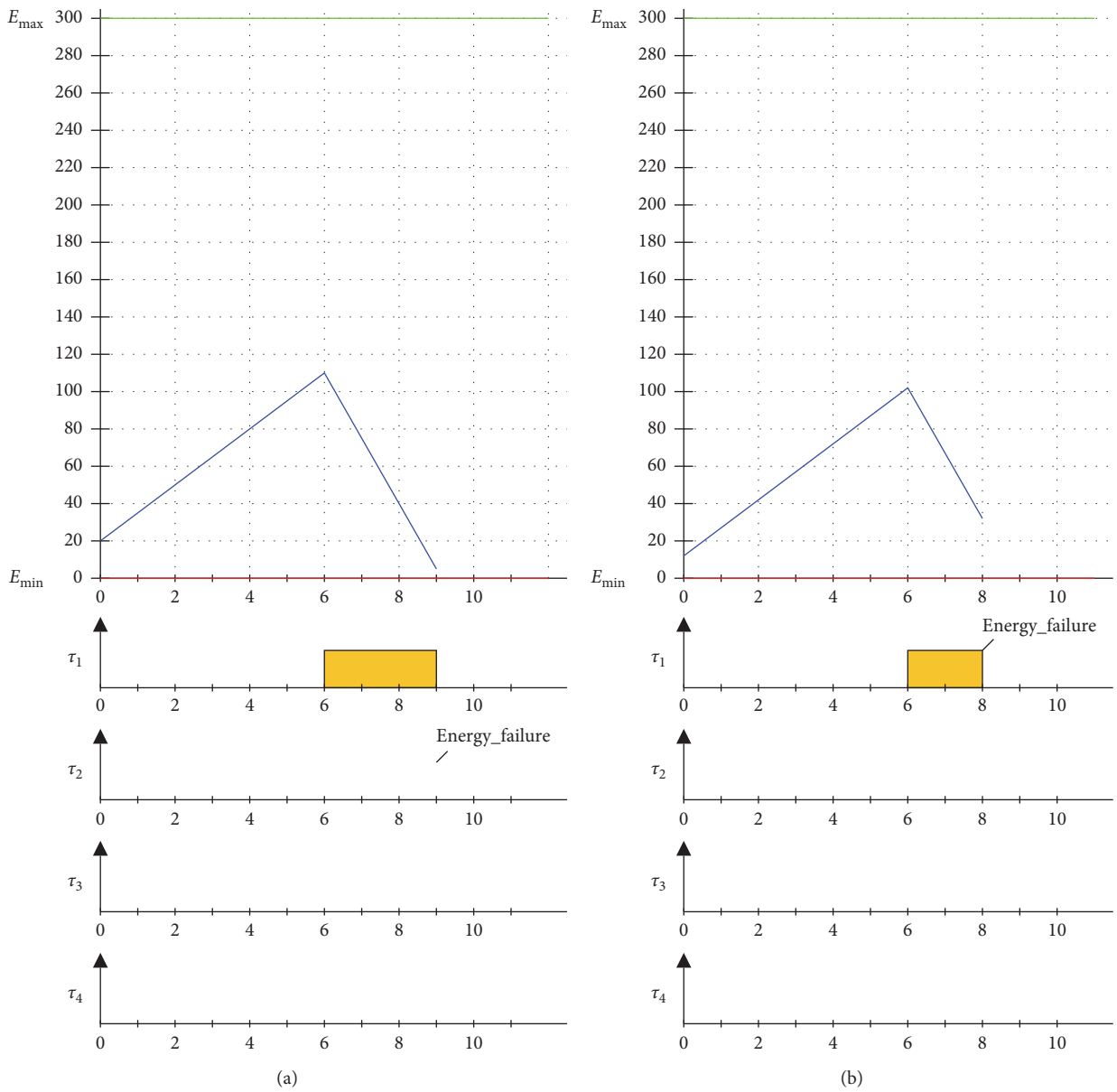
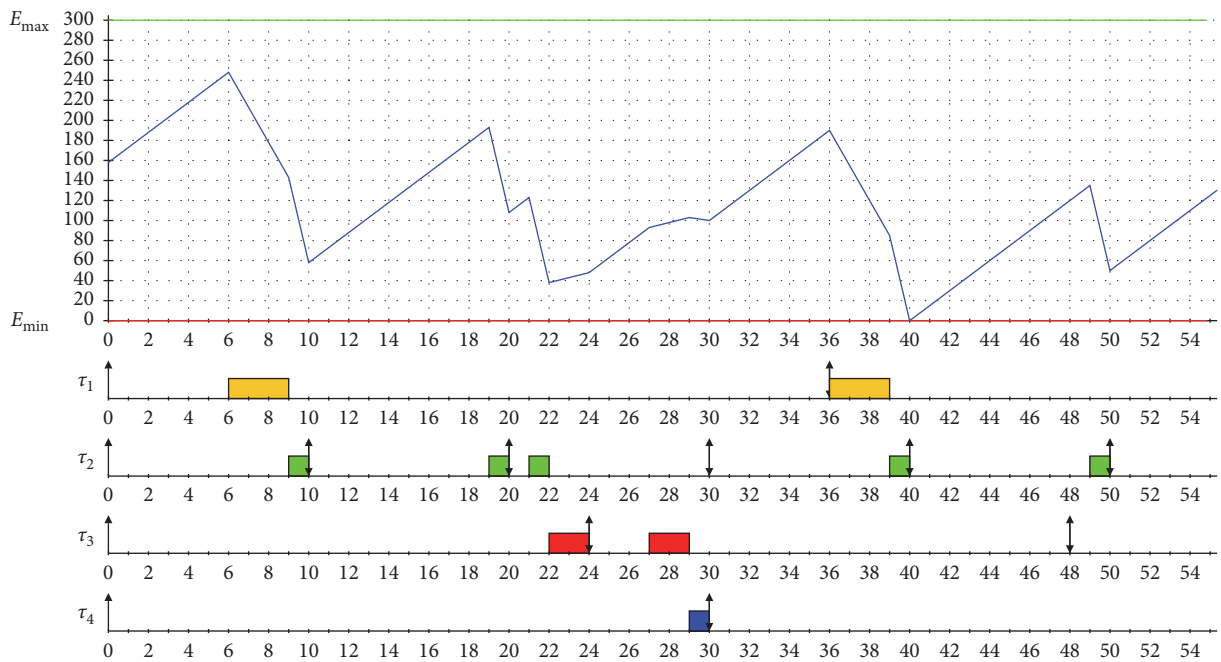


Figure 4: Continued.



(c)

Figure 4: Running status of task set τ_T with different initial energies. (a) Fixed initial energy (20). (b) WCRT-IELB algorithm calculating initial energy. (c) AC-IELB algorithm calculating initial energy.

7.3. Parameter Setting. In ALAP and ASAP, the influence of the battery initial value setting is compared. First, we vary the battery capacity to analyse the impact on the success rate. Second, we conducted 250 groups of experiments to observe the success rate of each algorithm under different battery capacities.

We set the same common parameters to ensure the correctness of the simulations. These parameters are set as follows: energy production power $R = 15$, the battery storage minimum energy $E_{\min} = 0$, the battery maximum capacity $C_M = \{50, 100, 200, 300, 400\}$, and the simulation execution time Duration = 2560. We perform three types of simulations with different initial energy levels on ALAP and ASAP.

(1) Sitting a fixed initial energy level of the battery ($E(0) = 20$).

(2) Setting the battery's initial energy level based on WCRT-IELB.

(3) Setting the battery's initial energy level based on AC-IELB.

7.4. Evaluation Metrics

7.4.1. Average Success Rate. We define the average success rate SR_a shown in equation (12) to evaluate the three algorithms, where T_f denotes the number of feasible task sets and T_a denotes the number of all task sets. We conducted 250 groups of experiments and divided them into 5 parts on average and calculated the success rate of each group (SRG_a), evaluating the average by formula (13), where T_{f_i} denotes the $i-th$ group of the number of the feasible task sets and T_{a_i} denotes the $i-th$ group of the number of all task sets.

$$SR_a = \frac{T_f}{T_a}, \tag{12}$$

$$SRG_a = \frac{1}{n}\sum_{i=0}^{n}\frac{T_{f\_i}}{T_{a\_i}}. \tag{13}$$

7.4.2. Average Energy Level. The average energy level is the average energy percentage of the battery or capacitor during the simulation. The higher the average energy level, the lower the energy limit of the system.

7.4.3. Average Overhead. It is the average time taken to execute a scheduled event during the simulation. The greater the average overhead is, the more likely the task will miss the task deadline.

7.5. Result Analysis. We use the WCRT-IELB and AC-IELB algorithms to calculate the initial battery capacity of 250 task sets under ALAP and ASAP. Take ALAP as an example here, as shown in Figure 5. The maximum battery capacity is 50, 100, 200, 300, and 400. As depicted, since the calculation method of WCRT-IELB determines the initial battery level based on the size of the busy period, most of the initial battery levels have reached the maximum battery capacity; although the initial battery level calculated by the AC-IELB algorithm also accounts for a large part of the maximum

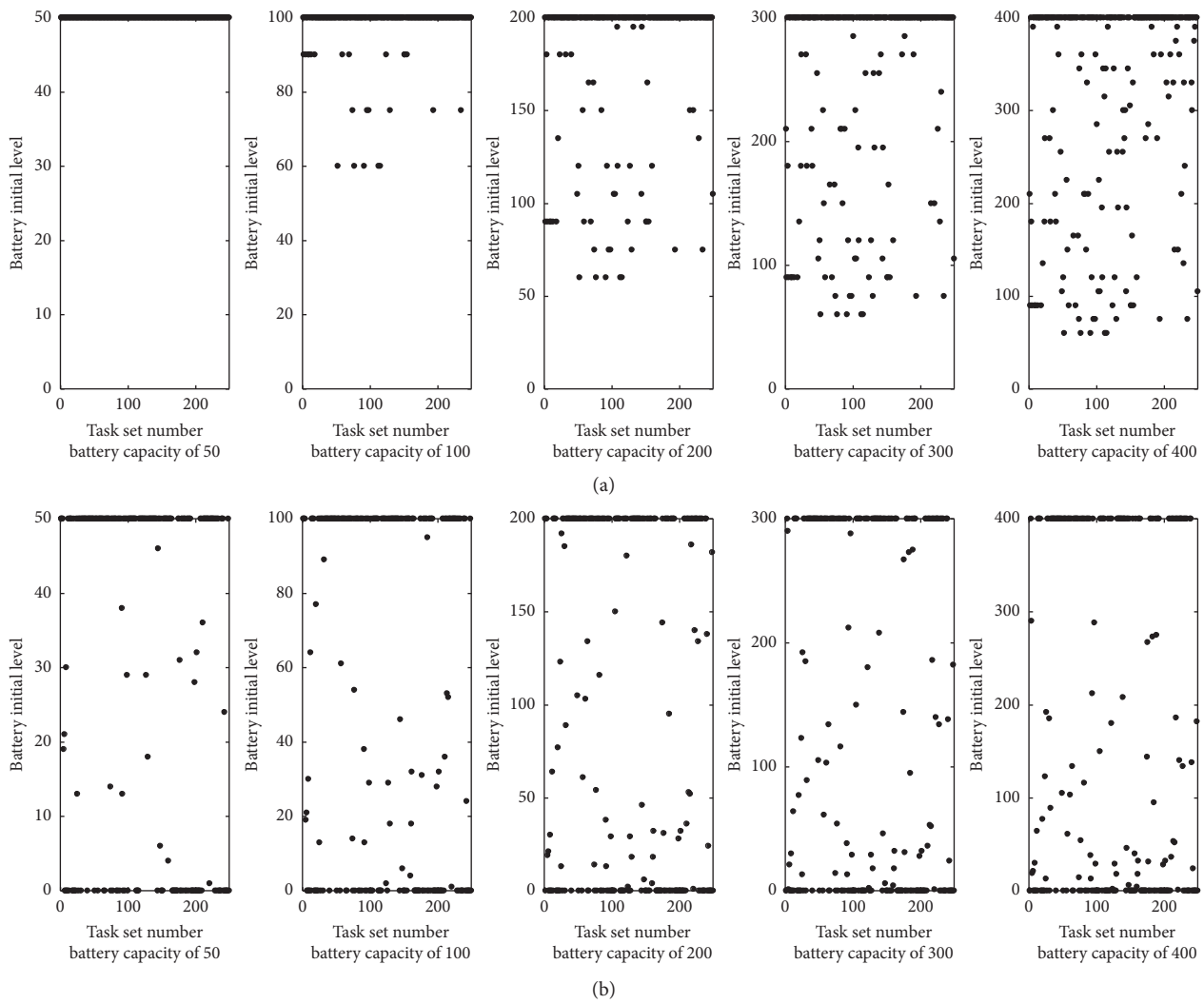


Figure 5: The ALAP scheduling algorithm calculating the initial battery level based on the WCRT-IELB and AC-IELB algorithms. (a) WCRT-IELB algorithm. (b) AC-IELB algorithm.

battery capacity, with the increase of the maximum battery capacity, this situation has eased. And the situation where the initial battery level is 0 is gradually increasing. This is because the ALAP scheduling algorithm is less affected by the initial battery level and is more affected by the maximum battery capacity. When the maximum capacity of the battery increases, the schedulability of the ALAP scheduling algorithm is gradually reduced by the initial level of the battery. Moreover, we found that when the battery capacity is 50, 100, and 200, the battery initial level calculated by the WCRT-IELB and AC-IELB algorithms has reached the maximum battery capacity in most cases, until the battery capacity is increased to 300 and 400. This situation began to ease. This is because according to the task set, it is calculated that the actual required battery initial level is greater than the maximum battery capacity. We have verified this in subsequent experiments. When the maximum battery capacity is 50, 100, and 200, the success rate of the task set is very low. It was not until the maximum battery capacity was increased to 300 and 400 that the success rate increased significantly. Most task sets did not achieve the proper initial battery level.

7.5.1. Average Success Rate. The 250 task sets were tested with the ALAP and ASAP in the following two scenarios.

Scenario 1: make 250 task sets run as a group with battery capacities of 50, 100, 200, 300, and 400.

Figure 6 shows the success rate of three different battery initial level-setting methods (fixed, WCRT-IELB algorithm calculation, and AC-IELB algorithm calculation) for ALAP and ASAP. The black line represents the scheduling algorithm that is set to run at a fixed initial battery level of 20, the red line represents the scheduling algorithm that uses WCRT-IELB to set the initial battery level, and the blue line represents the scheduling algorithm that uses AC-IELB to set the initial battery level.

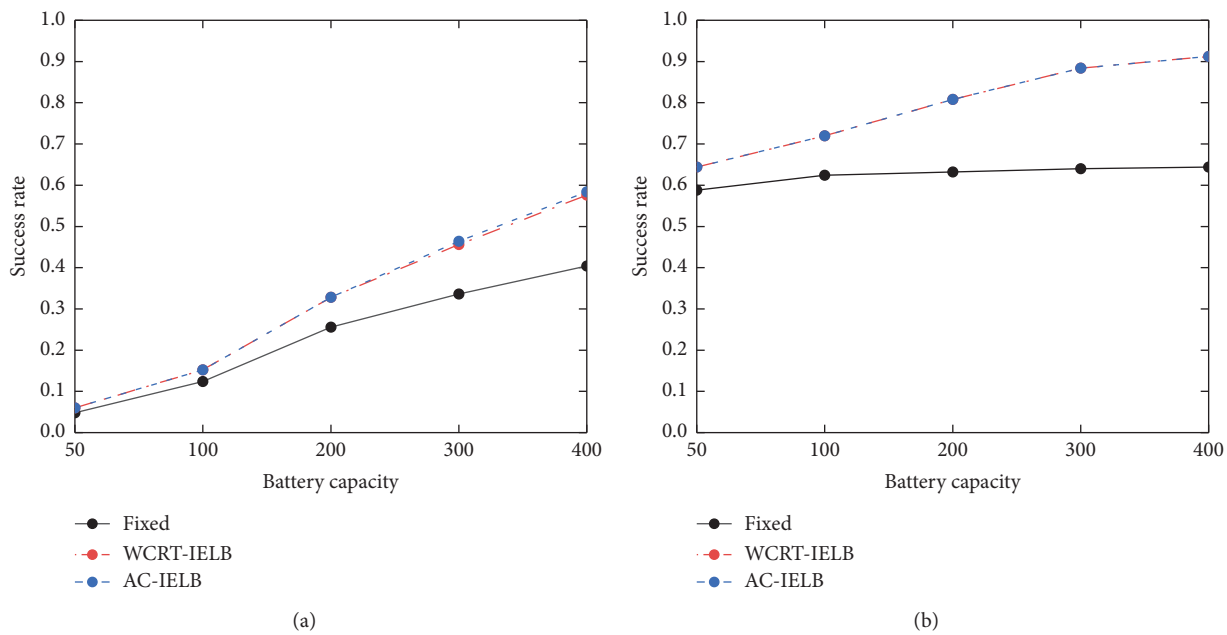


Figure 6: The success rate of 250 experiments. (a) ALAP scheduling algorithm. (b) ASAP scheduling algorithm.

As depicted, we propose AC-IELB and WCRT-IELB algorithm to applicate the ALAP and ASAP that performance is better than schedulling algorithms based on fixed model settings. This is expected; as Section 4 describes, the scheduling algorithms on their busy period have the most energy consumption; however, using this method to calculate the initial battery level in some very extreme cases is not precise, in which case we adopt the AC-IELB for every task to calculate.

On the other hand, the ALAP scheduling algorithm is less affected by the initial battery level and is more affected by the maximum battery capacity, and with the increase of maximum battery capacity, the success rate increases. It has a big rise tendency from a battery capacity of 100 to a battery capacity of 400. Compared with the ALAP scheduling algorithm based on fixed model settings, the success rate of the WCRT-IELB algorithm under battery capacity of 400 increased by 17.2%, while the AC-IELB algorithm increased by 18%. For the ASLP scheduling algorithm, the scheduling algorithm based on fixed model settings remain stable success rate form battery capacity of 50 to 400, suffering rarely fluence from battery capacity, ALAP scheduling algorithm based on AC-IELB and WCRT-IELB algorithm under form battery capacity of 50 to 200, its success rate has a big rise. In addition, by a battery capacity of 300 and a battery capacity of 400, its success rate is basically the same. Compared with the ASAP scheduling algorithm based on fixed model settings, the success rate of the AC-IELB and WCRT-IELB algorithms under battery capacity of 400 increased by 26.8%.

Scenario 2: divide 250 task sets into five groups and run with battery capacities of 50, 100, 200, 300, and 400.

Figure 7 compares the success rates of ALAP and ASAP using three different methods (fixed, WCRT-IELB algorithm calculation, and AC-IELB algorithm calculation) to obtain the initial battery capacity under five different maximum battery capacities. The black line represents the scheduling algorithm running at a fixed initial battery level of 20, the red line represents the scheduling algorithm that adopts the WCRT-IELB algorithm to set the initial battery level, and the blue line represents the scheduling algorithm that adopts the AC-IELB algorithm to set the initial battery level.

To begin with, for the ALAP scheduling algorithm, as depicted, the maximum capacity of the battery has a significant impact on the ALAP scheduling algorithm (the success rate of the ALAP scheduling algorithm with initial battery level increases as the maximum battery capacity increases). When the maximum battery capacity is 50, the success rate is low. Setting the battery's initial level has little significance. This is because the ALAP scheduling algorithm is limited by the battery capacity. Through calculation, most of the initial battery levels that we obtain are more than 50. As the maximum battery capacity increases, the success rate gradually increases, and the effect of using the WCRT-IELB and AC-IELB algorithms to set the initial level continues to improve. Overall, the increase in the number of tasks has little effect on the success rate of the ALAP scheduling algorithm, fluctuating between 5% and 18%.

On the other hand, for the ASAP scheduling algorithm, considering the ASAP algorithm with the fixed initial level, the maximum battery capacity has little effect on it. This is due to the unique scheduling strategy of the ASAP algorithm, which causes the battery energy level to remain relatively low. When the battery capacity is 50 or 100, the success rate of the ASAP scheduling algorithm changes in the same way. When the battery capacity is 200, 300, and 400, compared with the ASAP scheduling algorithm based

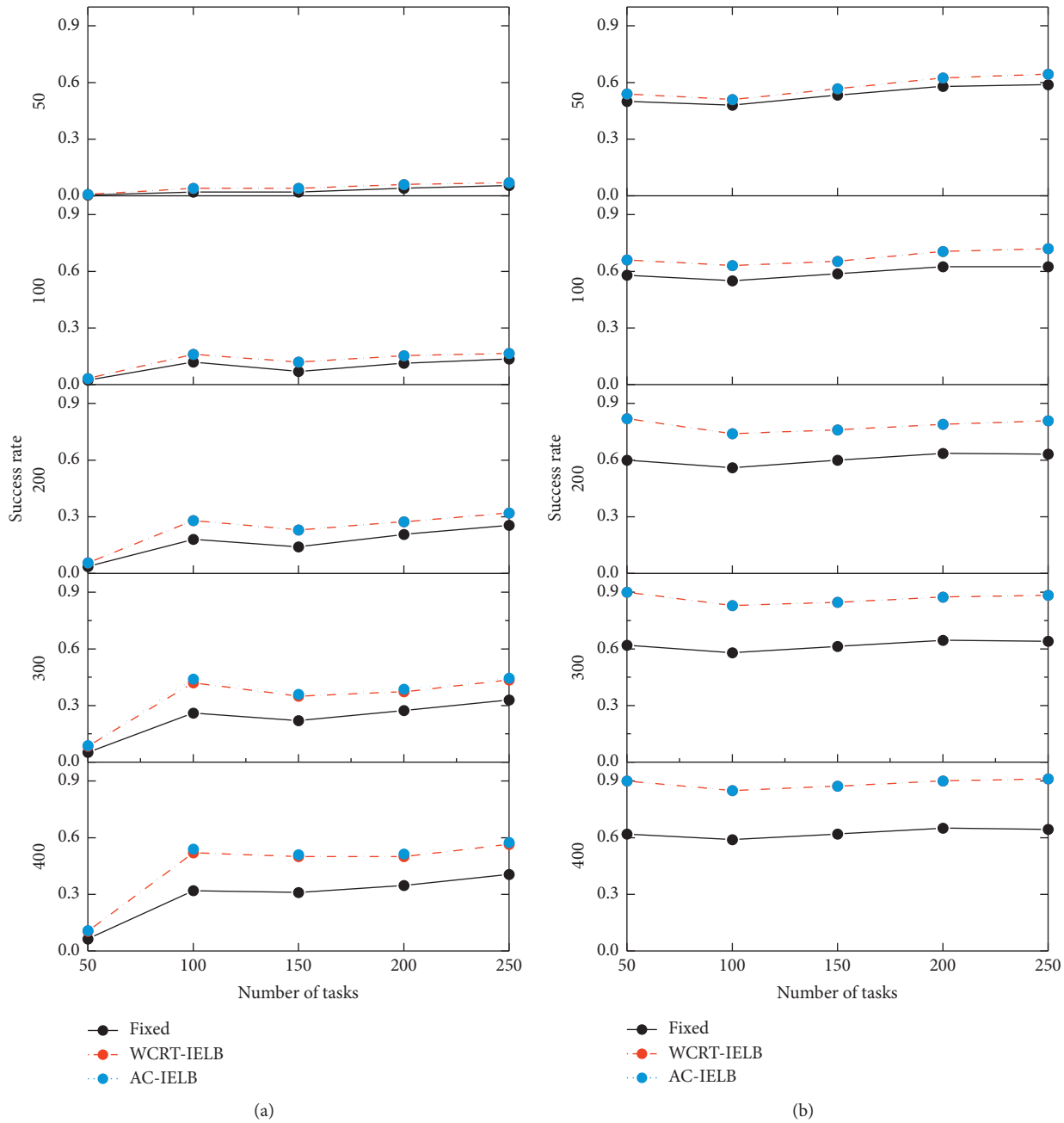


Figure 7: Success rate with a battery capacity of 50, 100, 200, 300, and 400. (a) ALAP scheduling algorithm. (b) ASAP scheduling algorithm.

on fixed model settings, the success rate of the ASAP scheduling algorithm based on the WCRT-IELB and AC-IELB algorithms to calculate the initial level is greatly improved, the success rate increases with the increase in number of task sets, and the task changes are relatively stable. Through experiments, we found that the overall energy level during the operation of the ASAP scheduling algorithm is low. If the busy period consumes large energy, there is not enough energy to run the task before the deadline, which requires a relatively large initial battery level. Therefore, when the battery capacity is high, the performance of the ASAP scheduling algorithm based on WCRT-IELB and AC-IELB is better than the ASAP scheduling algorithm with a fixed initial value.

7.5.2. Average Energy Level. As shown in Figure 8, we observe that the average energy level with adopting the WCRT-IELB and AC-IELB algorithm is higher than with fixed method settings on the ALAP (Figure 8(a)) and ASAP (Figure 8(b)), the primary reason is that we setting an initial battery level. Moreover, the increase of average energy level with the battery capacity increase. The average energy level of adopting AC-IELB algorithm is lower than

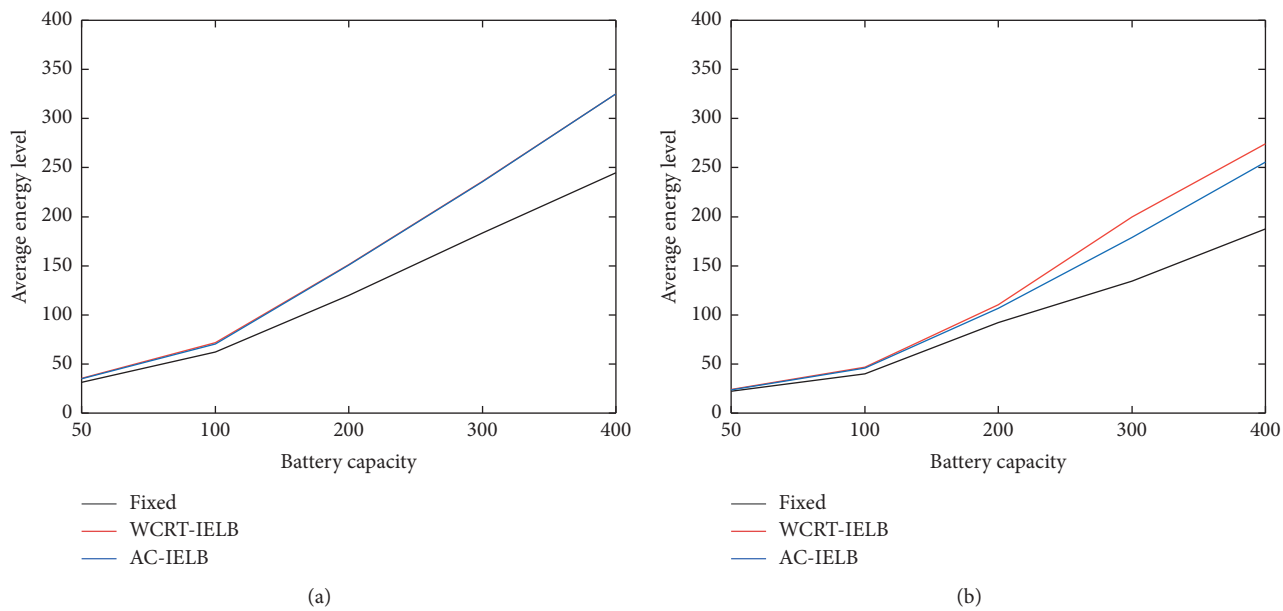


Figure 8: Average energy level.

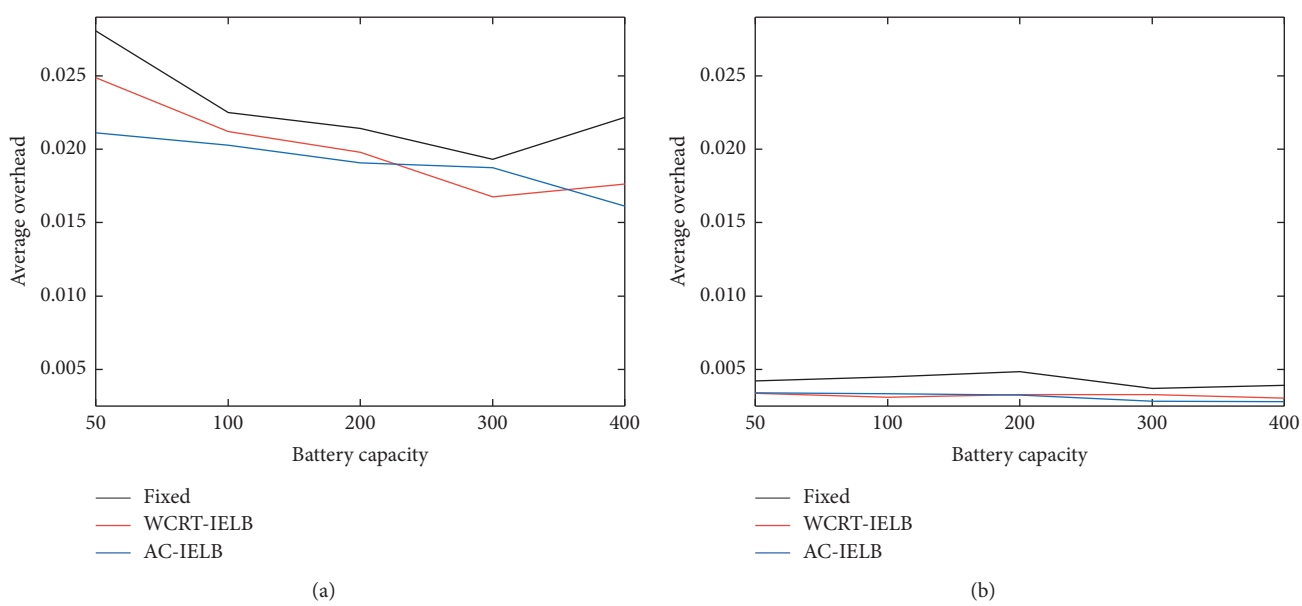


Figure 9: Average overhead.

adopting WCRT-IELB algorithm on the ASAP scheduling algorithm. Since compared with WCRT-IELB algorithm, AC-IELB algorithm calculate the initial battery level is more precise.

7.5.3. Average Overhead. As shown in Figure 9, we observe that the average overhead of adopting WCRT-IELB and AC-IELB algorithms is lower than that use initial battery level on ALAP (Figure (9a)) and ASAP (Figure 9b)). Also, these two different methods have the same tendency, which reduces with the increase of battery capacity.

8. Conclusions and Future Works

In this work, we proposed a filter algorithm named HEE that aimed to remove the infeasible task set of EHES, and we proposed two algorithms named WCRT-IELB and AC-IELB that aimed to improve the success rate of the scheduling algorithm to use the battery initial level to solve the ED problem. From the experiment, we can see that the best performance of the three scheduling algorithms without a proper battery initial level is achieved by employing the ASAP scheduling algorithm, which has a success rate of 60% at the maximum battery capacity, while the success rate

reached 97.2% after introducing the WCRT-IELB and AC-IELB algorithms. We ascribe this case to two problems: ED and the limitation of the maximum battery capacity. As a result, we found that the proper battery initial level and maximum battery capacity could improve the success rate of scheduling algorithms of EHES; however, the effect of this improvement depends on the strategy of scheduling algorithms.

In future work, a valuable endeavour is to calculate a suitable maximum capacity of the battery by implementing the algorithm and combining it with the battery's initial energy level to further improve the success of the scheduling algorithm. We will also try to build a real-world platform to collect real data and try to test the practicality of our proposed algorithms. Furthermore, the EHES computing unit based on multitask scheduling discussed in this work is a discrete computer system; thence, we will consider to research and discuss algorithms of this work on the continuous system.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Authors' Contributions

Jiayuan Wei's contribution in the experiment is equivalent to Xingyu Miao.

Acknowledgments

This study was supported in part by the Young Scholar in Western China of Chinese Academy of Sciences under grant no. XAB2018AW12, in part by the Ningxia Key Research and Development Projects under grant no. 2018BEB04020, and in part by the National Natural Science Foundation of China under grant no. 61862049.

References

[1] R. Norouzi, A. Kosari, and M. H. Sabour, "Real time estimation of impaired aircraft flight envelope using feedforward neural networks," *Aerospace Science and Technology*, vol. 90, pp. 434–451, 2019.

[2] S. Kato, S. Tokunaga, Y. Maruyama et al., "Autoware on board: enabling autonomous vehicles with embedded systems," in *Proceedings of the 2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*, pp. 287–296, IEEE, Porto, Portugal, April 2018.

[3] A. Sharma and P. Sharma, *Energy Harvesting Technology for IoT Edge Applications, in Smart Manufacturing-When Artificial Intelligence Meets the Internet of Things*, IntechOpen, London, UK, 2021.

[4] C. Psomas and I. Krikidis, "Wireless powered mobile edge computing: offloading or local computation?" *IEEE Communications Letters*, vol. 24, no. 11, pp. 2642–2646, 2020.

[5] P. Cong, J. Zhou, L. Li, K. Cao, T. Wei, and K. Li, "A survey of hierarchical energy optimization for mobile edge computing," *ACM Computing Surveys*, vol. 53, no. 2, pp. 1–44, 2020.

[6] Y. Hao, J. Cao, Q. Wang, and J. Du, "Energy-aware scheduling in edge computing with a clustering method," *Future Generation Computer Systems*, vol. 117, pp. 259–272.

[7] M. Malewski, D. M. Cowell, and S. Freear, "Review of battery powered embedded systems design for mission-critical low-power applications," *International Journal of Electronics*, vol. 105, pp. 893–909, 2018.

[8] S. Tzilis, P. Trancoso, and I. Sourdis, "Energy-Efficient runtime management of heterogeneous multicores using online projection," *ACM Transactions on Architecture and Code Optimization*, vol. 15, no. 4, pp. 1–26, 2019.

[9] M. Chetto and H. E. Ghor, "Scheduling and power management in energy harvesting computing systems with real-time constraints," *Journal of Systems Architecture*, vol. 98, pp. 243–248, 2019.

[10] S.-H. Lim, S. W. Lee, M. Sohn, and B.-H. Lee, "Queueing analysis of dynamic power management schemes for mobile devices," *IEEE Access*, vol. 8, pp. 97632–97642, 2020.

[11] N. Chawla, A. Singh, H. Kumar, M. Kar, and S. Mukhopadhyay, "Securing IoT devices using dynamic power management: machine learning approach," *IEEE Internet of Things Journal*, p. 1, 2020.

[12] D. Hosahalli and K. G. Srinivas, "Enhanced reinforcement learning assisted dynamic power management model for internet-of-things centric wireless sensor network," *IET Communications*, vol. 14, no. 12, pp. 3748–3760, 2020.

[13] G. L. Stavrinides and H. D. Karatza, "An energy-efficient, QoS-aware and cost-effective scheduling approach for real-time workflow applications in cloud computing systems utilizing DVFS and approximate computations," *Future Generation Computer Systems*, vol. 96, pp. 216–226, 2019.

[14] A. Toor, S. u. Islam, N. Sohail et al., "Energy and performance aware fog computing: a case of DVFS and green renewable energy," *Future Generation Computer Systems*, vol. 101, pp. 1112–1121, 2019.

[15] C. Zhuo, S. Luo, H. Gan, J. Hu, and Z. Shi, "Noise-Aware DVFS for efficient transitions on battery-powered IoT devices," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 7, pp. 1498–1510, 2020.

[16] D. Balsamo, B. J. Fletcher, A. S. Weddell, G. Karatziolas, B. M. Al-Hashimi, and G. V. Merrett, "Momentum: power-neutral performance scaling with intrinsic MPPT for energy harvesting computing systems," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 17, 2019.

[17] M. Hasanloo and M. Kargahi, "Harvesting-aware charge management in embedded systems equipped with a hybrid electrical energy storage," *Computers & Electrical Engineering*, vol. 69, pp. 98–114, 2018.

[18] J. Kwak, K. Lee, T. Kim, J. Lee, and I. Shin, "Battery aging deceleration for power-consuming real-time systems," in *Proceedings of the 2019 IEEE Real-Time Systems Symposium (RTSS)*, pp. 353–365, IEEE, Houston, TX, USA, May 2019.

[19] A. Allavena and D. Mossé, "Scheduling of frame-based embedded systems with rechargeable batteries, in workshop on power management for real-time and embedded systems," (in Conjunction with RTAS 2001) http://igm.univ-mlv.fr/masson/pdfANDps/allavena_mosse_01.pdf, 2001.

[20] C. Moser, D. Brunelli, L. Thiele, and L. Benini, "Lazy scheduling for energy harvesting sensor nodes," in *Proceedings of the IFIP Working Conference on Distributed and*

Parallel Embedded Systems, pp. 125–134, Springer, Milano, Italy, September 2006.

[21] R. Jayaseelan, T. Mitra, and X. Li, "Estimating the worst-case energy consumption of embedded software," in *Proceedings of the 12th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'06)*, pp. 81–90, IEEE, San Jose, CA, USA, April 2006.

[22] Y. Abdeddaïm, Y. Chandarli, and D. Masson, "The optimality of PFPasap algorithm for fixed-priority energy-harvesting real-time systems," in *Proceedings of the 2013 25th Euromicro Conference on Real-Time Systems*, pp. 47–56, IEEE, Los Alamitos, CA, USA, July 2013.

[23] Y. Chandarli, Y. Abdeddaïm, and D. Masson, "The fixed priority scheduling problem for energy harvesting real-time systems," in *Proceedings of the 2012 IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, pp. 415–418, IEEE, Seoul, South Korea, August 2012.

[24] Y. Abdeddaïm, Y. Chandarli, and D. Masson, "Toward an optimal fixed-priority algorithm for energy-harvesting real-time systems," in *Proceedings of the RTAS 2013 WiP*, pp. 45–48, Montreal, QC, Canada, January 2013.

[25] T. Kim, "Application-driven low-power techniques using dynamic voltage scaling," in *Proceedings of the 12th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA'06)*, pp. 199–206, IEEE, Piscataway, NJ, USA, August 2006.

[26] Y. Abdeddaïm, Y. Chandarli, R. I. Davis, and D. Masson, "Schedulability analysis for fixed priority real-time systems with energy-harvesting," in *Proceedings of the 22nd International Conference on Real-Time Networks and Systems*, pp. 311–320, Versaille, France, October 2014.

[27] E. Ghadaksaz and S. Safari, "Storage capacity for EDF-ASAP algorithm in energy-harvesting systems with periodic implicit deadline hard real-time tasks," *Journal of Systems Architecture*, vol. 89, pp. 10–17, 2018.

[28] Y. Ge, Y. Dong, and H. Zhao, "Energy-efficient task scheduling and task energy consumption analysis for real-time embedded systems," in *Proceedings of the 2014 Theoretical Aspects of Software Engineering Conference*, pp. 135–138, IEEE, Changsha, China, September 2014.

[29] Y. Chandarli, M. Qamhieh, F. Fauberteau, and D. Masson, "Yartiss: a generic, modular and energy-aware scheduling simulator for real-time multiprocessor systems," 2014, https://hal.archives-ouvertes.fr/hal-01076022/file/journal.pdf.

[30] Y. Chandarli, F. Fauberteau, D. Masson, S. Midonnet, and M. Qamhieh, "Yartiss: a tool to visualize, test, compare and evaluate real-time scheduling algorithms," 2012, https://hal-upec-upem.archives-ouvertes.fr/hal-00691985/document.

[31] P. Emberson, R. Stafford, and R. I. Davis, "Techniques for the synthesis of multiprocessor tasksets," in *Proceedings 1st International Workshop on Analysis Tools and Methodologies for Embedded and Real-Time Systems (WATERS 2010)*, pp. 6–11, Brussels, Belgium, June 2010.

Hindawi
Complexity
Volume 2021, Article ID 6686563, 12 pages
https://doi.org/10.1155/2021/6686563

Research Article

A Two-Stage Offline-to-Online Multiobjective Optimization Strategy for Ship Integrated Energy System Economical/Environmental Scheduling Problem

Qing An,[1] Jun Zhang,[2] Xin Li,[3,4] Xiaobing Mao,[3] Yulong Feng,[5] Xiao Li,[5] Xiaodi Zhang,[6] Ruoli Tang,[3,4] and Hongfeng Su[7]

[1]*Artificial Intelligence School, Wuchang University of Technology, Wuhan 430223, China*
[2]*Zhejiang Electronic Information Products Inspection and Research Institute (Key Laboratory of Information Security of Zhejiang Province), No. 50 Tian Mu Shan Road, Hangzhou, China*
[3]*School of Energy and Power Engineering, Wuhan University of Technology, Wuhan 430063, China*
[4]*Key Lab. of Marine Power Engineering and Tech. Authorized by MOT, Wuhan 430063, China*
[5]*Shanghai Marine Diesel Engine Research Inistute, Shanghai, China*
[6]*State Grid Beijing Electric Maintenance Company, Beijing 100080, China*
[7]*Sichuan Vocational and Technical College of Communications, Chengdu 611130, China*

Correspondence should be addressed to Jun Zhang; zj@zdjy.org.cn, Xin Li; xinli0503@hotmail.com, Xiaobing Mao; maoxiaobing2009@163.com, and Hongfeng Su; 568120525@qq.com

Received 3 December 2020; Revised 5 February 2021; Accepted 5 March 2021; Published 17 March 2021

Academic Editor: Chen Wang

The economical/environmental scheduling problem (EESP) of the ship integrated energy system (SIES) has high computational complexity, which includes more than one optimization objective, various types of constraints, and frequently fluctuated load demand. Therefore, the intelligent scheduling strategies cannot be applied to the ship energy management system (SEMS) online, which has limited computing power and storage space. Aiming at realizing green computing on SEMS, in this paper a typical SIES-EESP optimization model is built, considering the form of decision vectors, the economical/environmental optimization objectives, and various types of real-world constraints of the SIES. Based on the complexity of SIES-EESPs, a two-stage offline-to-online multiobjective optimization strategy for SIES-EESP is proposed, which transfers part of the energy dispatch online computing task to the offline high-performance computer systems. The specific constraints handling methods are designed to reduce both continuous and discrete constraints violations of SIES-EESPs. Then, an establishment method of energy scheduling scheme-base is proposed. By using the big data offline, the economical/environmental scheduling solutions of a typical year can be obtained and stored with more computing resources and operation time on land. Thereafter, a short-term multiobjective offline-to-online optimization approach by SEMS is considered, with the application of multiobjective evolutionary algorithm (MOEA) and typical schemes corresponding to the actual SIES-EESPs. Simulation results show that the proposed strategy can obtain enough feasible Pareto solutions in a shorter time and get well-distributed Pareto sets with better convergence performance, which can well adapt to the features of real-world SIES-EESPs and save plenty of operation time and storage space for the SEMS.

1. Introduction

With the increasing depletion of traditional fossil energy and the exhaust gas generated by ship combustion, the fuel consumption and emission pollution of marine diesel engine have become the main factors affecting the economy and environmental protection of ships [1–3]. In addition, due to the poor working conditions of the diesel engine, especially in the motor operating conditions, parts are easy to be damaged, and the daily maintenance cost is also the main economic factor. With the rapid development of world trade, ship transportation accounts for a considerable proportion, and CO_2 and other

exhaust gas produced by the shipping industry cannot be ignored [4]. Currently, there are two main forms of power generation on ships: traditional diesel generator and clean energy. Although the traditional diesel generator can provide strong power for ships, environmental pollution cannot be avoided. Because of its green and renewable nature, clean energy has a good development prospect in the future shipping market. Clean energy, including wind energy, solar energy, nuclear energy, fuel cells, and tidal energy, has been preliminarily applied on ships [5]. A multienergy ship can be defined as a ship with two or more kinds of energy storage equipment, energy supply unit, or energy conversion unit as the power source, and at least one of them can provide electric energy [6]. Compared with the pure electric ship, the multienergy ship has better endurance and redundancy and has the advantages of less fuel consumption, less pollution, and lower noise compared with the ship only using diesel generator set as power source [7, 8].

With the improvement of ship power grid capacity, it is necessary to improve the economy and environmental protection while meeting the reliability and safety of the ship integrated energy system (SIES). In other words, how to improve the computing performance of ship energy management system (SEMS) has become a significant issue. In the existing research, the EMS control strategy aiming at improving the safety, reliability, and fuel economy of ships mainly considers the following aspects: generator set limitation, load limit, power balance limit, power loss prevention constraint, and global condition constraint [9–11]. With the development of computer technology and the improvement of communication technology, the research on the energy management strategy of the multienergy ship is more and more in depth. At present, there are mainly fuzzy logic control strategy, baseline control strategy, optimization theory control strategy, logic gate limit control strategy (such as PID control), and modular control strategy. The essence of the fuzzy control strategy is to simulate people's thinking and to design complex tasks with simple strategies. Its main features include that the accuracy of statistical information is very inclusive; it can have a nonlinear controller with fast response speed; and when the system and external parameters change with the height fitting, it can better optimize the performance of each system energy unit and improve the operation cycle and economic performance of the system. But its disadvantage is that the fuzzy rules are based on the operator's experience, which has certain subjective factors and is prone to control distortion [12, 13].

The scheduling strategy based on optimization theory is to minimize the fuel and emissions of multienergy SEMS under certain constraints. When the cycle conditions and resistance conditions remain unchanged, a global optimization problem can be obtained, which can be solved by dynamic programming. The modular energy management strategy is a kind of management strategy which divides the complex system into several relatively simple subsystems and selects the appropriate control mode for each subsystem. Reference [14] analyzes the optimal operation of marine electric power system including all electric propulsion and energy storage system. An optimal power management method is proposed to minimize the ship operation cost, limit pollutant emissions, and improve the technical and operational level of the ship power system. Reference [15] proposes a new method to design ship power system and multiagent based power and energy management system, so as to reduce emissions and meet economic and other multiobjectives while minimizing fuel consumption and improving energy efficiency. Reference [16] proposes an energy management system for a fishing vessel power plant with a DC microgrid structure. This kind of EMS has multiobjective functions, such as reducing fuel consumption and pollution emissions and improving the economy. Reference [17] studies the energy storage control strategy of supercapacitor and battery to make the terminal voltage of supercapacitor and battery tend to be balanced and stable in a relatively short time. Reference [18] proposes the energy management system of the electrical system of luxury ships. Starting from the analysis of the current configuration of the electrical system, it puts forward some improvement schemes to reduce the fuel consumption of the diesel driven permanent magnet synchronous generator, so as to reduce the fuel consumption, reduce the pollution emission from related ships, and improve the economy. In [19], an optimal power-flow scheduling of maritime photovoltaic/battery/diesel/cold-ironing hybrid energy systems is proposed to explore solar energy sufficiently and minimize the ship electricity cost. An adaptive multi-context cooperatively coevolving particle swarm optimization (PSO) algorithm is proposed to solve the optimization problem efficiently. In [20], the optimal operation of SIES is modelled as an optimization problem subject to a number of constraints, including emission regulations of ports. Optimal control and model predictive control (MPC) methods are developed to dispatch the power flow when the ship is in port. In addition, the studies on on-land IES can be also used on SIES scheduling problem [21–23]. It can be seen that the intelligent control and optimization methods are widely used in dealing with SIES and on-land IES energy scheduling problems.

From the above works, it is not difficult to find that there are still some problems in the research and application of SIES energy scheduling.

(1) Energy scheduling of SIES is a multiobjective optimization problem, which involves the consideration of economy, environmental protection, reliability, and so on. However, most scholars focus on single-objective optimization, but a few on multiobjective scheduling. Although some factors are considered during the scheduling process, the evaluation methodology may not be reasonable. Therefore, the scheduling results cannot meet the decision-makers' preference accurately.

(2) The structure of SIES is various, which means different types of ships may have different devices, SEMS, and limits. Thus, the mathematical multiobjective optimization model may change according to different scenarios, which results in that the solution space of the optimization

problem is complex and changeable. The computation resources of SEMS alone may not meet the requirements of multiobjective energy scheduling.

(3) Comparing with on-land IES, in the actual navigation process, the route of the ship and the task of SIES is often fixed in a certain period. Therefore, the corresponding schemes may have something in common, which can be utilized to obtain knowledge based on on-land high-performance computer system and help the SEMS to get proper schemes online. How to effectively apply the historical data mining results to the calculation of SEMs online scheduling scheme is an urgent problem to be solved, and also the key to realize the green calculation of SEMS.

Motivated by the above discussions, a two-stage multiobjective optimization strategy for SIES economical/environmental scheduling problem (EESP) is proposed. The main novelty and contributions are described as follows:

(1) Considering that there are various types of constraints in real-world SIES-EESPs, some specific constraints handling methods are introduced to deal with different kinds of constraints. With the application of the constraints domination principle and proposed individual repair approach, the algorithm can move to the feasible regions faster.

(2) An establishment method of energy scheduling scheme-base is proposed. The on-land high-performance computer system is utilized to obtain feasible Pareto schemes by big data and the typical solutions are stored. In this way, more computing resources and operation time can be spent to obtain SIES-EESPs knowledge.

(3) A short-term multiobjective offline-to-online optimization approach by SEMS is considered, with the application of multiobjective evolutionary algorithm (MOEA) and typical schemes from scheme-bases. Therefore, the rational allocation of computing resources can be realized and the online computing of SIES-EESPs can be more efficient.

The remainder of this paper is organized as follows: the typical SIES-EESP optimization model is built in Section 2. Then, a two-stage multiobjective optimization strategy for SIES-EESP is proposed in Section 3. Thereafter, the case studies are given by comparing the results of the proposed method with other approaches. Finally, the conclusions of this paper are drawn in Section 5.

2. Typical SIES-EESP Optimization Model

In this section, a typical SIES is introduced including diesel generators (DGs), energy storage system (ESS), and wind turbine (WT). The mathematical models of output power are presented and the EESP optimization model is built.

2.1. Mathematical Models of Devices Output. The DGs are the main supplier of marine power. The performance of marine DGs affects the dynamic performance and safety of the ship. In this paper, the structure of diesel engine and generator is simplified, and the fuel consumption is used as the basis to estimate the economic index of the generator set, which can be described as follows:

$$C_{\text{fuel}}(t) = a \cdot p(t)^2 + b \cdot p(t) + c, \tag{1}$$

$$C_{\text{maintain}}(t) = m \cdot p(t), \tag{2}$$

$$C_{\text{start\_up}}(t) = \begin{cases} c, & u(t) = 1 \text{ and } u(t-1) = 0, \\ 0, & \text{else}, \end{cases} \tag{3}$$

$$C_{\text{DG\_total}} = \sum_{t=1}^{T} \left(C_{\text{fuel}}(t) + C_{\text{maintain}}(t) + C_{\text{start\_up}}(t) \right), \tag{4}$$

where $C_{\text{fuel}}(t)$, $C_{\text{start\_up}}(t)$, and $C_{\text{maintain}}(t)$ are the fuel cost, start-up cost, and the maintenance cost at time t, respectively; c is the cold start cost; $u(t)$ is on/off status of the DGs; and $C_{\text{DG\_total}}$ is the total cost of DG in a scheduling period T.

In equation (4), the maintenance cost and start-up cost can be calculated by the formulas below:

$$C_{\text{maintain}}(t) = P_{\text{DG}}(t)\text{OM}, \tag{5}$$

$$C_{\text{start\_up}}(t) = \sigma + \delta \left[1 - \exp\left(\frac{-T_{\text{off}}(t)}{\tau} \right) \right] (1 - u(t)), \tag{6}$$

where OM is the maintenance coefficient; σ and δ are the hot/cold start-up costs of the DG, respectively; $T_{\text{off}}(t)$ is the time the DG has been off; and τ is the cooling time constant.

The marine WT is installed on the ship. Generally, the wind turbine is installed on the upper deck of the ship to obtain the most wind energy, of which the output power can be obtained by the equation below:

$$P_{\text{WT}}(t) = \begin{cases} 0, & V(t)_t < V_{ci}, \\ a_w V(t)^3 + b_w V(t)^2 + c_w V(t) + d_w, & V_{ci} < V(t)_t < V_r, \\ P_r, & V_r < V(t)_t < V_{co}, \\ 0, & V_t > V_{co}, \end{cases} \tag{7}$$

where P_r is the rated power of WT; V_{ci} and V_{co} are the cut-in and cut-out wind speed, respectively; and V_r and $V(t)$ are the rated and actual wind speed at time t, respectively. a_w, b_w, c_w, and d_w are the parameters depending on the wind turbine types. The values of all the parameters above can be found elsewhere [24].

The ESS is applied to provide power when the other devices cannot meet the load demand and store electricity when additional power is generated, which can be expressed as [25]

$$\text{SOC}(t) = \text{SOC}(t-1) + \eta_{\text{char}} P_{\text{char}} \Delta t - \frac{1}{\eta_{\text{dischar}}} P_{\text{dischar}} \Delta t, \tag{8}$$

where $\mathrm{SOC}(t)$ is the state of charge of ESS; η_{char} and η_{dischar} are the charging and discharging efficiencies; and Δt is the time interval.

2.2. Decision Vector. The decision vectors contain the output/input power of the devices and the on/off status of DGs, which can be described as

$$\mathbf{X} = \left[P_g, U_g\right], \tag{9}$$

where P_g and U_g can be calculated by the following formulas:

$$P_g = \left[P_{\mathrm{DG}}, P_{\mathrm{ESS}}, P_{\mathrm{WT}}\right], \tag{10}$$

$$U_g = \left[u_1, u_2, \ldots, u_T\right], \quad u_k \in \{0, 1\}, \tag{11}$$

where P_{DG}, P_{ESS}, and P_{WT} are the output/input power vectors of DGs, ESS, and WT in a scheduling period, which can be described by the expressions below:

$$P_{\mathrm{DG},i} = \left[P_{\mathrm{DG},i}(1), P_{\mathrm{DG},i}(2), \ldots, P_{\mathrm{DG},i}(T)\right], \quad i = 1, 2, \ldots, N_g, \tag{12}$$

$$P_{\mathrm{ESS}} = \left[P_{\mathrm{ESS}}(1), P_{\mathrm{ESS}}(2), \ldots, P_{\mathrm{ESS}}(T)\right], \tag{13}$$

$$P_{\mathrm{WT}} = \left[P_{\mathrm{WT}}(1), P_{\mathrm{WT}}(2), \ldots, P_{\mathrm{WT}}(T)\right], \tag{14}$$

where $P_{\mathrm{DG},i}(t)$, $P_{\mathrm{ESS}}(t)$, and $P_{\mathrm{WT}}(t)$ are the output power of the i-th DG, ESS, and WT, respectively, and N_g and T are the number of DGs and the operation period of SIES, respectively.

2.3. Optimization Objectives. There are two economic evaluation indexes for ships, one is the ship income status and the other is the shipping operation cost. In this paper, the operation consumption of SIES is considered as follows:

$$\mathrm{Cost} = C_{\mathrm{DG\_total}} + C_{\mathrm{ESS\_total}} + C_{\mathrm{WT\_total}}. \tag{15}$$

The main air pollution is a diesel generator, which includes COx, Sox, and NOx. The emissions of various pollutants are directly proportional to diesel consumption. This paper uses the ship energy efficiency operation index (EEOI) to evaluate the environmental objectives. The emission value of EEOI includes the emission value during navigation and that of ships berthing in port. The emission subject includes the emission of main and auxiliary engines, boilers, and other equipment, which is also closely related to the sailing distance and cargo carrying capacity, which can be expressed as

$$\mathrm{EEOI} = \frac{A}{B}, \tag{16}$$

$$A = \sum_{t=1}^{T} \left(\left(c_2 r_g(t)^2 + c_1 r_g(t) + c_0\right)\Delta t\right), \tag{17}$$

$$B = \sum_{t_p \in T_p} \left(m_{\mathrm{AES}}^{t_p} \mathrm{Dist}^{t_p}\right), \tag{18}$$

where t_p is the time of berthing, $m_{\mathrm{AES}}^{t_p}$ is the cargo load of ships, Dist is the sailing distance; $r_g(t)$ is the standard unitary value of the active contribution.

2.4. Typical Constraints

2.4.1. Load Balance Constraints. The power output by the devices of SIES should meet the load demand, which can be described as

$$\sum_{i=1}^{nG} P_{\mathrm{DG},i}(t)\eta_g + P_{\mathrm{ESS}}(t) + P_{\mathrm{WT}}(t) = \frac{P_{lp}(t)}{\eta_p} + P_{ls}(t), \tag{19}$$

where η_g and η_p are the efficiency value of DGs and propulsion, respectively. P_{lp}^t and P_{ls}^t are the load of propulsion and service, respectively.

2.4.2. Rated Power Constraints. The output of DGs and WT should be lower than the rated power, which can be expressed as

$$r_g(t) \le 1.0 \tag{20}$$

2.4.3. SOC Constraints. The ESS can neither be overcharged nor overused, so the SOC should follow constraint as follows:

$$\mathrm{SOC}_{\min} \le \mathrm{SOC}_t \le \mathrm{SOC}_{\max}. \tag{21}$$

2.4.4. Ramp Rate Constraints. The ramp rates of DGs outputs cannot be too large, which can be expressed as

$$\left|P_{\mathrm{DG},i}(t) - P_{\mathrm{DG},i}(t-1)\right| \le P_{\mathrm{DG},i,\mathrm{ramp}}. \tag{22}$$

2.4.5. Minimum On/Off Time Limits. There are minimum on/off time limits for the DGs, which can be expressed as

$$\left(T_{\mathrm{on},i,t-1} - \mathrm{MUT}_i\right)\left(u_i(t-1) - u_i(t)\right) \ge 0, \tag{23}$$

$$\left(T_{\mathrm{off},i,t-1} - \mathrm{MDT}_i\right)\left(u_i(t) - u_i(t-1)\right) \ge 0, \tag{24}$$

where T_{off} and T_{on} are the actual on/off time of DGs and MUT and MDT are the minimum on/off time for the DGs.

3. Two-Stage Multiobjective Optimization Strategy for SIES-EESP

It can be seen from Section 2 that the SIES contains various types of energy sources and may cost much computation resource when dealing with EESPs by EMS. Therefore, considering the features of SIES discussed in Section 1, a two-stage multiobjective optimization with specific

constraints handling methods is proposed in this paper to solve the SIES-EESPs.

3.1. Specific Constraints Handling Methods. According to Section 2, there are all kinds of constraints related to SIES-EESPs. Some of the constraints are simple but some are not. Therefore, a general constraints handling method may not be effective in finding the feasible regions. In this paper, a comprehensive constraints handling framework is proposed considering specific types of constraints.

3.1.1. Continuous Constraints Handling Strategy. The continuous constraints involved in SIES-EESP are similar to those in other scheduling problems on land, which can be solved effectively by the constraints domination principle (CDP). CDP designed by Deb [26] is an efficient constraint handling approach, which uses the feasibility and the overall constraints violation of solutions to decide the domination levels. To compare the actual engineering constraints fairly, different types of constraints are normalized and summed up first in this paper, which can be expressed as follows:

$$g_{k,\text{normal}}(\mathbf{x}) = \frac{g_k(\mathbf{x}) - g_{k,\min}}{g_{k,\max} - g_{k,\min}}, \tag{25}$$

$$G(\mathbf{x}) = \sum_{k=1}^{k} w_k g_{k,\text{normal}}(\mathbf{x}), \tag{26}$$

where $g_{k,\text{normal}}$ and g_k are the normalized violation value and actual value of the k-th type of constraint, respectively; $g_{k,\min}$ and $g_{k,\max}$ are the minimum and maximum violation value of the k-th type of constraint in the population, respectively; and G is the overall constraints violation value.

3.1.2. Discrete Constraints Handling Strategy. It can be seen from equation (11) that there are discrete variables (0/1) in the decision variable vector, which represent the on/off statuses at those time points. In the MOEA process, these discrete variables cannot be optimized gradually during evolution, and it is always difficult to measure the gap between the infeasible discrete variable vector and the feasible region. Therefore, in a practical SIES-EESP, it may not be efficient to handle the constraints related to the discrete variable vector (such as the on/off time limits).

To avoid the violations of this constraint when using MOEA, this paper proposes an individual repair approach (IRA) based on variables separation and recombination.

The repair process of the decision vector is actually to modify the infeasible subvector u_i, which can be described as the following steps:

(i) Step 1: For the status variable vector $\mathbf{u}_i = [u_i(1), u_i(2), \ldots, u_i(T)]$ of the i-th DG, decompose it into several subvectors, where there are only ones or zeros as elements. Separate the subvectors into two sets by the elements, in which Set U only has subvectors composed of ones and Set D only has subvectors composed of zeros. Then, the numbers of the subvector dimensions will be composed to a new vector in each set, which can be described as $x_U = [x_{U,1}, x_{U,2}, \ldots, x_{U,u}]$ and $x_D = [x_{D,1}, x_{D,2}, \ldots, x_{D,d}]$, respectively. Then, the elements in x_U and x_D will be recombined into a new vector $\mathbf{x}_{\text{UD}}$, according to the order of elements $\mathbf{u}_i$.

(ii) Step 2: If $u_k(1) = 1$, go to Step 3; if not, move to Step 4.

(iii) Step 3: From the first nonzero element of $\mathbf{x}_{\text{UD}}$, check all of the nonzero elements from left to right. For the nonzero elements in the odd positions, identify whether they meet the minimum on-time limits, while for those in the even positions, determine whether they meet the minimum off-time limits. Suppose that there is a nonzero element in the odd position (let us say $x_{\text{UD},j}$) which does not meet the constraint. In another word, if $\text{MUT}_k - x_{\text{UD},j} = \triangle U_j > 0$, the modification procedure starts, of which the pseudocode is presented in Algorithm 1. After the modification procedures, the new vector $\mathbf{x}_{\text{UD}}'$ is obtained. Then, go to Step 5.

(iv) Step 4: This step is similar to Step 3. The only difference is that for the nonzero elements in the odd positions, identify whether they meet the minimum off-time limits, and for those in the even positions, determine whether they meet the minimum on-time limits. Algorithm 1 still applies to the violations. Then, the repair process moves to Step 5.

(v) Step 5: remove the zero elements of vector $\mathbf{x}_{\text{UD}}'$ and keep the nonzero elements in the original order, which form a new vector $\mathbf{x}_{\text{UD,repair}}$. The elements of vector $\mathbf{x}_{\text{UD,repair}}$ represent the amount of the on/off hours. The on and off statuses of the DGs occur alternately, and whether the first status is on or off is related to value of $u_i(1)$.

(vi) Step 6: establish the feasible status variable vector u_i' based on $\mathbf{x}_{\text{UD}}'$ and $u_i(1)$.

It is obvious from the repair process that the relationship between the status variable vector $\mathbf{u}_i$ generated randomly in the T-dimensional solution space Q_T and the obtained feasible status variable vector u_i' is a many-to-one mapping, which means that, by this proposed method any infeasible status variable vectors can be transferred into a certain vector in the feasible region, and no feasible vectors would be lost. To guarantee the effectiveness of the optimization process, the replacement ratio is set 15% according to [27].

3.2. Offline-to-Online Multiobjective Optimization Strategy. The specific constraints handling methods can handle different types of constraints during the evolutionary process on actual SIES-EESPs, which can help MOEA to find feasible solutions efficiently. However, it is evident that the constraints handling process may require excessive computational resources and reduce the efficiency of EMS in dealing with energy dispatch problems. Therefore, an offline-to-online multiobjective optimization strategy is

```
Input: x_{UD,j}
Output: x'_{UD,j}
x'_{UD,j} = x_{UD,j};
for m ← 1 to V − j/2 do
n = 2m − 1;
if x_{UD,j+n} > △U_j
x'_{UD,j} = MUT_k;
x'_{UD,j+n} = x_{UD,j+n} − △U_j;
break;
else
x'_{UD,j} = x'_{UD,j} + x_{UD,j+n} + x_{UD,j+n+1};
x'_{UD,j+n} = 0;
x'_{UD,j+n+1} = 0;
if x'_{UD,j} ≥ △U_j
break;
end
end
end
if x'_{UD,j} < MUT_k
for p ← 1 to V − j do
x'_{UD,j} = x'_{UD,j} + x_{UD,j+p};
x_{UD,j+p} = 0;
end
end
```

Algorithm 1: Modification procedure.

proposed to transfer some of the computing tasks from SEMS to offline computers on land.

3.2.1. Establishment of Energy Scheduling Scheme-Base Using Big Data. Since the courses and tasks of inland cargo ships are similar in the same months of a year, of which the scheduling schemes may not have much difference, it is possible to use the historical experiences to guide the day-ahead economical/environmental scheduling optimization. By using the big data of the typical curves of load demands, the wind speed of specific regions can be fitted out and the applications of all kinds of energy resources can be derived. On the other hand, the establishment of the energy scheduling scheme-base using big data can be realized by a high-performance computer system with more computing resources on land [28, 29]. Therefore, the scheme-base for a specific inland cargo ship can be obtained including all the economical/environmental scheduling solutions of a typical year by big data. Each scheduling task is formed as a SIES-EESP with fitted curves of load and wind speed. And the MOEA is applied with the proposed specific constraints handling methods to search for the typical Pareto schemes. Since there are no time and computing resources limits, this process can be realized on the high-performance computer system and stored in the mass storage devices on land.

The typical Pareto schemes would be stored in scheme-base for SIES-EESP, which contains two subbases, namely, Scheme-base A and Scheme-base B, which store the feasible Pareto SIES-EESP schemes meeting all the constraints and only the discrete constraints, respectively. By introducing the high-performance computers on land with large memory space and unlimited operation time, the obtained Pareto solutions can be more close to the true Pareto frontiers and the typical schemes can ensure the low cost and emission under various types of working conditions of a year.

3.2.2. Short-Term Multiobjective Offline-to-Online Optimization by SEMS. Although the economical/environmental scheduling solutions in the scheme-base can be applied in typical working conditions, it is clear that, in practice, the load may change according to the actual requirement, and the environment can also be different so that the output of renewable energy sources would not follow the typical schemes. Therefore, the short-term dispatch scheme is still needed by the SEMS based on its own conditions. To reduce the operation time of SEMS when dealing with SIES-EESPs, a novel multiobjective offline-to-online optimization strategy is proposed based on the energy scheduling scheme-base. The procedure can be described as follows:

(i)Step 1: Select the Pareto energy scheduling scheme sets of the typical SIES-EESP which has the highest fitting degree with the short-term forecast curves of load demand and wind speed.

(ii)Step 2: Choose a kind of MOEA as the optimization tool and use CDP and IRA as the constraints handling methods.

(iii)Step 3: Use the typical solutions in Scheme-base A to form the initial population.

(iv)Step 4: Run MOEA to find the Pareto solutions of the actual SIES-EESP. During the evolutionary process, check the proportion α of the infeasible solutions. If $\alpha > \alpha_0$, move to Step 5; otherwise, go to Step 6.

(v)Step 5: Replace the infeasible solutions with the typical solutions in Scheme-base B in the current population, so that the algorithm can move to the feasible regions corresponding to the discrete constraints efficiently.

(vi)Step 6: Output the Pareto solutions.

It can be seen from the above that comparing with the traditional optimum dispatching methods, which utilize the forecast data and solve the SIES-EESPs in one operation, the proposed two-stage offline-to-online multiobjective optimization strategy can utilize the big data to find the knowledge of the SIES-EESPs by offline computers on land so that the reference typical solutions are derived and stored. When a new actual SIES-EESP needs to be solved, the SEMS can find the Pareto solutions with the help of the most relevant reference schemes from the solution-base. In this way, the online computing resources can be saved and the computing efficiency is improved.

The flowchart of the short-term multiobjective offline-to-online optimization by SEMS is shown in Figure 1.

4. Simulation Results and Discussions

4.1. Data and Parameter Settings. In this section, two actual SIES-EESP cases are introduced considering different combinations of energy sources. In Case 1, there are only two DGs and one ESS, and, in Case 2, a WT is introduced as the renewable energy source. The parameter settings of ESS and WT can be found in Table 1. Other parameter settings can be found in [24, 30]. The fitted curves by big data and the actual curves of the load demand and wind speed are shown in Figures 2(a) and 2(b).

Besides, in this paper, the nondominated sorting genetic algorithm (NSGA-II) is utilized as the core optimization algorithm, which is an efficient MOEA in solving all kinds of multiobjective optimization problems. The parameter settings can be found in [26]. The maximum generation is set as 100000 when searching for the typical Pareto schemes. In addition, to study the performance of the proposed strategy, four methods are utilized. Method 1 and Method 2 use the proposed offline-to-online optimization strategy, of which the difference is that only Method 1 uses the specific constraints handling methods to deal with the constraints while Method 2 only uses CDP. Method 3 is the original NSGAII with CDP, but in Method 4, the punishment function method is introduced, which can be found in [25]. The maximum generation is 2000 for Methods 1 and 2 and 10000 for Methods 3 and Method 4, respectively.

4.2. Case Studies. The average values of best cost and emission for the two cases by the four methods can be found in Table 2. It can be seen from Table 2 that the results by Methods 1 and 2 are similar in the two cases, and the results in Case 2 are better, which indicates that by introducing the wind turbine, renewable energy can be made full use of and the cost and pollutions can be reduced. The results also show that, by utilizing the typical solutions in the scheme-base as reference individuals, the algorithm can find better solutions so that the distribution of the Pareto frontiers can be guaranteed. On the contrary, if NSGAII is used directly without any prior knowledge of the related SIES-EESP, the algorithm may trap in local optimum and cannot find satisfying nondominate solutions for the decision-makers. The results of Methods 3 and 4 also show that the CDP method is better than PFM in dealing with SIES-EESP cases.

To further study the performance of the proposed two-stage multiobjective optimization strategy, the population of the last generation, the feasible solutions, and the Pareto frontiers obtained by the four methods are presented in Figures 3–5. Also, the average numbers of feasible solutions are shown in Table 3.

It can be seen from Figures 3(a) and 5 that almost all of the solutions by Method 1 are nondominate comparing with those by other methods. Besides, nearly 48 solutions are feasible ones, and the solutions distribute evenly on the Pareto frontier, which can provide more trade-off schemes for the decision-makers. As for Method 2, most of the solutions are dominated by those found by Method 1 but are much better than those using Methods 3 and 4. The numbers of obtained feasible solutions are also lower than those by Method 1, most of which are located in a smaller region. The results by Methods 3 and 4 are the worst in Cases 1 and 2, which have higher cost and emission values comparing with those by Methods 1 and 2. It can be seen from Figure 3 and that both Methods 3 and 4 trap in local optimum and all the solutions obtained are dominated by other methods. Moreover, according to Figures 4(a) and 4(b), only about 28 and 15 solutions in the two cases are feasible for Method 3, and about 22 and 10 feasible ones are obtained by Method 4. Based on the results in Figure 5, no more than 6 solutions are nondominate in the population for the last two methods, which means they cannot provide enough trade-off schemes for the decision-makers.

The results in Cases 1 and 2 imply that, with the applications of energy scheduling scheme-base, the algorithm can jump out of the infeasible regions and find the feasible regions efficiently. Therefore, more computing resources can be utilized to search for the Pareto solutions, so that the convergence and diversity of the population can be guaranteed. On the other hand, the introduction of specific constraints handling methods, especially the proposed IRA for discrete constraints, helps the algorithm to deal with all kinds of constraints effectively, so more feasible Pareto solutions can be obtained. In addition, the results by Methods 3 and 4 indicate that CDP is efficient in dealing with common constraints and is necessary for the short-term multiobjective offline-to-online optimization.

To evaluate the Pareto frontiers obtained by the four methods quantitatively, the hypervolume method is introduced. The evaluation method based on hypervolume value was proposed by Zitzler et al. [31], which utilizes the volume of hypercube surrounded by the Pareto solutions and reference point in the search space to evaluate the quality of Pareto frontiers obtained. The average values and standard deviations of hypervolume values by the four methods are shown in Table 4. It can be seen from

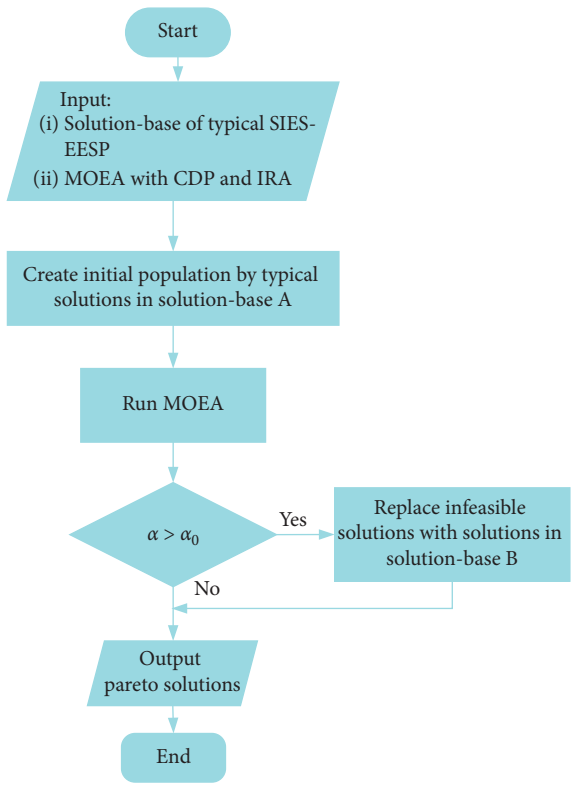


Figure 1: Flowchart of short-term multiobjective offline-to-online optimization by SEMS.

Table 1: Parameter settings of ESS.

Initial cost (m.u./kW)	567
Replacement cost (m.u./kW)	467.9
Maintenance cost (kW·h·y)	2.67
Charging efficiency	0.86
Discharge efficiency	0.86
SOC_{max}	0.9
SOC_{min}	0.3
Initial SOC	0.5
Service life (yr)	2.5

Table 4 that Method 1 obtains the highest hypervolume values in both cases (0.7051 and 0.8014). The hypervolume values by Method 2 are lower but the difference is small. However, Method 3 only gets 0.1648 and 0.0737 on hypervolume values, which are far lower than the other two. This demonstrates that if the original NSGAII is utilized to solve the actual SIES-EESP, it may not get satisfied Pareto solutions with limited computing resources and operation time, so the goals of cost and emission reduction cannot be achieved. The hypervolume values by Method 4 are the lowest, which means the commonly used punishment function methods are not suitable for actual SIES-EESPs. However, with the utilization of energy scheduling scheme-base and proposed constraints handling method, the feasible regions can be found efficiently and the Pareto sets can be obtained with better convergence, spread, and distribution by less computing resources and operation time.

As for the standard deviations, it can be seen from Table 4 that Method 1 gets the lowest levels, which means that the proposed strategy is more reliable in searching for Pareto solutions on the SIES-EESPs. This is necessary in the real world since the online dispatching requires the SEMS to obtain the Pareto schemes in a short time, which does not allow the algorithm to run many times. Thus, a reliable optimization tool is needed to ensure a reliable/economical/environmental dispatching scheme.

The average CPU time by the four methods is shown in Table 5. Since Methods 3 and 4 take iterative 10000 times, the CPU time is much larger than the other two. According to the results above, it is evident that, by using the typical solutions from the energy scheduling scheme-base, the SEMS can get more feasible and nondominate solutions within less operation time, which can save more computing resources. In addition, it can be seen that Method 1 takes a little more time than Method 2 because of the introduction of specific constraints handling

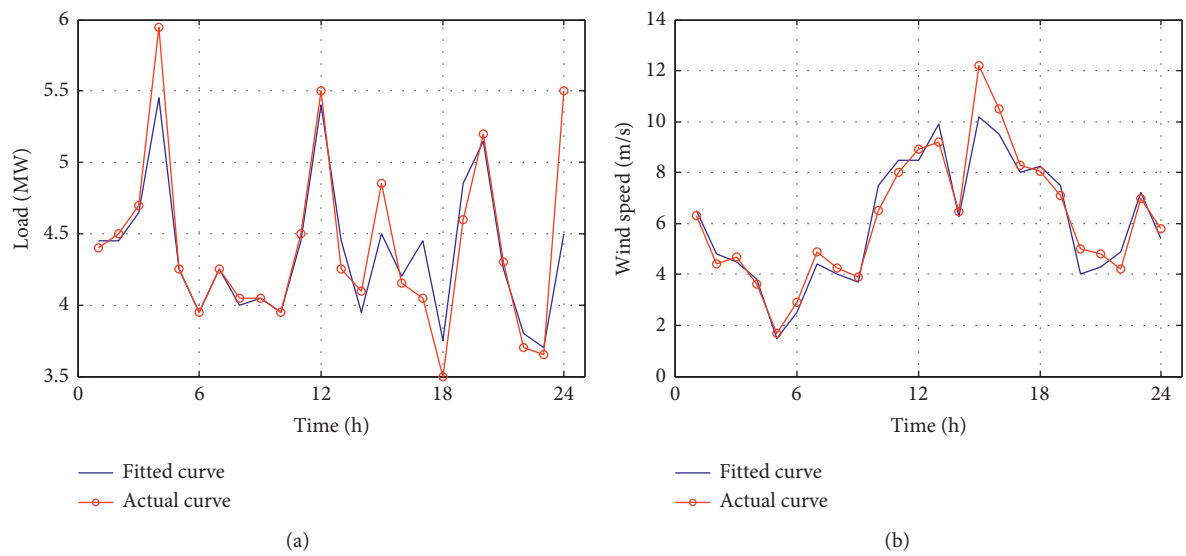


Figure 2: Fitted curves by big data and the actual curves of the load demand and wind speed. (a) Load demand. (b) Wind speed.

Table 2: Average values of best cost and emission by the four methods.

Cases	Method 1		Method 2		Method 3		Method 4	
	Cost (m.u.)	Emission (10^4 u.CO_2)	Cost (m.u.)	Emission (10^4 u.CO_2)	Cost (m.u.)	Emission (10^4 u.CO_2)	Cost (m.u.)	Emission (10^4 u.CO_2)
Case 1	7.8254	4.4166	7.9610	4.4621	8.6542	4.8458	8.6721	4.8593
Case 2	7.0542	4.2985	7.2544	4.4203	7.8461	4.5319	7.8836	4.5546

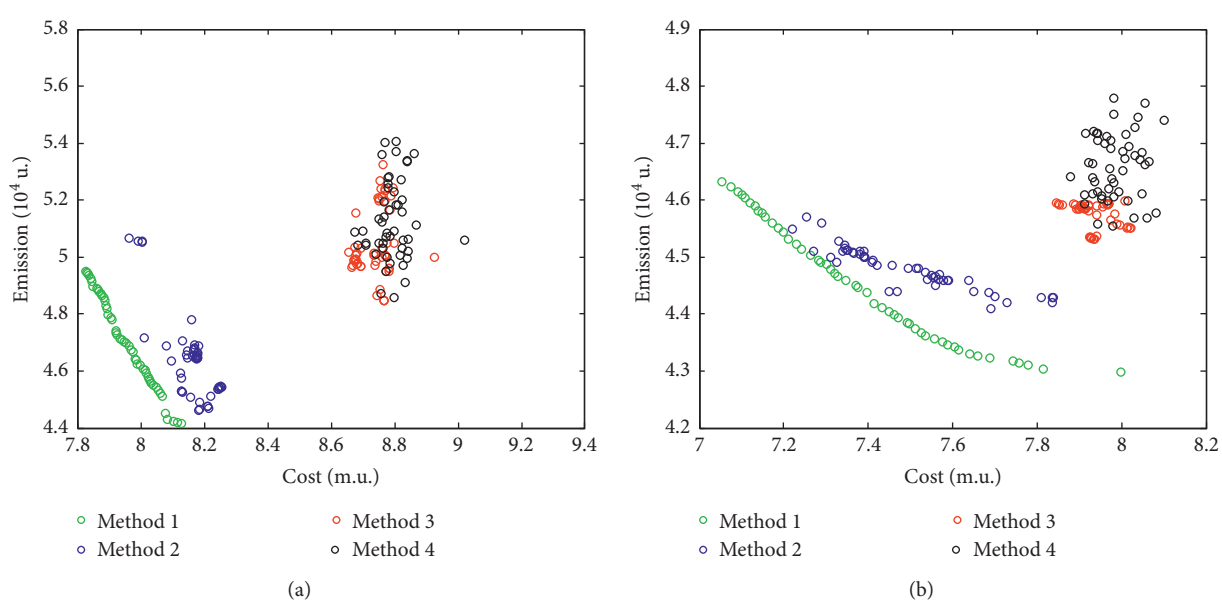


Figure 3: Populations of the last generation by the four methods. (a) Case 1. (b) Case 2.

methods, but the difference is not large (only about 5 s–7 s). On the other hand, the number of feasible solutions and the hypervolume values by Method 1 are better based on the above results. All in all, the proposed Method 1 is a proper optimization tool for the SEMS in dealing with the SIES-EESPs.

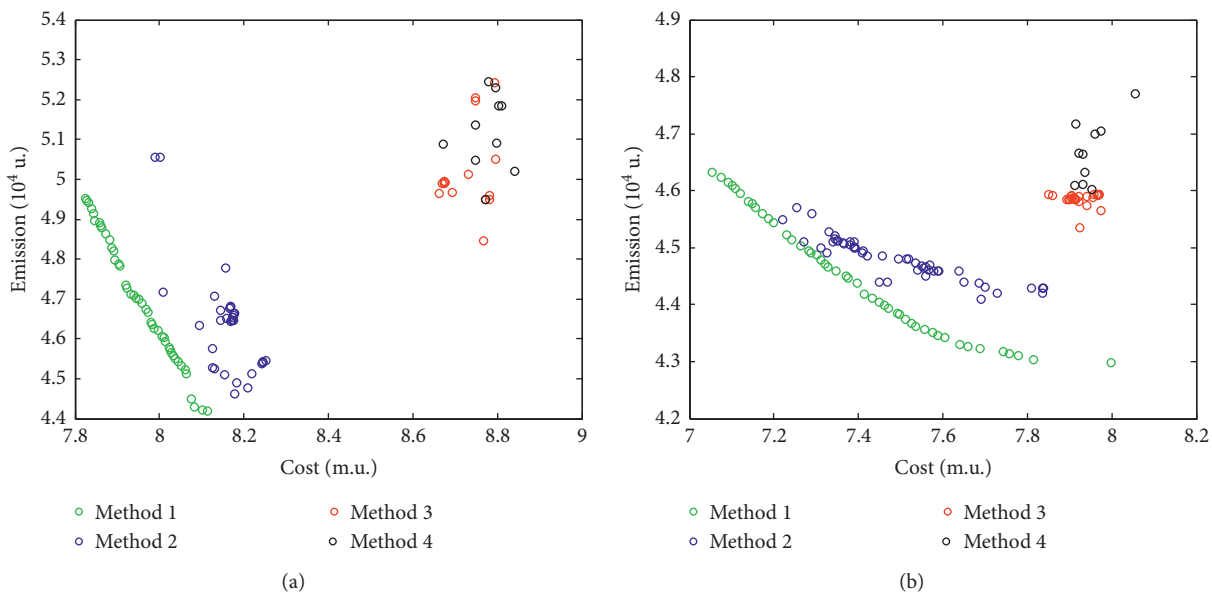


Figure 4: Feasible solutions by the four methods. (a) Case 1. (b) Case 2.

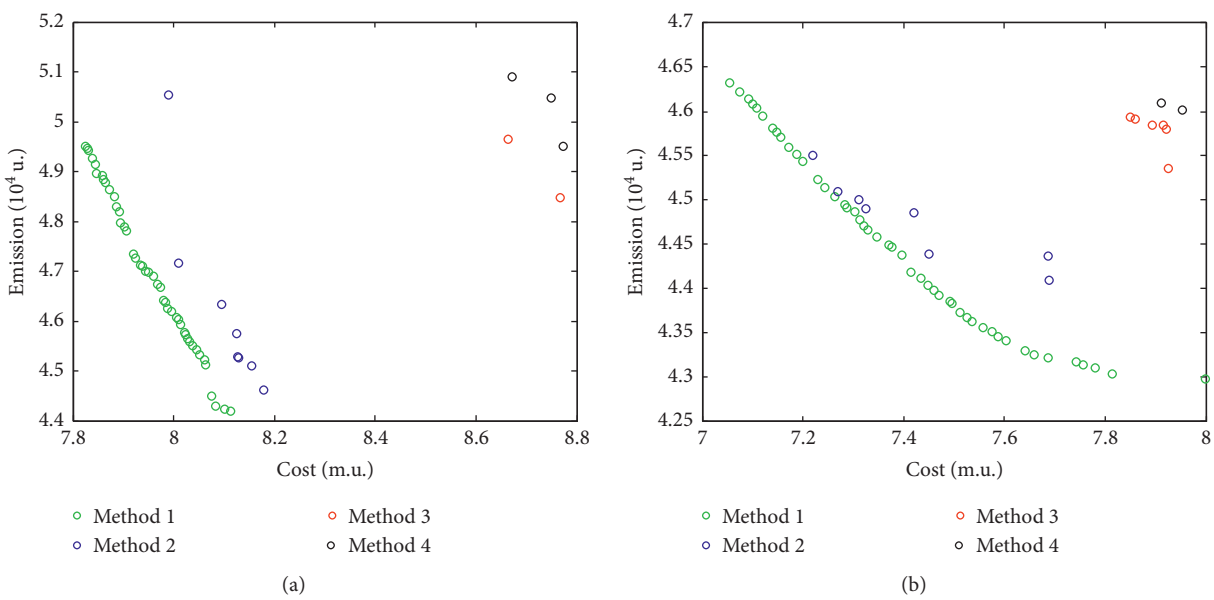


Figure 5: Pareto frontiers obtained by the four methods. (a) Case 1. (b) Case 2.

Table 3: Average number of feasible solutions by the four methods.

Cases	Method 1	Method 2	Method 3	Method 4
Case 1	47.8	42.0	28.2	21.3
Case 2	45.3	32.5	15.1	9.8

Table 4: Average values and standard deviations of hypervolume values by the four methods.

Cases	Method 1		Method 2		Method 3		Method 4	
	H-values	Standard deviations	H-values	Standard deviations	H-values	Standard deviations	H-values	Standard deviations
Case 1	0.7051	$5.2314e-03$	0.6241	$9.9878e-03$	0.1648	$3.2148e-02$	0.1135	$7.2615e-02$
Case 2	0.8014	$7.2457e-03$	0.6987	$1.5414e-02$	0.0737	$7.0325e-02$	0.0341	$7.9984e-02$

Table 5: Average CPU time(s) by the four methods.

Cases	Method 1	Method 2	Method 3	Method 4
Case 1	36.9827	31.2548	250.2489	312.5132
Case 2	41.0065	34.0124	259.3257	333.2147

5. Conclusions

This paper proposes a two-stage offline-to-online multiobjective optimization strategy for SIES-EESP, by which the SIES-EESP schemes are derived by using the big data offline to form the prior knowledge and the MOEAs to search for feasible Pareto solutions online. Before the application of intelligent computing, the hybrid constraints handling strategies are designed considering both continuous constraints and discrete constraints. Then, the energy scheduling scheme-base is established using big data, so that the typical SIES-EESP schemes can be obtained and stored offline by high-performance computer systems. On the other hand, the typical Pareto solutions in the scheme-base are applied to the actual SIES-EESPs online corresponding to similar load demands and environment conditions, which are used as reference vectors to guide the algorithm to converge to the feasible regions. The results show that the proposed method can obtain enough feasible solutions and get well-distributed Pareto sets with better convergence performance. Moreover, the operation time can be reduced evidently with less computing resources, comparing with the current optimization studies. By the two-stage multiobjective optimization strategy, the rational allocation of computing resources and the advantage of SIES big data and high-performance computer systems on land can be realized. Meanwhile, the EMS only needs to select the most relevant reference schemes and uses them to find the Pareto solutions on actual SIES-EESPs. In this way, part of the computing task is transferred to the offline computers on land, so that the online computing resources can be saved. It should be noted that this paper only considers a typical SIES-EESP optimization model and divide the constraints into two types (general ones and special ones) to establish the scheme-bases. Further studies are needed on the following aspects:

(1) More SIES-EESP optimization models can be studied based on the needs in the real world so that the adaptability of the proposed method can be tested.

(2) More types of constraints can be derived to describe the SIES-EESP more accurately, and the scheme-bases can be established based on more kinds of rules. In this way, the selected typical solutions during the online computing process by SEMS can help MOEA to reach feasible regions more efficiently.

(3) More kinds of MOEAs can be utilized in the proposed optimization strategy so that a proper MOEA can be selected when dealing with a specific actual SIES-EESP.

Data Availability

The prior studies and data are cited at relevant places within the text as references [24–26, 30].

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 (Grants nos. 51909199 and 51709215), the Green Intelligent Inland Ship Innovation Programme, by projects from Key Lab of Marine Power Engineering and Tech. Authorized by MOT (KLMPET2019-03 and KLMPET2019-02), and the Opening Foundation of Key Laboratory of Information Security of Zhejiang Province (Grant no. KF201912).

References

[1] W. Zhou and X. Xiaohua, "Tariff-driven demand side management of green ship," *Solar Energy*, vol. 170, pp. 991–1000, 2018.

[2] L. Bilgili and U. B. Celebi, "Developing a new green ship approach for flue gas emission estimation of bulk carriers," *Measurement*, vol. 120, pp. 121–127, 2018.

[3] J. Lampe, E. Rüde, Y. Papadopoulos, and S. Kabir, "Model-based assessment of energy-efficiency, dependability, and cost-effectiveness of waste heat recovery systems onboard ship," *Ocean Engineering*, vol. 157, pp. 234–250, 2018.

[4] Y. Hongsheng, W. Liyang, and Y. Jianxing, "The environmental impact analysis of hazardous materials and the development of green technology in the shipbreaking process," *Ocean Engineering*, vol. 161, pp. 187–194, 2018.

[5] Z. Li, Y. Xu, L. Wu, and X. Zheng, "A risk-averse adaptively stochastic method for multi-energy ship operation under diverse uncertainties," *IEEE Transactions on Power Systems*, 2020.

[6] Z. Li, Y. Xu, S. Fang et al., "Robust coordination of A hybrid AC/DC multi-energy ship microgrid with flexible voyage and thermal loads," *IEEE Transactions on Smart Grid*, vol. 99, p. 1, 2020.

[7] Z. Li, Y. Xu, S. Fang, Y. Wang, and X. Zheng, "Multiobjective coordinated energy dispatch and voyage scheduling for a multienergy ship microgrid," *IEEE Transactions on Industry Applications*, vol. 56, no. 2, pp. 989–999, 2020.

[8] C. Zhang, D. Zhang, M. Zhang et al., "Data-driven ship energy efficiency analysis and optimization model for route planning in ice-covered arctic waters," *Ocean Engineering*, vol. 186, Article ID 106071, 2019.

[9] E. H. Trinklein, G. G. Parker, and T. J. Mccoy, "Modeling, optimization, and control of ship energy systems using exergy methods," *Energy*, vol. 191, pp. 1–8, 2020.

[10] S. Fang and Y. Xu, "Multi-objective robust energy management for all-electric shipboard microgrid under uncertain wind and wave," *International Journal of Electrical Power and Energy Systems*, vol. 117, pp. 1–11, 2020.

[11] N. Vahabzad, M. Jadidbonab, B. Mohammadiivatloo et al., "Energy management strategy for a short-route hybrid cruise ship: an IGDT-based approach," *IET Renewable Power Generation*, vol. 14, no. 10, 2020.

[12] Q. Zhang, Z. Ding, and M. Zhang, "Adaptive self-regulation PID control of course-keeping for ships," *Polish Maritime Research*, vol. 27, no. 1, pp. 39–45, 2020.

[13] A. Ouroua, L. Domaschk, and J. H. Beno, "Electric ship power system integration analyses through modeling and

simulation," in *Proceedings of the Electric Ship Technologies Symposium*, IEEE, Paris, France, October 2005.

[14] F. D. Kanellos, "Optimal power management with GHG emissions limitation in all-electric ship power systems comprising energy storage systems," *IEEE Transactions on Power Systems*, vol. 29, no. 1, pp. 330–339, 2013.

[15] D. Tang, X. Yan, Y. Yuan et al., "Multi-agent based power and energy management system for hybrid ships," in *Proceedings of the 2015 International Conference on Renewable Energy Research and Applications (ICRERA)*, pp. 383–387, IEEE, Palermo, Italy, November 2015.

[16] A. Accetta and M. Pucci, "A first approach for the energy management system in DC micro-grids with integrated RES of smart ships," in *Proceedings of the 2017 IEEE Energy Conversion Congress and Exposition (ECCE)*, pp. 550–557, IEEE, Cincinnati, OH, USA, October 2017.

[17] Z. Jingnan and Z. Ying, "Control strategy of hybrid energy storage system in ship electric propulsion," in *Proceedings of the 2018 IEEE International Conference on Mechatronics and Automation (ICMA)*, pp. 1026–1030, IEEE, Changchun, China, August 2018.

[18] A. Accetta and M. Pucci, "Energy management system in DC micro-grids of smart ships: main gen-set fuel consumption minimization and fault compensation," *IEEE Transactions on Industry Applications*, vol. 55, no. 3, pp. 3097–3113, 2019.

[19] R. Tang, X. Li, J. Lai et al., "A novel optimal energy-management strategy for a maritime hybrid energy system based on large-scale global optimization," *Applied Energy*, vol. 228, pp. 254–264, 2018.

[20] R. Tang, Z. Wu, X. Li et al., "Optimal operation of photovoltaic/battery/diesel/cold-ironing hybrid energy system for maritime application," *Energy*, vol. 162, pp. 697–714, 2018.

[21] L. Jinglu, W. Anna, Q. Yanhua et al., "Coordinated operation of multi-integrated energy system based on linear weighted sum and grasshopper optimization algorithm," *IEEE Access*, vol. 6, p. 1, 2018.

[22] A. Chaouachi, R. M. Kamel, R. Andoulsi, and K. Nagasaka, "Multiobjective intelligent energy management for a microgrid," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 4, pp. 1688–1699, 2013.

[23] A. B. Ani, H. A. Ebrahim, and M. J. Azarhoosh, "Simulation and multi-objective optimization of a trickle-bed reactor for diesel hydrotreating by a heterogeneous model using non-dominated sorting genetic algorithm II," *Energy and Fuels*, vol. 29, pp. 3041–3051, 2015.

[24] X. Li and Y. Fang, "Dynamic environmental/economic scheduling for microgrid using improved MOEA/D-M2M," *Mathematical Problems in Engineering*, vol. 2016, Article ID 2167153, 14 pages, 2016.

[25] X. Li, J. Lai, and R. Tang, "A hybrid constraints handling strategy for multiconstrained multiobjective optimization problem of microgrid economical/environmental dispatch," *Complexity*, vol. 2017, Article ID 6249432, 12 pages, 2017.

[26] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[27] L. Y. Tseng and C. Chen, "Multiple trajectory search for single objective constrained real-parameter optimization problems," in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 1–7, IEEE, Barcelona, Spain, July 2010.

[28] J. Lai, X. Lu, X. Yu, and A. Monti, "Stochastic distributed secondary control for ac microgrids via event-triggered communication," *IEEE Transactions on Smart Grid*, vol. 11, no. 4, pp. 2746–2759, 2020.

[29] J. Lai, X. Lu, X. Yu, and A. Monti, "Cluster-oriented distributed cooperative control for multiple ac microgrids," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 11, pp. 5906–5918, 2019.

[30] X. Chen, Q. Wei, and X. Li, *Research on Multiobjective Optimization Strategy of Economic/Environmental Energy Management for Multi-Energy Ship Based on MOEA/D, Bio-Inspired Computing: Theories and Applications*, pp. 135–146, 2020.

[31] E. Zitzler and L. Thiele, "Multiobjective evolutionary algorithms: a comparative case study and the strength Pareto approach," *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 4, pp. 257–271, 1999.

Hindawi
Complexity
Volume 2021, Article ID 6646881, 18 pages
https://doi.org/10.1155/2021/6646881

Research Article

Performance Optimization of Cloud Data Centers with a Dynamic Energy-Efficient Resource Management Scheme

Yu Cui,[1,2] Shunfu Jin,[1] Wuyi Yue,[3] and Yutaka Takahashi[4]

[1]*School of Information Science and Engineering, Yanshan University, Qinhuangdao 066004, China*
[2]*College of Mathematics and Information Science & Technology, Hebei Normal University of Science & Technology, Qinhuangdao 066004, China*
[3]*Department of Intelligence and Informatics, Konan University, Kobe 658-8501, Japan*
[4]*Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan*

Correspondence should be addressed to Shunfu Jin; jsf@ysu.edu.cn

Received 24 December 2020; Revised 18 January 2021; Accepted 21 January 2021; Published 13 February 2021

Academic Editor: Yongsheng Hao

As an advanced network calculation mode, cloud computing is becoming more and more popular. However, with the proliferation of large data centers hosting cloud applications, the growth of energy consumption has been explosive. Surveys show that a remarkable part of the large energy consumed in data center results from over-provisioning of the network resource to meet requests during peak demand times. In this paper, we propose a solution to this problem by constructing a dynamic energy-efficient resource management scheme. As a way of saving energy as well as maintaining cloud user's quality of experience, the scheme presents a multitier cloud architecture by configuring physical machines (PMs) into two pools: a hot (running) pool and a warm (turned on, but in dynamic sleep) pool. Each PM is configured with a resource search engine (RSE) that finds an available virtual machine (VM) for the request, and a synchronous sleep mechanism is introduced to the warm pool. To analyze the end-to-end performance of the cloud system's service with the proposed scheme, we establish a hybrid queueing system composed of three stochastic submodels by using a matrix-geometric solution. Accordingly, the average latency of requests and the energy-saving rate of the system are derived. Through numerical results, we show the influence of the synchronous sleep mechanism on the system performance. Moreover, from the perspective of economics, we build a system cost function to study the trade-off between different performance measures. An improved Salp Swarm Algorithm (SSA) is presented to minimize the system cost and optimize the sleep parameter.

1. Introduction

As a direct result of the rapid growth in the number of cloud users, some cloud providers have already built large numbers of data centers to satisfy the resources demands [1]. The consequences are massive increases in energy consumption, an excessive increase in carbon emissions, and a reduction in benefits for the cloud providers [2]. Statistical results show that the average data center can consume as much energy as 25,000 ordinary households [3]. Therefore, based on the concept of green computing, obviously, the development of a greener, more energy-efficient resource management mechanism for cloud systems is becoming more desirable [4, 5].

The main contributions of this paper are summarized as follows:

(i) We present a cloud architecture composed of a task-scheduling decision layer, a resource-provisioning layer, and an actual service layer. Over the multitier cloud architecture, we propose an energy-efficient resource management scheme with a synchronous sleep mechanism.

(ii) We establish a queueing model composed of three subqueues to capture the proposed scheme. By using a Markov chain-based approach, we derive two performance measures: the average latency of requests and the energy-saving rate of the system.

(iii) Taking into account the trade-off between the average latency of requests and the energy-saving rate of the system, we build a system cost function and present an improved Salp Swarm Algorithm (SSA) to optimize the sleep mechanism.

2. Related Work

In this section, we review the related work on energy conservation research in cloud systems based on virtualization technology, sleep mode, and multitier cloud architecture. And then, we set forth the motivation for our research.

2.1. Virtualization Technology-Based Energy Conservation Research. In recent years, for utilizing the physical resource optimally, the study of energy conservation strategy for virtual machine (VM) configuration, migration, and consolidation has become a focus of energy conservation research in cloud systems.

Auday et al. considered migration and placement of VMs to enhance the energy efficiency in cloud infrastructure. In order to minimize the additional energy consumption generated by the VM migration, they proposed a distributed approach to an energy-efficient dynamic VM consolidation policy. The approach determined which VMs are migrated and where the selected VMs for migration are placed [6]. For solving the problem of under-utilization of servers in a cloud system, Zakarya et al. used VM consolidation to reduce the number of hosts in use. They explored the impact of VM allocation on energy efficiency and proposed a dynamic VM migration approach, in which the VMs are migrated only if the migration cost could be recovered [7].

Through modeling the energy-aware allocation and consolidation, Ghribi et al. presented an optimal allocation algorithm with a consolidation algorithm relying on migration of VMs to minimize the overall energy consumption in the cloud system. The allocation algorithm was solved as a bin-packing problem aiming to minimize the energy consumption. The consolidation algorithm was based on a linear and integer formulation of VM migration to adapt the placement for released resources [8]. Aiming to save energy and minimize resource wastage, Sharma et al. proposed a multiobjective VM allocation and migration scheme, in which the allocation of VMs was carried out using a hybrid approach of a genetic algorithm and particle swarm optimization [9]. Based on the virtualization technology, the above research improved the utilization rate of the physical resources in use and contributed to the energy conservation.

2.2. Sleep Mode-Based Energy Conservation Research. The sleep mode-based energy conservation strategy is implemented by switching the idle server to a low-power sleep state for the purpose of reducing idle energy consumption in the cloud system.

Jin et al. proposed a clustered VM allocation strategy on the resource layer of the cloud system based on a sleep mode with a wake-up threshold. By establishing a queue with an N-policy and asynchronous vacations of partial servers, they derived the performance measures in terms of the average latency of requests and the energy-saving rate of the system [10]. By using a hybrid shuffled frog leaping algorithm, Luo et al. proposed a dynamic VM allocation scheme, which applied a live VM migration strategy and switched some free resource nodes into a sleep mode to reduce energy consumption [3]. Farahnakian et al. developed a dynamic VM consolidation method to solve the optimization problem for setting the number of active hosts based on the utilization of existing resources. The proposed method could make a decision on when to switch a host into the working or sleep mode [11].

Sridharshini et al. proposed an energy-aware scheduling algorithm and a live migration algorithm to efficiently utilize the resources in a cloud system. These two algorithms were used to consolidate heterogeneous workloads to minimize the number of physical machines (PMs) and switch the idle PMs to the sleep mode to reduce energy consumption [12]. The studies mentioned above showed a certain degree of enhanced energy efficiency due to the introduction of a sleep mode.

2.3. Energy Conservation Research under a Multitier Cloud Architecture. A multitier cloud architecture contains multiple separate parts such as an "application layer," a "management layer," and a "resource layer" [13]. Some works have appeared examining the energy consumption management in a multitier cloud architecture.

Usman et al. proposed a cloud architecture composed of four modules: broker, cloud manager, VM manager, and resource scheduler. By using an Interior Search Algorithm (ISA), they developed an energy-efficient VM allocation technique to overcome high energy consumption and reduce under-utilized resources in a cloud system [14]. Aiming to use the computing resources productively and energy efficiently, Beloglazov presented a three-tier cloud architecture composed of a global resource manager, user applications, and resource pools. He proposed a distributed dynamic VM consolidation approach utilizing fine-grained fluctuations in the application workloads to minimize the number of active physical nodes [15].

Zhu et al. proposed a cloud framework composed of four modules: application agent, VM allocation center, global scheduling center, and resource pools. In addition, they designed a resource allocation and scheduling strategy to reduce the energy consumption on both the system level and the component level [16]. In order to promote energy efficiency in a cloud system, Ghosh et al. developed a multitier cloud architecture composed of a resource provisioning decision layer, a VM deployment layer, and an actual service layer. Furthermore, for reducing the complexity of performance analysis, they developed a multilevel interactive stochastic submodel method to derive the performance measures of the system [17]. Obviously, it is more reasonable to study the energy consumption problem by considering a multitier cloud architecture.

2.4. Motivation for Our Research. Inspired by the work mentioned above, in this paper, we propose a dynamic energy-efficient resource management scheme in a cloud system. Considering that it is more realistic to study energy conservation under a multitier cloud architecture, we present a cloud architecture composed of a task scheduling decision layer, a resource provisioning layer, and an actual service layer. It's noted that switching all the idle servers to a low-power sleep state may deteriorate the response performance. To save energy as well as to maintain the cloud user's quality of experience, we configure PMs into two pools: a hot pool and a warm pool. The PMs in the hot pool keep working continuously to provide cloud services instantly for the arriving requests. The PMs in the warm pool are turned on, but remain in a dynamic sleep mode to reduce energy consumption.

In addition, this paper also considers the provisioning process of VMs in both of the two pools. Concretely, each PM is configured with a resource search engine (RSE) that finds an available VM for each request, and the RSE is set to sleep synchronously with all the VMs on the PM to conserve energy. To analyze the proposed scheme, we establish a hybrid queueing system composed of three stochastic submodels with synchronous multiple vacations, and we study the system performance through theoretical analysis and numerical experiments. By building a system cost function, we study the trade-off between different performance measures and present an improved SSA to optimize the sleep mechanism.

The remainder of this paper is organized as follows. In Section 3, by considering a multitier cloud architecture and two PM pools, we propose an energy-efficient resource management scheme with a synchronous sleep mechanism. In Section 4, we establish a hybrid queueing system composed of three submodels. In Section 5, we analyze the steady-state probability distribution of the queueing system by establishing a three-dimensional Markov chain. In Section 6, based on model analysis results, we evaluate the average latency of requests and the energy-saving rate of the system. In Section 7, we show the influence of the sleep mechanism on the performance measures by using numerical results. In Section 8, we present an improved intelligent algorithm to optimize the sleep mechanism. Finally, we summarize the whole paper in Section 9.

3. Scheme Description

Proper deployment of VMs is critical for the energy conservation and the Quality of Service (QoS) guarantee in a cloud system. In order to save energy and maintain the QoS, this paper proposes a dynamic energy-efficient resource management scheme, where the PMs are grouped into two pools: a hot pool and a warm pool. In the hot pool, the PMs are running continuously and the VMs hosted on a PM are always available. This means that the requests allocated to the hot pool can be served quickly so that the QoS of the cloud system can be guaranteed. In the warm pool, a synchronous sleep mechanism is introduced for the purpose of achieving a better energy-saving effect. The service provided by the warm pool can be delayed by the sleep mechanism. We call the PMs, the RSE, and the VMs in hot pool, the hot PMs, the hot RSE, and the hot VMs. And we call the PMs, the RSE, and the VMs in warm pool, the warm PMs, the warm RSE, and the warm VMs. Based on a multitier cloud architecture and a grouping approach for the PMs, we propose a novel resource management scheme shown in Figure 1.

In Figure 1, we assume that each PM is equipped with a RSE and the maximum number of VMs deployed on one PM is m. We also assume that the numbers of the identical PMs in the hot pool and the warm pool are n_h and n_w, respectively, where $n_h = 1, 2, \ldots$ and $n_w = 1, 2, \ldots$. The life cycle of a request with the resource management scheme proposed in this paper is illustrated as follows:

(1) All the requests are assumed to be homogeneous and enter a first-come, first-served (FCFS) queue in the system buffer. The request at the head of the queue firstly receives the service of the Task Scheduling Decision Engine (TSDE). As long as the hot pool is not full, the request will be allocated by the TSDE to the hot pool. Otherwise, the request will be allocated to the warm pool.

(2) The request allocated to the hot pool randomly enters the FCFS queue in one of the hot PM buffers. The request at the head of the queue is processed by a RSE, which is used to find a VM on the selected PM for resource provision. If at least one idle VM exists on one of the hot PMs, the RSE provisions an available VM to the request, and the request is immediately served by the running VM. After the service is completed, the request will depart the system.

(3) The request allocated to the warm pool randomly enters the FCFS queue in one of the warm PM buffers. The request at the head of the queue can have its service delayed due to the introduction of the sleep mechanism. On one of the warm PMs, once all the requests are processed, the RSE together with all the VMs enter a sleep period. Meanwhile, a sleep timer is started. When the sleep timer expires, if at least one request exists in the warm buffer, the RSE and all the VMs on the PM will wake up, otherwise they will enter the next sleep period.

Then, we build a hybrid queueing system to mathematically derive the system performance measures and to solve the performance optimization problem with the proposed scheme.

4. System Model

In this section, we model the proposed scheme as three submodels based on the continuous-time environment as follows. Then, we obtained the continuous-time Markov chains (CTMC) of the hot PM and the warm PM, respectively.

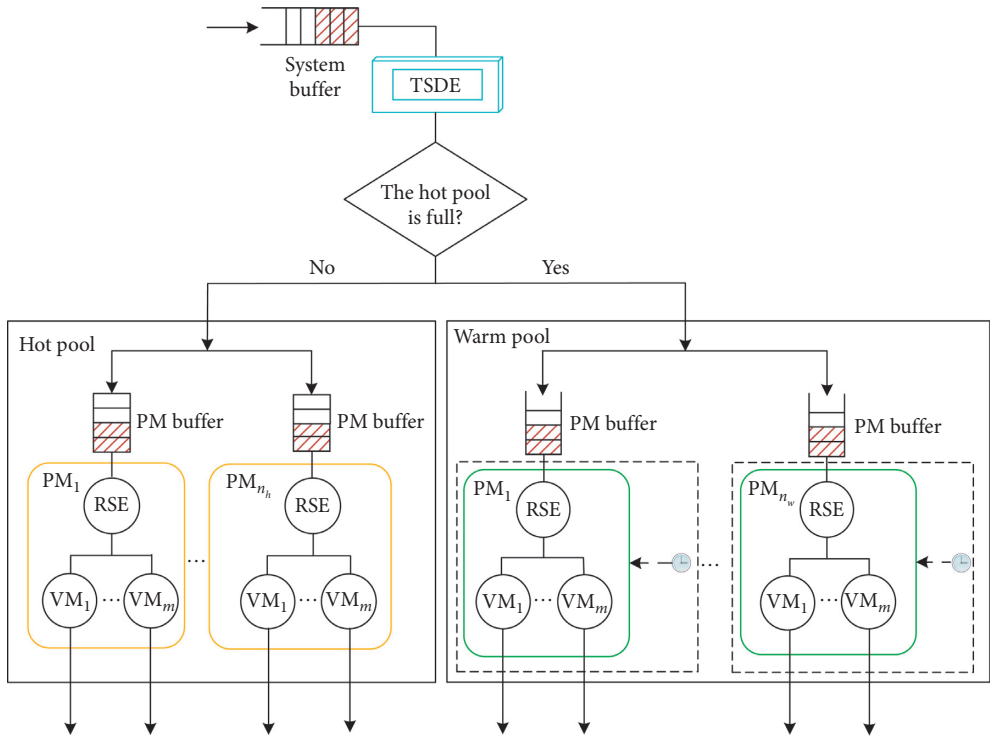


Figure 1: Multitier cloud architecture-based resource management scheme proposed.

4.1. TSDE Submodel. In cloud systems, some practical requests are independent with each other, while other practical requests are correlated. The computing requests initiated by users are usually uncorrelated. Therefore, the arrival process with Poisson distribution is considered to be appropriate for capturing the stochastic behavior of a cloud computing system with uncorrelated traffic [18].

In this research, we focus on user's initiated requests. Therefore, we can make the following assumptions. In the request scheduling decision process, we assume that the arrival intervals of requests and the service times of requests are independent, identically distributed (i.i.d) random variables. Request arrivals at the cloud system presented in this paper are supposed to follow a Poisson process with arrival rate $\lambda_0, \lambda_0 > 0$. The service time of a request processed by the TSDE is supposed to follow an exponential distribution with service rate $\delta, \delta > 0$.

Therefore, we build a single server queue for the task-scheduling decision process. We define the service intensity ρ_0 of the TSDE to be the number of request arrivals at the TSDE during the service time of a request. ρ_0 is given as follows:

$$\rho_0 = \frac{\lambda_0}{\delta}. \tag{1}$$

We define the latency W_{dec} of a request in the TSDE buffer to be the time duration from the instant of a request arriving at the TSDE buffer to the instant of the request departing the TSDE buffer. The average latency $E[W_{\text{dec}}]$ of requests in the TSDE buffer is obtained as follows:

$$E[W_{\text{dec}}] = \frac{\rho_0}{\delta(1-\rho_0)}. \tag{2}$$

Substituting equation (1) into equation (2), we have

$$E[W_{\text{dec}}] = \frac{\lambda_0}{\delta(\delta-\lambda_0)}. \tag{3}$$

4.2. Hot Pool Submodel. In this paper, we focus on a hot PM to build a queue model as a submodel of the system called the hot pool submodel and study the performance of the hot pool. Let $L, L < +\infty$, be the capacity of the hot PM buffer. Let random variable $N_1(t) = i, i \in \{0, 1, \ldots, L\}$, be the number of requests in the hot PM buffer at instant $t, t \geq 0$. Let random variable $J_1(t) = j, j \in \{0, 1\}$, be the state of the RSE, whether it is busy with provisioning a VM $(j = 1)$ or not $(j = 0)$. Each hot VM processes a request by loading a software environment (SE). Let random variable $S_1(t) = k, k \in \{0, 1, \ldots, m\}$, be the number of hot VMs loaded with an SE at instant t. We call $N_1(t)$ the system level, $J_1(t)$ the system stage, and $S_1(t)$ the system phase. $\{(N_1(t), J_1(t), S_1(t)), t \geq 0\}$ constitutes a three-dimensional continuous-time stochastic process with state space Ω_1 as follows:

$$\begin{aligned}\Omega_1 = &\{(0, j, k) | j \in \{0, 1\}, k \in \{0, 1, \ldots, m-1\}\} \\ &\cup \{(i, 0, m) | i \in \{0, 1, \ldots, L\}\} \\ &\cup \{(i, 1, k) | i \in \{1, 2, \ldots, L\}, k \in \{0, 1, \ldots, m-1\}\}.\end{aligned} \tag{4}$$

We assume that a newly arriving request is randomly allocated to one of the hot PMs. The decomposition of a

Poisson process yields multiple Poisson processes [19]. The request arrivals at each hot PM are supposed to follow a Poisson process with arrival rate λ_1. We have

$$\lambda_1 = \frac{\lambda_0}{n_h}. \tag{5}$$

We assume that the service time of a request processed by the hot RSE follows an exponential distribution with service rate $\beta_1, \beta_1 > 0$. The service time of a request processed by the hot VM loaded with SE is supposed to follow an exponential distribution with service rate $\mu_1, \mu_1 > 0$.

Based on these assumptions, the stochastic process $\{(N_1(t), J_1(t), S_1(t)), t \geq 0\}$ can be regarded as a CTMC.

We define $\pi_{i,j,k}$ as the steady-state probability distribution of the hot PM for the system level being equal to i, the system stage being equal to j, and the system phase being equal to k. $\pi_{i,j,k}$ is expressed as follows:

$$\pi_{i,j,k} = \lim_{t \longrightarrow \infty} P\{N_1(t) = i, J_1(t) = j, S_1(t) = k\}, \tag{6}$$

where $(i, j, k) \in \Omega_1$.

We define $\boldsymbol{\pi}_i$ as the steady-state probability distribution vector of the system level being equal to i. $\boldsymbol{\pi}_i$ can be given as follows:

$$\boldsymbol{\pi}_i = \begin{cases} \left(\pi_{0,0,0}, \pi_{0,0,1}, \ldots, \pi_{0,0,m-1}, \pi_{0,1,0} \ldots, \pi_{0,1,m-1}\right), & i = 0, \\ \left(\pi_{i,0,m}, \pi_{i,1,0}, \ldots, \pi_{i,1,m-1}\right), & 0 < i \leq L. \end{cases} \tag{7}$$

The steady-state probability distribution $\boldsymbol{\Pi}_1$ of the CTMC $\{(N_1(t), J_1(t), S_1(t)), t \geq 0\}$ is composed of $\boldsymbol{\pi}_\mathbf{i}, 0 \leq i \leq L$. $\boldsymbol{\Pi}_1$ is given as follows:

$$\boldsymbol{\Pi}_1 = \left(\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_L\right). \tag{8}$$

4.3. Warm Pool Submodel. In order to evaluate the performance of the warm pool, we focus on a warm PM to build a queue model as another submodel of the system called the warm pool submodel. We assume that the capacity of the warm PM buffer is infinite. Let $N_2(t) = i, i \in \{0, 1, \ldots\}$, be the number of requests in the warm PM buffer at instant t. Unlike the hot PMs, a synchronous sleep mechanism is introduced to each warm PM. The RSE and all the VMs on one warm PM will go to sleep synchronously if possible. Let $J_2(t) = j, j \in \{0, 1, 2\}$, be the state of the warm RSE. $j = 0$ means the warm RSE is asleep, $j = 1$ means the warm RSE is idle, and $j = 2$ means the warm RSE is busy with provisioning a VM for a request. Just like those in the hot pool, each warm VM also needs to load an SE for processing a request. Let $S_2(t) = k, k \in \{0, 1, \ldots, m\}$, be the number of warm VMs loaded with an SE at instant t. We call $N_2(t)$ the system level, $J_2(t)$ the system stage, and $S_2(t)$ the system phase. $\{(N_2(t), J_2(t), S_2(t)), t \geq 0\}$ constitutes a three-dimensional continuous-time stochastic process with state space Ω_2 as follows:

$$\begin{aligned} \Omega_2 = &\{(0, 1, k) | k \in \{1, 2, \ldots, m-1\}\} \\ &\cup \{(i, 0, 0) | i \in \{0, 1, \ldots\}\} \\ &\cup \{(i, 1, m) | i \in \{0, 1, \ldots\}\} \\ &\cup \{(i, 2, k) | i \in \{0, 1, \ldots\}, k \in \{0, 1, \ldots, m-1\}\}. \end{aligned} \tag{9}$$

The general input flow is split into two streams, one is into the hot pool and the other is into the warm pool. In Section 4.1, the general request arrivals are assumed to follow a Poisson process, so the request arrivals at the warm pool also follow a Poisson process. We assume that a newly arriving request is randomly allocated to one of the warm PMs. The arrival rate of the requests at each warm PM is given as follows:

$$\lambda_2 = \frac{\lambda_0 (1 - q)}{n_w}, \tag{10}$$

where λ_0 is the arrival rate of the requests at the TSDE submodel and q is the probability that a newly arriving request can be accepted by the hot pool. q is calculated as follows:

$$q = 1 - \left(\sum_{i=0}^{m-1} \pi_{L,1,i} + \pi_{L,0,m}\right)^{n_h}. \tag{11}$$

We assume that the service time of a request processed by the warm RSE follows an exponential distribution with service rate $\beta_2, \beta_2 > 0$. The service time of a request processed by the warm VM loaded with an SE is supposed to follow an exponential distribution with service rate $\mu_2, \mu_2 > 0$. A sleep timer is used to control the time length of a sleep period. The time length of the sleep timer is assumed to follow an exponential distribution with sleep parameter $\phi, \phi > 0$.

Based on these assumptions, the stochastic process $\{(N_2(t), J_2(t), S_2(t)), t \geq 0\}$ can be regarded as a CTMC.

We define $\pi^*_{i,j,k}$ as the steady-state probability distribution of the warm PM for the system level being equal to i, the system stage being equal to j, and the system phase being equal to k. $\pi^*_{i,j,k}$ is given by

$$\pi^*_{i,j,k} = \lim_{t \longrightarrow \infty} P\{N_2(t) = i, J_2(t) = j, S_2(t) = k\}, \tag{12}$$

where $(i, j, k) \in \Omega_2$.

We define $\boldsymbol{\pi}^*_i$ as the steady-state probability distribution vector of the warm PM for the system level being equal to i. $\boldsymbol{\pi}^*_i$ can be given as follows:

$$\boldsymbol{\pi}^*_i = \begin{cases} \left(\pi^*_{0,0,0}, \pi^*_{0,1,1}, \ldots, \pi^*_{0,1,m}, \pi^*_{0,2,0}, \ldots, \pi^*_{0,2,m-1}\right), & i = 0, \\ \left(\pi^*_{i,0,0}, \pi^*_{i,1,m}, \pi^*_{i,2,0}, \ldots, \pi^*_{i,2,m-1}\right), & i \geq 1. \end{cases} \tag{13}$$

The steady-state probability distribution $\boldsymbol{\Pi}_2$ of the CTMC $\{(N_2(t), J_2(t), S_2(t)), t \geq 0\}$ is composed of $\boldsymbol{\pi}^*_i, i \geq 0$ and given as follows:

$$\boldsymbol{\Pi}_2 = \left(\boldsymbol{\pi}^*_0, \boldsymbol{\pi}^*_1, \ldots\right). \tag{14}$$

5. Model Analysis

In this section, we construct the transition rate matrixes in the context of CTMC and derive the steady-state probability distributions of the hot PM and the warm PM, respectively.

5.1. Steady-State Probability Distributions of the Hot PM. Let $\mathbf{Q}_1$ be the one-step state transition rate matrix of the CTMC $\{(N_1(t), J_1(t), S_1(t)), t \geq 0\}$. Let $\mathbf{Q}_{u,v}$ be the one-step state transition rate submatrix of $\mathbf{Q}_1$ for the system level changing to $v, v = 0, 1, \ldots, L$, from $u, u = 0, 1, \ldots, L$. For the convenience of expression, we denote $\mathbf{Q}_{u,u-1}$ as $\mathbf{B}_u$, $\mathbf{Q}_{u,u}$ as $\mathbf{A}_u$, and $\mathbf{Q}_{u,u+1}$ as $\mathbf{C}_u$.

(1) For the case of $u = 0$, there are no requests in the hot PM buffer.

If a new request arrives at the hot PM, the system state changes in the following two cases:

(a) When the number of the hot VMs loaded with an SE is less than m and the hot RSE is idle, the newly arriving request accesses the hot RSE immediately. The system level and the system phase remain unchanged, but the system stage increases by one. The system state transfers to $(0, 1, k)$ from $(0, 0, k), 0 \leq k \leq m - 1$, with λ_1.

(b) When the number of the hot VMs loaded with an SE is less than m, but the hot RSE is busy, or the number of the hot VMs loaded with an SE is up to m, the newly arriving request has to wait in the hot PM buffer. The system level increases by one, but the system stage and the system phase remain unchanged. The system state transfers to $(1, 1, k)$ from $(0, 1, k), 0 \leq k \leq m - 1$, or to $(1, 0, m)$ from $(0, 0, m)$, with λ_1.

If a request is completely processed by the hot RSE, one of the deployed hot VMs loads the SE and processes this request. The system level remains unchanged, the system stage decreases by one, and the system phase increases by one. The system state transfers to $(0, 0, k + 1)$ from $(0, 1, k), 0 \leq k \leq m - 1$, with β_1.

If a request is completely processed by a hot VM and departs the system, the SE is removed. The system level and the system stage remain unchanged, but the system phase decreases by one. The system state transfers to $(0, 0, k - 1)$ from $(0, 0, k), 1 \leq k \leq m$, or to $(0, 1, k - 1)$ from $(0, 1, k), 1 \leq k \leq m - 1$, with $k\mu_1$.

Otherwise, the system state remains fixed at $(0, 0, k), 0 \leq k \leq m$, with $-(\lambda_1 + k\mu_1)$, or at $(0, 1, k), 0 \leq k \leq m - 1$, with $-(\lambda_1 + k\mu_1 + \beta_1)$.

In summary, $\mathbf{A}_0$ is a $(2m + 1) \times (2m + 1)$ matrix given as follows:

$$\mathbf{A}_0 = \begin{pmatrix} -\lambda_1 & & & & & & & \lambda_1 & & & & & & \\ \mu_1 & -(\lambda_1 + \mu_1) & & & & & & & \lambda_1 & & & & & \\ & 2\mu_1 & -(\lambda_1 + 2\mu_1) & & & & & & & \lambda_1 & & & & \\ & & \ddots & \ddots & & & & & & & \ddots & & & \\ & & & (m-1)\mu_1 & -(\lambda_1 + (m-1)\mu_1) & & & & & & & & & \lambda_1 \\ & & & & m\mu_1 & -(\lambda_1 + m\mu_1) & & & & & & & & \\ \beta_1 & & & & & & -(\lambda_1 + \beta_1) & & & & & & & \\ & \beta_1 & & & & & \mu_1 & -(\lambda_1 + \beta_1 + \mu_1) & & & & & & \\ & & \ddots & & & & & \ddots & \ddots & & & & & \\ & & & \beta_1 & & & & & (m-1)\mu_1 & -(\lambda_1 + \beta_1 + (m-1)\mu_1) & & & & \end{pmatrix}. \tag{15}$$

$\mathbf{C}_0$ is a $(2m + 1) \times (m + 1)$ matrix given as follows:

$$\mathbf{C}_0 = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \\ \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_1 \end{pmatrix}. \tag{16}$$

(2) For the case of $1 \leq u < L$, there is at least one request in the hot PM buffer, and the hot PM buffer is not full. If a new request arrives at the hot PM, the newly arriving request has to wait at the hot PM buffer. The system level increases by one, but the system stage and the system phase remain unchanged. The system state transfers to $(u + 1, 0, m)$ from $(u, 0, m)$ or to $(u + 1, 1, k)$ from $(u, 1, k), 0 \leq k \leq m - 1$, with λ_1.

If a request is completely processed by the hot RSE, one of the deployed hot VMs loads the SE and provides service for this request. The system state changes in the following two cases:

(a) When the number of the hot VMs loaded with an SE is less than m, the hot RSE processes the first request waiting in the hot PM buffer

immediately. The system level decreases by one, the system stage remains unchanged, and the system phase increases by one. The system state transfers to $(u-1,1,k+1)$ from $(u,1,k), 0 \le k \le m-2$, with β_1.

(b) When the number of the hot VMs loaded with an SE is up to m, the hot RSE becomes idle, and the requests in the hot PM buffer keep waiting. The system level remains unchanged, the system stage decreases by one, and the system phase increases by one. The system state transfers to $(u,0,m)$ from $(u,1,m-1)$ with β_1.

If a request is completely processed by a hot VM and departs the system, the SE is removed. The system state changes in the following two cases:

(a) When the hot RSE is idle, the first request waiting in the hot PM buffer accesses the hot RSE immediately. The system level and the system phase decrease by one, and the system stage increases by one. The system level transfers to $(u-1,1,m-1)$ from $(u,0,m)$ with $m\mu_1$.

(b) When the hot RSE is busy, the requests in the hot PM buffer keep waiting. The system level and the system stage remain unchanged, but the system phase decreases by one. The system state transfers to $(u,1,k-1)$ from $(u,1,k), 1 \le k \le m-1$, with $k\mu_1$.

Otherwise, the system state remains fixed at $(u,0,m)$ with $-(\lambda_1 + m\mu_1)$, or at $(u,1,k), 0 \le k \le m-1$, with $-(\lambda_1 + k\mu_1 + \beta_1)$.

In summary, $\mathbf{B}_1$ is an $(m+1)\times(2m+1)$ matrix given as follows:

$$\mathbf{B}_1 = \begin{pmatrix} 0 & \cdots & 0 & 0 & \cdots & m\mu_1 \\ 0 & \cdots & 0 & \beta_1 & \cdots & 0 \\ \vdots & & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & \beta_1 \\ 0 & \cdots & 0 & 0 & \cdots & 0 \end{pmatrix}. \tag{17}$$

Let $\mathbf{B}$ represent $\mathbf{B}_u, u = 2,3,\ldots,L-1$. $\mathbf{B}$ is an $(m+1)\times(m+1)$ matrix given as follows:

$$\mathbf{B} = \begin{pmatrix} 0 & 0 & 0 & \cdots & m\mu_1 \\ 0 & 0 & \beta & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & \beta_1 \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix}. \tag{18}$$

Let $\mathbf{A}$ represent $\mathbf{A}_u, u = 1,2,\ldots,L-1$. $\mathbf{A}$ is an $(m+1)\times(m+1)$ lower triangular matrix given by

$$\mathbf{A} = \begin{pmatrix} -(\lambda_1 + m\mu_1) & & & & \\ & -(\lambda_1 + \beta_1) & & & \\ & \mu_1 & -(\lambda_1 + \beta_1 + \mu_1) & & \\ & & \ddots & \ddots & \\ \beta_1 & & (m-1)\mu_1 & -(\lambda_1 + \beta_1 + (m-1)\mu_1) & \end{pmatrix}. \tag{19}$$

Let $\mathbf{C}$ represent $\mathbf{C}_u, u = 1,2,\ldots,L-1$. $\mathbf{C}$ is an $(m+1)\times(m+1)$ diagonal matrix given as follows:

$$\mathbf{C} = \begin{pmatrix} \lambda_1 & & & \\ & \lambda_1 & & \\ & & \ddots & \\ & & & \lambda_1 \end{pmatrix}. \tag{20}$$

(3) For the case of $u = L$, the hot PM buffer is full. Therefore, no new requests can join the hot PM.

If a request is completely processed by the hot RSE, one of the deployed VMs loads the SE and processes this request. The system state changes in the following two cases:

(a) When the number of the hot VMs loaded with an SE is less than m, the hot RSE processes the first request waiting in the hot PM buffer immediately. The system level decreases by one, the system stage remains unchanged, and the system phase increases by one. The system state transfers to $(L-1,1,k+1)$ from $(L,1,k), 0 \le k \le m-2$, with β_1.

(b) When the number of the hot VMs loaded with an SE is up to m, no other hot VMs can be provisioned by the hot RSE, so the hot RSE becomes idle, and all the requests in the hot PM buffer keep waiting. The system level remains unchanged, the system stage decreases by one, and the system phase increases by one. The system state transfers to $(L,0,m)$ from $(L,1,m-1)$ with β_1.

If a request is completely processed by a hot VM and departs the system, the SE is removed. The system state changes in the following two cases:

(a) When the hot RSE is idle, the first request waiting in the hot PM buffer is processed by the hot RSE immediately. The system level and the system phase decrease by one, and the system stage increases by

one. The system state transfers to $(L-1,1,m-1)$ from $(L,0,m)$ with $m\mu_1$.

(b) When the hot RSE is busy, all the requests in the hot PM buffer keep waiting. The system level and the system stage remain unchanged, but the system phase decreases by one. The system state transfers to $(L,1,k-1)$ from $(L,1,k), 1 \le k \le m-1$, with $k\mu_1$.

Otherwise, the system state remains fixed at $(L,0,m)$ with $-m\mu_1$, or at $(L,1,k), 0 \le k \le m-1$, with $-(k\mu_1+\beta_1)$.

Obviously, $\mathbf{B}_L$ is an $(m+1)\times(m+1)$ matrix, and $\mathbf{B}_L = \mathbf{B}$. $\mathbf{A}_L$ is an $(m+1)\times(m+1)$ lower triangular matrix given as follows:

$$\mathbf{A}_L = \begin{pmatrix} -m\mu_1 & & & & \\ & -\beta_1 & & & \\ & \mu & -(\beta_1+\mu_1) & & \\ & & \ddots & \ddots & \\ \beta_1 & & (m-1)\mu_1 & -(\beta_1+(m-1)\mu_1) & \end{pmatrix}. \tag{21}$$

At present, we have obtained all the submatrices in the one-step state transition rate matrix $\mathbf{Q}_1$. $\mathbf{Q}_1$ can be written as follows:

$$\mathbf{Q}_1 = \begin{pmatrix} \mathbf{A}_0 & \mathbf{C}_0 & & & & \\ \mathbf{B}_1 & \mathbf{A} & \mathbf{C} & & & \\ & \mathbf{B} & \mathbf{A} & \mathbf{C} & & \\ & & \ddots & \ddots & \ddots & \\ & & & \mathbf{B} & \mathbf{A} & \mathbf{C} \\ & & & & \mathbf{B} & \mathbf{A}_L \end{pmatrix}. \tag{22}$$

The steady-state probability distribution $\mathbf{\Pi}_1$ of the CTMC $\{(N_1(t), J_1(t), S_1(t)), t \ge 0\}$ satisfies the following equilibrium equation and normalization condition:

$$\begin{cases} \mathbf{\Pi}_1\mathbf{Q}_1 = \mathbf{0}, \\ \mathbf{\Pi}_1\mathbf{e}_1 = 1, \end{cases} \tag{23}$$

where $\mathbf{e}_1$ is an $((L+2)m+L+1)\times 1$ vector with all elements being equal to 1.

By solving equation (23), we derive the steady-state probability distribution $\mathbf{\Pi}_1$ of the CTMC $\{(N_1(t), J_1(t), S_1(t)), t \ge 0\}$, where $\mathbf{\Pi}_1 = (\boldsymbol{\pi}_0, \boldsymbol{\pi}_1, \ldots, \boldsymbol{\pi}_L)$.

5.2. Steady-State Probability Distribution of the Warm PM. Let $\mathbf{Q}_2$ be the one-step state transition rate matrix of the CTMC $\{(N_2(t), J_2(t), S_2(t)), t \ge 0\}$. Let $\mathbf{Q}^*_{u,v}$ be the one-step state transition rate submatrix of $\mathbf{Q}_2$ for the system level changing to $v, v = 0,1,\ldots$, from $u, u = 0,1,\ldots$. We denote $\mathbf{Q}^*_{u,u-1}$ as $\mathbf{B}^*_u$, $\mathbf{Q}^*_{u,u}$ as $\mathbf{A}^*_u$, and $\mathbf{Q}^*_{u,u+1}$ as $\mathbf{C}^*_u$.

(1) For the case of $u = 0$, there are no requests in the warm PM buffer.

If a new request arrives at the warm PM, the system state changes in the following three cases:

(a) When the warm RSE and the warm VMs are asleep, the newly arriving request has to wait in the warm PM buffer until the sleep timer expires. The system level increases by one, but the system stage and the system phase remain unchanged. The system state transfers to $(1,0,0)$ from $(0,0,0)$ with λ_2.

(b) When the warm RSE and the warm VMs are awake, and the number of the warm VMs loaded with an SE is up to m; no other warm VMs can be provisioned by the hot RSE. The newly arriving request has to wait in the warm PM buffer. The system level increases by one, but the system stage and the system phase remain unchanged. The system state transfers to $(1,1,m)$ from $(0,1,m)$ with λ_2.

(c) When the warm RSE and the warm VMs are awake, and the number of the warm VMs loaded with an SE is less than m, at least one VM can be provisioned. If the warm RSE is busy, the newly arriving request has to wait in the warm PM buffer. The system level increases by one, but the system stage and the system phase remain unchanged. The system state transfers to $(1,2,k)$ from $(0,2,k), 0 \le k \le m-1$, with λ_2. If the warm RSE is idle, the newly arriving request accesses the warm RSE immediately. The system level and the system phase remain unchanged, but the system stage increases by one. The system state transfers to $(0,2,k)$ from $(0,1,k), 1 \le k \le m-1$, with λ_2.

If a request is completely processed by the warm RSE, one of the deployed warm VMs loads the SE and processes this request. The system level remains unchanged, the system stage decreases by one, and the system phase increases by one. The system state transfers to $(0,1,k+1)$ from $(0,2,k), 0 \le k \le m-1$, with β_2.

If a request is completely processed by a warm VM and departs the system, the system state changes in the following two cases:

(a) When the warm RSE is idle and there is only one warm VM loaded with an SE, the used SE is removed. The warm RSE and the warm VMs enter a sleep period immediately. The system level remains unchanged, but the system stage and the system phase decrease by one. The system state transfers to $(0,0,0)$ from $(0,1,1)$ with μ_2.

(b) When the warm RSE is idle and there are at least two warm VMs loaded with an SE or the warm RSE is busy and there is at least one warm VM loaded with an SE, the used SE is removed. The system level and the system stage remain unchanged, but the system phase decreases by one. The system state transfers to $(0,1,k-1)$ from $(0,1,k), 2 \le k \le m$, or to $(0,2,k-1)$ from $(0,2,k), 1 \le k \le m-1$, with $k\mu_2$.

Otherwise, the system state remains fixed at $(0,0,0)$ with $-\lambda_2$, at $(0,1,k), 1 \le k \le m$, with $-(\lambda_2 + k\mu_2)$, or at $(0,2,k), 0 \le k \le m-1$, with $-(\lambda_2 + k\mu_2 + \beta_2)$.

In summary, $\mathbf{A}_0^*$ is a $(2m+1) \times (2m+1)$ matrix given as follows:

$$\mathbf{A}_0^* = \begin{pmatrix} -\lambda_2 & & & & & & & & & & & & & \\ \mu_2 & -(\lambda_2+\mu_2) & & & & & & \lambda_2 & & & & & & \\ & 2\mu_2 & -(\lambda_2+2\mu_2) & & & & & & \lambda_2 & & & & & \\ & & \ddots & \ddots & & & & & & \ddots & & & & \\ & & & (m-1)\mu_2 & -(\lambda_2+(m-1)\mu_2) & & & & & & & & \lambda_2 & \\ & & & & m\mu_2 & -(\lambda_2+m\mu_2) & & & & & & & & \\ & \beta_2 & & & & & -(\lambda_2+\beta_2) & & & & & & & \\ & & \beta_2 & & & & \mu_2 & -(\lambda_2+\beta_2+\mu_2) & & & & & & \\ & & & \ddots & & & & \ddots & \ddots & & & & & \\ & & & & \beta_2 & & & & & (m-1)\mu_2 & -(\lambda_2+\beta_2+(m-1)\mu_2) & & & \end{pmatrix}. \tag{24}$$

$\mathbf{C}_0^*$ is a $(2m+1) \times (m+2)$ matrix given as follows:

$$\mathbf{C}_0^* = \begin{pmatrix} \lambda_2 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_2 \end{pmatrix}. \tag{25}$$

(2) For the case of $1 \le u < \infty$, there is at least one request in the warm PM buffer.

If there are no new request arrivals on the warm PM, while the warm RSE and the warm VMs are asleep, once the sleep timer expires, the warm RSE wakes up and processes the first request waiting in the warm PM buffer immediately. The system level decreases by one, the system stage increases by two, and the system phase remains unchanged. The system state transfers to $(u-1,2,0)$ from $(u,0,0)$ with ϕ.

If a new request arrives at the warm PM, the newly arriving request has to wait in the warm buffer. The system level increases by one, but the system stage and the system phase remain unchanged. The system state transfers to $(u+1,0,0)$ from $(u,0,0)$, to $(u+1,1,m)$ from $(u,1,m)$, or to $(u+1,2,k)$ from $(u,2,k), 0 \le k \le m-1$, with λ_2.

If a request is completely processed by the warm RSE, one of the deployed warm VMs loads the SE and provides service for this request. The system state changes in the following two cases:

(a) When the number of the warm VMs loaded with an SE is less than m, the warm RSE processes the first request waiting in the warm PM buffer immediately. The system level decreases by one, the system stage remains unchanged, and the system phase increases by one. The system state transfers to $(u-1,2,k+1)$ from $(u,2,k), 0 \le k \le m-2$, with β_2.

(b) When the number of the warm VMs loaded with an SE is up to m, no other warm VMs can be provisioned by the warm RSE. Therefore, the warm RSE becomes idle and none of the requests waiting in the warm PM buffer can access the warm RSE. The system level remains unchanged, the system stage decreases by one, and the system phase increases by one. The system state transfers to $(u,1,m)$ from $(u,2,m-1)$ with β_2.

If a request is completely processed by a warm VM and departs the system, the used SE is removed. The system state changes in the following two cases:

(a) When the warm RSE is idle, the first request waiting in the warm PM buffer accesses the warm RSE immediately. The system level and the system phase decrease by one, but the system stage increases by one. The system state transfers to $(u-1,2,m-1)$ from $(u,1,m)$ with $m\mu_2$.

(b) When the warm RSE is busy, none of the requests waiting in the warm PM buffer can access the warm RSE. The system level and the system stage remain unchanged, but the system phase decreases by one. The system state transfers to $(u,2,k-1)$ from $(u,2,k), 1 \le k \le m-1$, with $k\mu_2$.

Otherwise, the system state remains fixed at $(u,0,0)$ with $-(\lambda_2+\phi)$, at $(u,1,m)$ with $-(\lambda_2+m\mu_2)$, or at $(u,2,k), 0 \le k \le m-1$, with $-(\lambda_2+k\mu_2+\beta_2)$.

In summary, $\mathbf{B}_1^*$ is an $(m+2) \times (2m+1)$ matrix given as follows:

Table 1: Iteration algorithm to compute the rate matrix $\mathbf{R}$.

Step 1. Initialize the upper bound of error ε (for example, $\varepsilon = e^{-10}$). Initialize $m, L, \lambda_0, \mu_1, \mu_2, \beta_1, \beta_2$ and ϕ as needed. Initialize the matrix $\mathbf{R} = \mathbf{0}$ with the order of $(m+2) \times (m+2)$.

Step 2. Construct the matrices $\mathbf{B}^*$, $\mathbf{A}^*$, and $\mathbf{C}^*$ by

$$\mathbf{B}^* = \begin{pmatrix} 0 & 0 & \phi & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & m\mu_2 \\ 0 & 0 & 0 & \beta_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \beta_2 \\ 0 & 0 & 0 & 0 & \cdots & 0 \end{pmatrix},$$

$$\mathbf{A}^* = \begin{pmatrix} -(\lambda_2+\phi) & & & & & \\ & -(\lambda_2+m\mu_2) & & & & \\ & & -(\lambda_2+\beta_2) & & & \\ & & \mu_2 & -(\lambda_2+\beta_2+\mu_2) & & \\ & & & \ddots & \ddots & \\ & & & (m-1)\mu_2 & -(\lambda_2+\beta_2+(m-1)\mu_2) & \\ & & \beta_2 & & & \end{pmatrix},$$

$$\mathbf{C}^* = \begin{pmatrix} \lambda_2 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_2 \end{pmatrix}.$$

Step 3. Calculate $\mathbf{W}, \mathbf{V}$, and $\mathbf{R}_1$ by

$\mathbf{W} = \mathbf{B}^* (\mathbf{A}^*)^{-1}$,
$\mathbf{V} = \mathbf{C}^* (\mathbf{A}^*)^{-1}$,
$\mathbf{R}_1 = -\mathbf{R}^2\mathbf{W} - \mathbf{V}$.

Step 4. While $\{\|\mathbf{R} - \mathbf{R}_1\|_\infty > \varepsilon\}$,
% $\|\mathbf{R} - \mathbf{R}_1\|_\infty = \max\left\{\sum_{i=1}^{(m+2)} \sum_{j=1}^{(m+2)} |r_{i,j} - r^*_{i,j}|\right\}$, where $r_{i,j}$ and $r^*_{i,j}$ are elements in $\mathbf{R}$ and $\mathbf{R}_1$,
Respectively.%
$\mathbf{R} = \mathbf{R}_1$.
$\mathbf{R}_1 = -\mathbf{R}^2\mathbf{W} - \mathbf{V}$.
Endwhile
$\mathbf{R} = \mathbf{R}_1$.

Step 5. Output $\mathbf{R}$.

$$\mathbf{B}_1^* = \begin{pmatrix} 0 & \cdots & \phi & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & \cdots & m\mu_2 \\ 0 & \cdots & 0 & \beta_2 & \cdots & 0 \\ \vdots & & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & \beta_2 \\ 0 & \cdots & 0 & 0 & \cdots & 0 \end{pmatrix}. \quad (26)$$

Let $\mathbf{B}^*$ represent $\mathbf{B}_u^*\,(u = 2, 3, \ldots)$. $\mathbf{B}^*$ is an $(m+2) \times (m+2)$ upper triangular matrix given as follows:

$$\mathbf{B}^* = \begin{pmatrix} 0 & 0 & \phi & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & \cdots & m\mu_2 \\ 0 & 0 & 0 & \beta_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \beta_2 \\ 0 & 0 & 0 & 0 & \cdots & 0 \end{pmatrix}. \quad (27)$$

Let $\mathbf{A}^*$ represent $\mathbf{A}_u^*\,(u = 1, 2, \ldots)$. $\mathbf{A}^*$ is an $(m+2) \times (m+2)$ lower triangular matrix given by

$$\mathbf{A}^* = \begin{pmatrix} -(\lambda_2+\phi) & & & & & \\ & -(\lambda_2+m\mu_2) & & & & \\ & & -(\lambda_2+\beta_2) & & & \\ & & \mu_2 & -(\lambda_2+\beta_2+\mu_2) & & \\ & & & \ddots & \ddots & \\ & \beta_2 & & (m-1)\mu_2 & -(\lambda_2+\beta_2+(m-1)\mu_2) & \end{pmatrix}. \tag{28}$$

Let $\mathbf{C}^*$ represent $\mathbf{C}_u^*\,(u=1,2,\ldots)$. $\mathbf{C}^*$ is an $(m+2)\times(m+2)$ diagonal matrix given as follows:

$$\mathbf{C}^* = \begin{pmatrix} \lambda_2 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_2 \end{pmatrix}. \tag{29}$$

At present, we have obtained all the submatrices in the one step state transition rate matrix $\mathbf{Q}_2$. $\mathbf{Q}_2$ can be written as follows:

$$\mathbf{Q}_2 = \begin{pmatrix} \mathbf{A}_0^* & \mathbf{C}_0^* & & & & \\ \mathbf{B}_1^* & \mathbf{A}^* & \mathbf{C}^* & & & \\ & \mathbf{B}^* & \mathbf{A}^* & \mathbf{C}^* & & \\ & & \mathbf{B}^* & \mathbf{A}^* & \mathbf{C}^* & \\ & & & \ddots & \ddots & \ddots \end{pmatrix}. \tag{30}$$

Based on the structure of the one-step state transition rate matrix $\mathbf{Q}_2$, the three-dimensional CTMC $\{(N_2(t), J_2(t), S_2(t)), t \geq 0\}$ of the warm PM can be regarded as a type of Quasi Birth-and-Death (QBD) process. Thus, we can apply the method of a matrix-geometric solution [20, 21] to derive the steady-state probability distribution $\mathbf{\Pi}_2$ of the CTMC $\{(N_2(t), J_2(t), S_2(t)), t \geq 0\}$, where $\mathbf{\Pi}_2 = (\boldsymbol{\pi}_0^*, \boldsymbol{\pi}_1^*, \ldots)$.

First, we set up a matrix quadratic equation as follows:

$$\mathbf{R}^2\mathbf{B}^* + \mathbf{R}\mathbf{A}^* + \mathbf{C}^* = \mathbf{0}. \tag{31}$$

Since $\mathbf{A}^*$ must be nonsingular, from equation (31), we have

$$\mathbf{R}^2\mathbf{B}^*\left(\mathbf{A}^*\right)^{-1} + \mathbf{R} + \mathbf{C}^*\left(\mathbf{A}^*\right)^{-1} = \mathbf{0}. \tag{32}$$

By deducing equation (32), we obtain

$$\mathbf{R} = -\mathbf{R}^2\mathbf{W} - \mathbf{V}, \tag{33}$$

where $\mathbf{W} = \mathbf{B}^*(\mathbf{A}^*)^{-1}$ and $\mathbf{V} = \mathbf{C}^*(\mathbf{A}^*)^{-1}$.

In order to compute the rate matrix $\mathbf{R}$, we present an iteration algorithm in Table 1.

Using the rate matrix $\mathbf{R}$ obtained in Table 1, we further construct a square matrix as follows:

$$B[\mathbf{R}] = \begin{pmatrix} \mathbf{A}_0^* & C_0^* \\ \mathbf{B}_1^* & \mathbf{R}\mathbf{B}^* + \mathbf{A}^* \end{pmatrix}. \tag{34}$$

The steady-state probability distribution vectors $\boldsymbol{\pi}_0^*$ and $\boldsymbol{\pi}_1^*$ satisfy the following equation:

$$\begin{cases} (\boldsymbol{\pi}_0^*, \boldsymbol{\pi}_1^*)B[\mathbf{R}] = 0, \\ \boldsymbol{\pi}_0^*\mathbf{e}_0^* + \boldsymbol{\pi}_1^*(\mathbf{I}-\mathbf{R})^{-1}\mathbf{e}_1^* = 1, \end{cases} \tag{35}$$

where $\mathbf{e}_0^*$ is a $(2m+1)\times 1$ vector and $\mathbf{e}_1^*$ is an $(m+2)\times 1$ vector, respectively, with all elements being equal to 1.

By using the Gauss–Seidel method, we solve equation (35) to obtain $\boldsymbol{\pi}_0^*$ and $\boldsymbol{\pi}_1^*$. Other steady-state probability distribution vectors $\boldsymbol{\pi}_i^*, i = 2, 3, \ldots$, satisfy the matrix-geometric solution form as follows:

$$\boldsymbol{\pi}_i^* = \boldsymbol{\pi}_1^*\mathbf{R}^{i-1}, \quad i \geq 2. \tag{36}$$

Up to this point, we can mathematically give the steady-state probability distribution $\mathbf{\Pi}_2 = (\boldsymbol{\pi}_0^*, \boldsymbol{\pi}_1^*, \ldots)$ of the CTMC $\{(N_2(t), J_2(t), S_2(t)), t \geq 0\}$.

6. Performance Measures

In this section, we present two performance measures of the cloud system: the average latency of requests and the energy saving rate of the system.

The service intensity ρ_2 of the warm PM is given as follows:

$$\rho_2 = \lambda_2\left(\frac{1}{\mu_2} + \frac{1}{\beta_2}\right), \tag{37}$$

where λ_2 is the arrival rate of requests at a warm PM, μ_2 is the service rate of a request on a warm VM, and β_2 is the service rate of a request on the warm RSE.

For the proposed scheme, the service intensity ρ of the system is given as follows:

$$\rho = \max\{\rho_0, \rho_2\}, \tag{38}$$

where ρ_0 is the service intensity of the TSDE.

The necessary and sufficient condition for the system to be stable is $\rho < 1$. We evaluate the average latency of requests and the energy-saving rate of the system under the condition that the service intensity $\rho < 1$.

We define the latency of a request as the time duration from the instant a request arrives at the cloud system to the instant the request is about to receive the service. In this paper, the average latency of requests in the cloud system includes the average latency of requests queueing in the TSDE buffer and the average latency of requests queueing in the hot PM buffer or the warm PM buffer.

In Section 4.1, the average latency $E[W_{\text{dec}}]$ of requests queueing in the TSDE buffer has already been obtained. Next, we need to compute the average latency $E[W_{\text{vm}}]$ of requests queueing in the hot PM buffer or the warm PM buffer.

Using the steady-state probability distribution $\mathbf{\Pi}_1$ of the CTMC $\{(N_1(t), J_1(t), S_1(t)), t \geq 0\}$ given in Section 5.1, the average number $E[N_{\text{hot}}]$ of requests queueing in the hot PM buffer can be given by

$$E[N_{\text{hot}}] = \sum_{i=1}^{L} i \left(\sum_{k=0}^{m-1} \pi_{i,1,k} + \pi_{i,0,m} \right). \tag{39}$$

For the convenience of technique, we tag one of the hot PMs. Based on Little's law, the average latency $E[W_{\text{hot}}]$ of requests queueing in the buffer of the tagged hot PM is obtained as follows:

$$E[W_{\text{hot}}] = \frac{1}{\lambda_1 \left(1 - \sum_{i=0}^{m-1} \pi_{L,1,i} - \pi_{L,0,m}\right)} E[N_{\text{hot}}], \tag{40}$$

where λ_1 is the arrival rate of requests at the tagged hot PM.

We also tag one of the warm PMs. Using the steady-state probability distribution $\mathbf{\Pi}_2$ of the CTMC $\{(N_2(t), J_2(t), S_2(t)), t \geq 0\}$ given in Section 5.2, the average number $E[N_{\text{warm}}]$ of requests queueing in the buffer of the tagged warm PM is given as follows:

$$E[N_{\text{warm}}] = \sum_{i=1}^{l_1} i \left(\pi^*_{i,0,0} + \pi^*_{i,1,m} + \sum_{k=0}^{m-1} \pi^*_{i,2,k} \right). \tag{41}$$

l_1 shown in equation (41) is a sufficiently large number satisfying the following equation:

$$1 - \sum_{i=1}^{l_1} i \left(\pi^*_{i,0,0} + \pi^*_{i,1,m} + \sum_{k=0}^{m-1} \pi^*_{i,2,k} \right) < \varepsilon_1, \tag{42}$$

where ε_1, called a precision factor of the average number of requests in the warm PM buffer, is a number related to the precision of the average number of requests in the warm PM buffer. The smaller the value of ε_1 is, the more precisely the average number of requests in the warm PM buffer will be given.

Accordingly, the average latency $E[W_{\text{warm}}]$ of requests queueing in the tagged warm PM buffer is obtained as follows:

$$E[W_{\text{warm}}] = \frac{1}{\lambda_2} E[N_{\text{warm}}], \tag{43}$$

where λ_2 is the arrival rate of requests at the tagged warm PM.

Combining equations (40) and (43), the average latency $E[W_{\text{vm}}]$ of requests queueing in the hot PM buffer or the warm PM buffer can be obtained as follows:

$$E[W_{\text{vm}}] = qE[W_{\text{hot}}] + (1-q)E[W_{\text{warm}}], \tag{44}$$

where q is the probability that a newly arriving request can be accepted by the hot pool.

In summary, the average latency $E[W]$ of requests queueing in the cloud system can be derived by

$$E[W] = E[W_{\text{vm}}] + E[W_{\text{dec}}]. \tag{45}$$

Since the TSDE and the hot PMs are always running, the energy consumption there is normally constant. The energy-saving rate of the system is therefore measured as the energy conservation per unit time in the warm pool.

When the warm PMs are in the active state, the energy is consumed normally just like in the TSDE and the hot PMs. Let $w, w > 0$, be the energy consumption per second for the warm pool in the active state. Let $w_1, w_1 > 0$, be the energy consumption per second for the warm pool in the sleep state. When the warm PMs are in the sleep state, less energy will be consumed. It is obvious that $w > w_1$.

For a sleeping warm PM, if a sleep period is about to expire, the RSE and the VMs need to monitor the PM buffer. Therefore, additional energy will be consumed. Let $w_2, w_2 > 0$, be the energy consumption for each monitoring. It is noted that additional energy is also consumed when the warm PM wakes up from a sleep state. Let $w_3, w_3 > 0$, be the energy consumption for each wake up.

Therefore, in this paper, energy-saving rate S of the system is given as follows:

$$S = \sum_{i=0}^{l_2} \pi^*_{i,0,0} (w - w_1 - \phi w_2) - \sum_{i=1}^{l_2} \pi^*_{i,0,0} \phi w_3, \tag{46}$$

where ϕ is the sleep parameter of the proposed dynamic sleep mechanism defined in Section 4.3.

l_2, shown in equation (46), is a sufficiently large number, which satisfies the following equation:

$$1 - \sum_{i=1}^{l_2} \pi^*_{i,0,0} < \varepsilon_2, \tag{47}$$

where ε_2, called a precision factor of the energy-saving rate of the system, is a number related to the precision of the energy-saving rate of the system. The smaller the value of ε_2 is, the more precisely the energy saving rate of the system will be given.

7. Numerical Results

To numerically analyze the average latency $E[W]$ of requests and the energy-saving rate S of the system with the proposed scheme, we carry out experiments to provide numerical results based on MATLAB. All the experiments are carried out on a PC configured with Intel(R) Core(TM) i7-4790 CPU @ 3.60 GHz, 8.00 GB RAM, and 500G disk. The parameters set in the experiments are listed in Table 2.

7.1. Numerical Results for the Average Latency of Requests. Figure 2 illustrates the change trend for the average latency $E[W]$ of requests with the sleep parameter ϕ for a different number n_h of the hot PMs and a different number n_w of the warm PMs.

Table 2: Parameter settings in the experiments.

Parameters	Values
The maximum number m of VMs deployed on one PM	$m = 3$
Arrival rate λ_0 of requests at the TSDE	$\lambda_0 = 4.8\,\mathrm{s}^{-1}$
Service rate δ of a request on the TSDE	$\delta = 5\,\mathrm{s}^{-1}$
Service rate β_1 of a request on the hot RSE	$\beta_1 = 1\,\mathrm{s}^{-1}$
Energy consumption ω per second for the warm pool in the active state	$\omega = 2\,\mathrm{mJ}$
Energy consumption ω_1 per second for the warm pool in the sleep state	$\omega_1 = 0.3\,\mathrm{mJ}$
Additional energy consumption ω_2 for each monitoring	$\omega_2 = 0.1\,\mathrm{mJ}$
Additional energy consumption ω_3 for each wake up	$\omega_3 = 0.2\,\mathrm{mJ}$
Precision factor ε_1 of the average number of requests in the warm PM buffer	$\varepsilon_1 = 10^{-15}$
Precision factor ε_2 of the energy-saving rate of the system	$\varepsilon_2 = 10^{-15}$

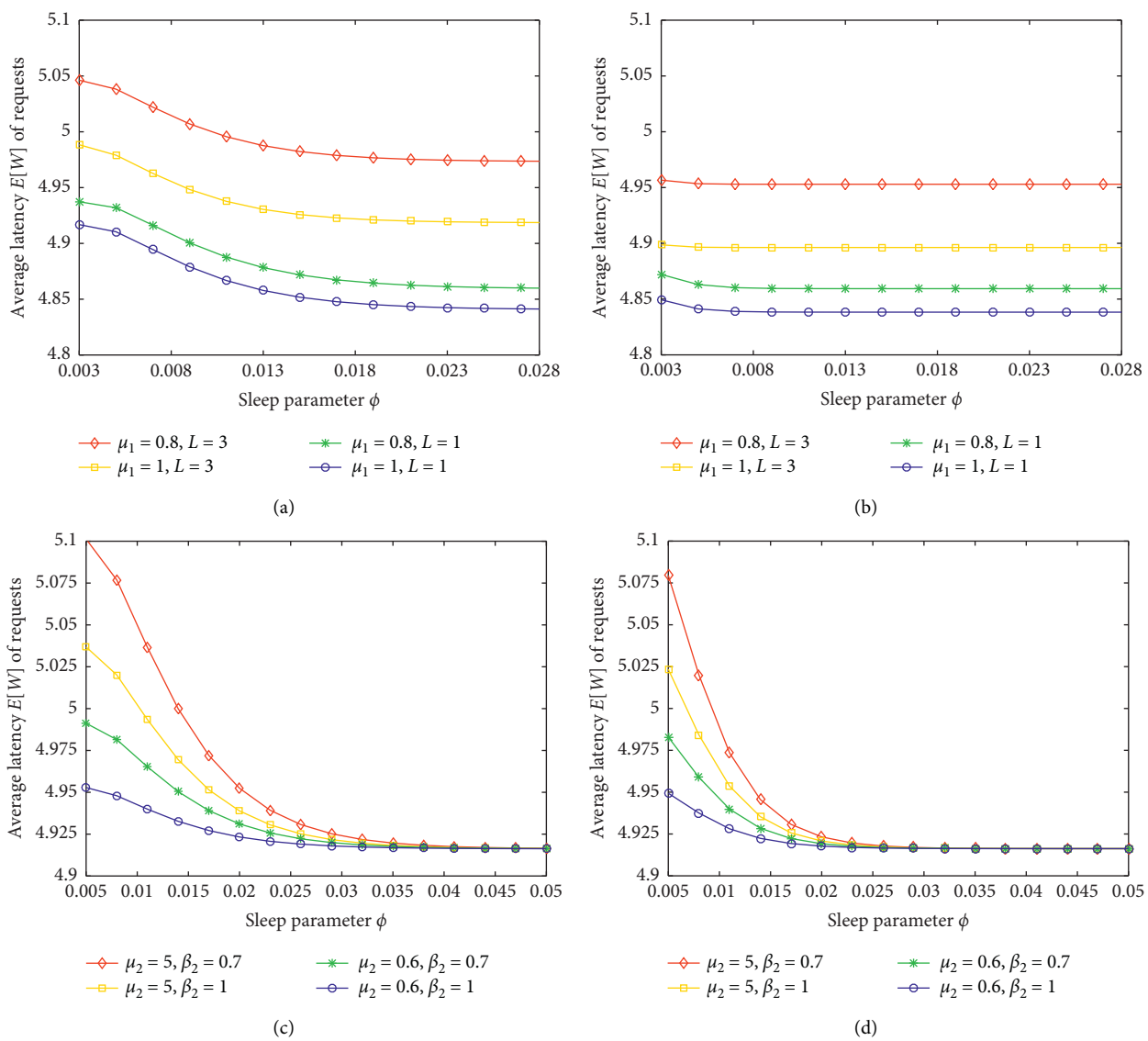


Figure 2: Average latency $E[W]$ of requests vs. sleep parameter ϕ. (a) $n_h = 2, n_w = 6$. (b) $n_h = 3, n_w = 6$. (c) $n_h = 2, n_w = 4$. (d) $n_h = 2, n_w = 6$.

In Figures 2(a) and 2(b), we show the average latency $E[W]$ of requests for the different service rates μ_1 of a request on a hot VM and the different capacities L of a hot PM buffer, respectively. In Figures 2(c) and 2(d), we show the average latency $E[W]$ of requests for the different service rates μ_2 of a request on a warm VM and the different service rates β_2 of a request on a warm RSE, respectively.

From Figure 2, we notice that, as the sleep parameter ϕ increases, the average latency $E[W]$ of requests firstly decreases accordingly and then tends to be fixed.

In the stage of the smaller sleep parameter ϕ, a newly arriving request has to wait for a longer time in the buffer of a sleeping warm PM. As the sleep parameter grows, the waiting time of a request in the warm PM buffer gets shorter. Therefore, the average latency $E[W]$ of requests shows a downtrend. This implies that the influence of the sleep mechanism on the response performance of the system is greater in the case of a smaller sleep parameter.

When the sleep parameter ϕ gets larger and grows to a certain value, the time length of a sleep period is close to zero. Therefore, a warm PM has little chance to go to sleep. As a result, the average latency $E[W]$ of requests tends to be fixed as the sleep parameter increases. This implies that the proposed sleep mechanism has little effect on the response performance of the system when the sleep parameter is large enough.

For the same sleep parameters ϕ in both Figures 2(a) and 2(b), we notice that the average latency $E[W]$ of requests goes up as the capacity L of a hot PM buffer increases. The larger a hot PM's buffer capacity is, the longer the requests wait in the hot PM buffer. This gives rise to an increase in the average latency of requests. We also notice that, as the service rate μ_1 of a request on a hot VM increases, the average latency of requests gets reduced. The higher the service rate is, the less time a request occupies the hot VM. Therefore, the average latency of requests shows a downtrend.

Comparing Figures 2(a) with 2(b), we find that, for the same capacity L of a hot PM buffer, the same service rate μ_1 of a request on a hot VM, and the same sleep parameter ϕ, as the number n_h of the hot PMs increases, the average latency $E[W]$ of requests becomes lower. The more the PMs are deployed in the hot pool, the earlier the requests arrive at the hot pool receive service. Therefore, the average latency of requests shows a downtrend. In addition, we also find that when the sleep parameter is smaller, the downtrend for the average latency of requests gets slighter as the number of the hot PMs increases. This implies that the more the PMs are deployed in the hot pool, the weaker the influence of the sleep mechanism on the response performance of the system becomes.

For the same sleep parameters ϕ in both Figures 2(c) and 2(d), we observe that the average latency $E[W]$ of requests rises up as the service rate μ_2 of a request on a warm VM increases. When the service rate of a request on a warm VM is higher, the probability of the warm RSE and the warm VMs being idle is greater. Therefore, the warm PM is more likely to be asleep, which causes the request to wait longer in the warm PM buffer. Accordingly, the average latency of requests gets larger. We also observe that, as the service rate β_2 of a request on a warm RSE increases, the average latency of requests is reduced. The higher the service rate of a request on a warm RSE is, the less time a request occupies the warm RSE. This leads to a lower average latency of requests.

Comparing Figures 2(c) with 2(d), we find that, for the same service rate μ_2 of a request on a warm VM, the same service rate β_2 of a request on a warm RSE, and the same sleep parameter ϕ, a greater number n_w of the warm PMs gives rise to a lower average latency $E[W]$ of requests. The more the PMs are deployed in the warm pool, the earlier the requests arrive at the warm pool receive service. Therefore, the average latency of requests gets reduced. In addition, we also find that the downtrend for the average latency of requests becomes sharper as the number of the warm PMs increases in the case of a smaller sleep parameter. This implies that the more the PMs are deployed in the warm pool, the stronger the influence of the sleep mechanism on the response performance of the system becomes.

7.2. Numerical Results for the Energy-Saving Rate of the System. Figure 3 shows the trends for the energy-saving rate S of the system with the sleep parameter ϕ for a different number n_h of the hot PMs and a different number n_w of the warm PMs.

In Figures 3(a) and 3(b), we show the energy-saving rate S of the system for the different service rates μ_1 of a request on a hot VM and the different capacities L of a hot PM buffer, respectively. In Figures 3(c) and 3(d), we show the energy-saving rate S of the system for the different service rates μ_2 of a request on a warm VM and the different service rates β_2 of a request on a warm RSE, respectively.

From Figure 3, we notice that, as the sleep parameter ϕ increases, the energy-saving rate S of the system shows a downward trend. In the stage of a smaller sleep parameter, the energy-saving rate of the system is initially higher. The smaller the sleep parameter is, the longer the time length of a sleep period is. For this case, frequent listening and waking up of the warm RSE and the warm VMs are avoided so that additional energy use is reduced.

As the sleep parameter ϕ gets larger, the energy-saving rate S of the system decreases. The larger the sleep parameter is, the shorter the time length of a sleep period is. For this case, the warm RSE and the warm VMs listen to the buffer and wake up from sleep frequently. This causes additional energy consumption.

For the same sleep parameters ϕ in both Figures 3(a) and 3(b), we notice that the energy-saving rate S of the system goes up as the capacity L of a hot PM buffer or the service rate μ_1 of a request on a hot VM increases. The larger the capacity of a hot PM buffer is, the more requests the hot PM can accept. The higher the service rate of a request on a hot VM is, the less time a request occupies the hot VM. Therefore, the processing capability of a hot PM becomes stronger. For this case, fewer requests are allocated to the warm pool so that the warm PMs are more likely to be in the sleep state. Accordingly, the energy-saving rate of the system is greater.

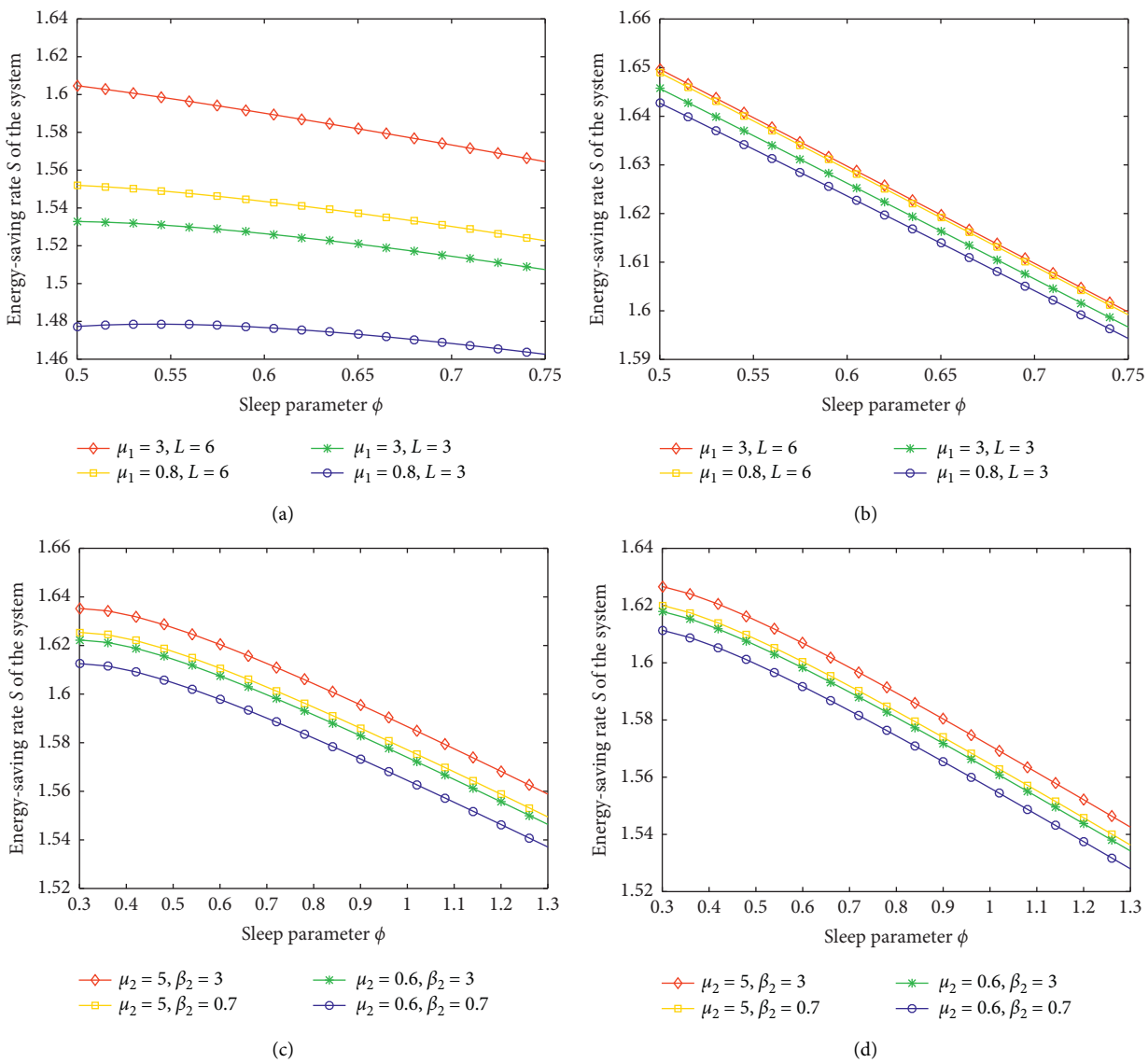


Figure 3: Energy-saving rate S of the system vs. sleep parameter ϕ. (a) $n_h = 4, n_w = 2$. (b) $n_h = 5, n_w = 2$. (c) $n_h = 4, n_w = 2$. (d) $n_h = 4, n_w = 3$.

Comparing Figures 3(a) with 3(b), we find that, for the same capacity L of a hot PM buffer, the same service rate μ_1 of a request on a hot VM, and the same sleep parameter ϕ, a larger number n_h of the hot PMs leads to a higher energy-saving rate S of the system. The more the PMs are deployed in the hot pool, the stronger the processing capability of the hot pool is. For this case, fewer requests are allocated to the warm pool so that the warm PM is more likely to be in the sleep state. This causes an increase in the energy-saving rate of the system. In addition, we also find that the more the PMs are deployed in the hot pool, the closer the energy-saving rates of the system with different capacities of a hot PM buffer and different service rates of a request on a hot VM are. This implies that the capacity of the hot PM buffer and the service rate of a request on a hot VM have less influence on the energy-saving rate of the system as the number of the hot PMs rises.

For the same sleep parameters ϕ in both Figures 3(c) and 3(d), we observe that the energy-saving rate S of the system rises as the service rate μ_2 of a request on a warm VM or service rate β_2 of a request on a warm RSE grows. The higher the service rate of a request on a warm RSE is, the less time a request occupies the warm RSE. The higher the service rate of a request on a warm VM is, the less time a request occupies the warm VM. For this case, the warm PM is more likely to become idle and enter a sleep period. Therefore, the energy-saving rate of the system shows a growth trend.

Comparing Figures 3(c) with 3(d), we find that, for the same service rate μ_2 of a request on a warm VM, the same service rate β_2 of a request on a warm RSE, and the same sleep parameter ϕ, a greater number n_w of the warm PMs leads to a higher energy-saving rate S of the system. The more the PMs are deployed in the warm pool, the stronger the processing capability of the warm pool is. For this case,

Table 3: Improved Salp Swarm Algorithm proposed to obtain the optimal sleep parameters.

Step 1. Initialize the number N of salps, maximum iteration $T_{\max}$ for each salp's position, initial inertia weight w_s, inertia weight w_m at the maximum iteration, upper search boundary u_b, and lower Search boundary l_b.

Step 2. Initialize the position $\phi_i\,(i = 1, 2, \ldots, N)$ for each salp by using a chaotic equation:
$\phi_1 = \text{rand}.$
% rand represents random numbers that obey uniform distribution between $(0, 1)$.%
for $i = 2$: N
$\phi_i = \xi\phi_{i-1}(1 - \phi_{i-1})$
% ξ is a given real parameter.%
endfor
for $i = 1$: N
$\phi_i = l_b + \phi_i(u_b - l_b)$
endfor

Step 3. Calculate the fitness $F_i\,(i = 1, 2, \ldots, N)$ for each salp:
$F_i = F(\phi_i) = f_1E[W] - f_2S.$

Step 4. Select the best position ϕ^* among all the salps as the source food and calculate the fitness F^* of the source food:
$\phi^* = \text{argmin}_{i\in\{1,\ldots,N\}}F_i,$
$F^* = F(\phi^*) = f_1E[W] - f_2S.$

Step 5. Set the initial number of iterations as $t = 1$.

Step 6. Update the coefficient c_1 and inertia weight $w(t)$ with a nonlinear decreasing function:
$c_1 = 2\exp(-4t/T_{\max}),$
$w(t) = (w_s - w_m)((T_{\max} + t)/t) + w_s.$

Step 7. Update the position ϕ_i and calculate the fitness $F_i\,(i = 1, 2, \ldots, N-1)$ for other salps.
for $i = 1$: $N - 1$
if $i \le \lfloor (N-1)/2 \rfloor$
$c_2 = \text{rand},\ c_3 = \text{rand}$
$$\phi_i = \begin{cases} F + c_1((u_b - l_b)c_2 + l_b), & c_3 \ge 0 \\ F - c_1((u_b - l_b)c_2 + l_b), & c_3 < 0 \end{cases}$$
else
$\phi_i = 1/2(\phi_i + w(t)\phi_{i-1})$
end if
$F_i = F(\phi_i) = f_1E[W] - f_2S$
end for

Step 8. Update the source food ϕ^* and calculate the fitness F^* of the source food:
$\phi^* = \text{argmin}_{i\in\{1,2,\ldots,N\}}F_i,$
$F^* = \min_{i\in\{1,2,\ldots,N\}}F_i.$

Step 9. Check the number of iterations:
if $t < T_{\max}$
$t = t + 1$, go to **Step 6**
endif

Step 10. Output the optimal sleep parameter ϕ^* and the minimum cost F^*.

Table 4: Optimal sleep parameters of the proposed scheme.

L	μ_2	ϕ^*	F^*
3	0.6	1.7660	25.5421
	1.1	1.9225	24.4361
	5.0	2.0739	24.3177
5	0.6	1.7771	25.1119
	1.1	1.9197	25.0055
	5.0	2.1480	24.8868
10	0.6	1.7640	26.5505
	1.1	1.9243	26.4440
	5.0	2.0710	26.3253

the probability of a warm PM being idle is higher, so the warm PM is more likely to be in the sleep state. This leads to an increase in the energy-saving rate of the system. In addition, we also find that the more the PMs are deployed in the warm pool, the closer the energy-saving rates of the system with different service rates of a request on a warm VM and different service rates of a request on a warm RSE are. This implies that when the number of warm PMs is greater, the energy-saving rate of the system is rarely affected by the service rate of a request on a warm VM and the service rate of a request on a warm RSE.

8. Performance Optimization

Based on the numerical results given in Section 7, we find that, with an increase in the sleep parameter ϕ, the average latency $E[W]$ of requests shows a downward trend, and the energy-saving rate S of the system also decrease. This indicates that when the sleep parameter tends to infinity, the average latency of requests will be minimized, and the energy-saving rate will be close to zero. Obviously, in this case, the energy-saving mechanism will not work at all. Conversely, when the sleep parameter tends to zero, the energy-

saving rate will be maximized, and the average latency of requests will become too great to be accepted. In this case, the cloud system cannot provide service normally. How to optimally set the sleep parameter is an important issue in any energy-efficient resource management scheme. In this paper, the criterion for optimization is to balance different performance measures. To do this, we combine the average latency of requests and the energy-saving rate of the system and construct a cost function $F(\phi)$ as follows:

$$F(\phi) = f_1 E[W] - f_2 S, \tag{48}$$

where f_1 and f_2 are the influencing factors for the average latency $E[W]$ of requests and the energy-saving rate S of the system, respectively, in regards to the cost function in the system parameters. It is noted that the higher the cloud user's demand for the response performance is, the larger the parameter f_1 should be set; the higher the cloud provider's demand for the energy efficiency is, the larger the parameter f_2 should be set.

We note that it is difficult to express the average latency $E[W]$ of requests and the energy-saving rate S of the system in closed forms. Therefore, we cannot easily figure out the monotonicity of the cost function. For minimizing the system cost $F(\phi)$ and optimizing the sleep parameter ϕ, we introduce a swarm-based algorithm: SSA.

SSA is an intelligent searching optimization algorithm inspired by the swarming behaviour of salps. In 2017, Seyedali et al. first established a mathematical model of salp chains and presented the SSA to settle many optimization problems [22]. SSA has only one main controlling parameter, so it is simple and easy to implement. However, like other swarm-based algorithms, SSA has the insufficiencies of low convergence precision and slow convergence speed when dealing with high-dimensional complex optimization problems [23]. In the classical SSA optimization process, global exploration and local exploitation are a pair of contradictions. If this process is out of balance, the algorithm easily falls into local optimization and leads to convergence stagnation. Consequently, in this paper, we present an improved SSA by introducing logistic chaotic initialization and adaptive inertia weight [24]. We call this improved SSA LA-SSA.

In this LA-SSA, we firstly adopt a logistic chaotic mapping method to generate the initial salp population. This enhances the diversity of the initial individuals and improves the convergence speed of the algorithm in the early stage. Secondly, we introduce an adaptive inertia weight to update the follower position. The inertia weight reflects the ability of the follower to inherit the salp position from the previous one. If the position of the follower is the locally optimal solution, it is easy to fall into the local optimum and result in convergence stagnation for SSA. Moreover, to improve the convergence precision and help SSA break out of the local optimum, in this paper, the inertia weight of linear decline is introduced, which determines the degree of influence of the previous individual on the current individual. This means salp individuals have strong global convergence capacity and relatively accurate results can be obtained in the later stage.

Table 3 shows the main steps of the LA-SSA.

In addition to utilize the parameters in Table 2, we set $f_1 = 5$, $f_2 = 1$, $N = 50$, $T_{\max} = 100$, $\xi = 4$, $w_s = 0.9$, $w_e = 0.4$, $ub = 5$, and $lb = 0$ as an example in the LA-SSA to optimize the dynamic energy-efficient resource management scheme proposed in this paper. For different capacities L of a hot PM buffer and different service rates μ_2 of a request on a warm VM, we produce the optimal sleep parameter ϕ^* and the minimum cost F^* in Table 4.

From Table 4, we observe that, for the same capacity L of a hot PM buffer, the optimal sleep parameter ϕ^* maintains an upward trend as the service rate μ_2 of a request on a warm VM increases. In contrast, the minimum cost F^* shows a downward trend when the service rate μ_2 of a request on a warm VM goes up.

9. Summary

Considering large amounts of energy consumption generated by cloud data centers, we proposed a dynamic energy-efficient resource management scheme under a multitier cloud architecture. In order to improve the energy efficiency while maintaining the quality of experience for cloud users, we grouped the PMs into different resource pools and introduced a synchronous sleep mechanism to the warm pool. By establishing a Markov chain, we obtained the average latency of requests and the energy-saving rate of the system. In addition, we provided numerical results to study the influence of the sleep mechanism on the system performance. To balance different performance measures, we constructed a system cost function. Moreover, we presented an improved SSA to obtain the optimal sleep parameters and the minimum costs.

In subsequent work, we consider to study energy conservation in cloud systems with heterogeneous cloud users and PMs. Furthermore, we consider to analyze the system models by considering any general stochastic processes, such as Markovian Arrival Process (MAP) and Markovian Service Process (MSP).

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by National Natural Science Foundation (nos. 61872311 and 61973261), China, and was supported in part by MEXT and JSPS KAKENHI , Grant JP17H01825 Japan.

References

[1] Y. Gao, H. Guan, Z. Qi, T. Song, F. Huan, and L. Liu, "Service level agreement based energy-efficient resource management in cloud data centers," *Computers and Electrical Engineering*, vol. 40, no. 5, pp. 1621–1633, 2014.

[2] Y. Hao, J. Cao, T. Ma, and S. Ji, "Adaptive energy-aware scheduling method in a meteorological cloud," *Future Generation Computer Systems*, vol. 101, pp. 1142–1157, 2019.

[3] J. Luo, X. Li, and M. Chen, "Hybrid shuffled frog leaping algorithm for energy-efficient dynamic consolidation of virtual machines in cloud data centers," *Expert Systems with Applications*, vol. 41, no. 13, pp. 5804–5816, 2014.

[4] K. Karthiban and J. Raj, "An efficient green computing fair resource allocation in cloud computing using modified deep reinforcement learning algorithm," *Soft Computing*, vol. 24, no. 3, pp. 14933–14942, 2020.

[5] Y. Hao, J. Cao, Q. Wang, and J. Du, "Energy-aware scheduling in edge computing with a clustering method," *Future Generation Computer Systems*, vol. 117, pp. 259–272, 2021.

[6] A. Auday, W. Itani, R. Zantout, and A. Zekri, "Type-aware virtual machine management for energy efficient cloud data centers," *Sustainable Computing: Informatics and Systems*, vol. 19, pp. 185–203, 2018.

[7] M. Zakarya and L. Gillam, "An energy aware cost recovery approach for virtual machine migration," *in Proc. International Conference on Economics of Grids, Clouds, Systems, and Services*, pp. 175–190, 2016.

[8] C. Ghribi, M. Hadji, and D. Zeghlache, "Energy efficient VM scheduling for cloud data centers: exact allocation and migration algorithms," in *Proceedings of the 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing*, pp. 671–678, Delft, Netherlands, 2013.

[9] N. Sharma and R. Guddeti, "Multi-objective energy efficient virtual machines allocation at the cloud data center," *IEEE Transactions on Services Computing*, vol. 12, no. 5, pp. 158–171, 2016.

[10] S. Jin, X. Qie, W. Zhao, W. Yue, and Y. Takahashi, "A clustered virtual machine allocation strategy based on a sleep-mode with wake-up threshold in a cloud environment," *Annals of Operations Research*, vol. 293, no. 1, pp. 193–212, 2019.

[11] F. Farahnakian, P. Liljeberg, and J. Plosila, "Energy-efficient virtual machines consolidation in cloud data centers using reinforcement learning," in *Proceedings of the 22nd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing*, pp. 500–507, Turin, Italy, 2014.

[12] Sridharshini and V. Sivagami, "Energy-aware scheduling using workload consolidation techniques in cloud environment," *International Journal of Computer Science and Engineering Communications*, vol. 3, no. 3, pp. 1141–1148, 2015.

[13] H. Mora, F. J. Mora Gimeno, M. T. Signes-Pont, and B. Volckaert, "Multilayer architecture model for mobile cloud computing paradigm," *Complexity*, vol. 2019, no. 2, 13 pages, Article ID 3951495, 2019.

[14] M. Usman, A. Samad, Ismail, H. Chizari, and A. Aliyu, "Energy-efficient virtual machine allocation technique using interior search algorithm for cloud data center," in *Proceedings of the 6th ICT International Student Project Conference*, pp. 1–4, Johor, Malaysia, 2017.

[15] A. Beloglazov, "Energy-efficient management of virtual machines in data centers for cloud computing," Ph.D. dissertation, The University of Melbourne, Melbourne, Australia, 2013.

[16] H. Zhu, W. Hai, and X. Liao, "Task scheduling model and virtual machine deployment algorithm for energy consumption optimization in cloud computing," *Systems Engineering-Theory and Practice*, vol. 36, no. 3, pp. 768–778, 2016.

[17] R. Ghosh, F. Longo, V. K. Naik, and K. S. Trivedi, "Modeling and performance analysis of large scale IaaS Clouds," *Future Generation Computer Systems*, vol. 29, no. 5, pp. 1216–1234, 2013.

[18] V. Paxson and S. Floyd, "Wide area traffic: the failure of Poisson modeling," *IEEE/ACM Transactions on Networking*, vol. 3, no. 3, pp. 226–244, 1995.

[19] W. Stewart, *Probability, Markov Chains, Queues, and Simulation*, Princeton University Press, Princeton, NJ, USA, 2009.

[20] G. Latouche and V. Ramaswami, "Introduction to matrix analytic methods in stochastic modeling," Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1999.

[21] M. Neuts, *Matrix-Geometric Solutions in Stochastic Models*, Johns Hopkins University Press, Baltimore, MD, USA, 1984.

[22] M. Seyedali, H. Amir, Z. Seyedeh, S. Shahrzad, F. Hossams, and M. Seyed, "Salp swarm algorithm: a bio-inspired optimizer for engineering design problems," *Advances in Engineering Software*, vol. 114, pp. 163–191, 2017.

[23] J. Wu, R. Nan, and L. Chen, "Improved salp swarm algorithm based on weight factor and adaptive mutation," *Journal of Experimental and Theoretical Artificial Intelligence*, vol. 31, no. 3, pp. 1–23, 2019.

[24] P. An, "Particle swarm optimization algorithm based on chaotic theory and adaptive inertia weight," *Journal of Nanoelectronics and Optoelectronics*, vol. 12, no. 4, pp. 404–408, 2017.

Hindawi
Complexity
Volume 2021, Article ID 6674639, 10 pages
https://doi.org/10.1155/2021/6674639

Research Article

KPDR : An Effective Method of Privacy Protection

Zihao Shen,[1,2] Wei Zhen,[1] Pengfei Li,[1] Hui Wang,[1] Kun Liu,[1] and Peiqian Liu[1]

[1]*School of Computer Science and Technology, Henan Polytechnic University, Jiao'zuo 454000, China*
[2]*College of Computer Science and Technology, Jilin University, Chang'chun 130012, China*

Correspondence should be addressed to Hui Wang; wanghui_jsj@foxmail.com

Received 12 November 2020; Revised 28 December 2020; Accepted 29 January 2021; Published 9 February 2021

Academic Editor: Yongsheng Hao

To solve the problem of user privacy disclosure caused by attacks on anonymous areas in spatial generalization privacy protection methods, a K and P Dirichlet Retrieval (KPDR) method based on k-anonymity mechanism is proposed. First, the Dirichlet graph model is introduced, the same kind of information points is analyzed by using the characteristics of Dirichlet graph, and the anonymous set of users is generated and sent to LBS server. Second, the relationship matrix is generated, and the proximity relationship between the user position and the target information point is obtained by calculation. Then, the private information retrieval model is applied to ensure the privacy of users' target information points. Finally, the experimental results show that the KPDR method not only satisfies the diversity of $l(3/4)$, but also increases the anonymous space, reduces the communication overhead, ensures the anonymous success rate of users, and effectively prevents the disclosure of user privacy.

1. Introduction

Thanks to the emergence of mobile terminal equipment and the rapid development of location service systems, great changes have occurred in our lives, and people can buy their favourite products without leaving home. There are Taobao for dressing, Meituan for eating, Flying Pig for lodging, Didi for travelling, and strips for travelling. People can get services anytime and anywhere through various apps [1], all of which are derived from the rapid development of Location-Based Services (LBS). According to statistics [2], the global market share of LBS and Real-Time Location Systems (RTLS) will increase from 11.36 billion in 2015 to 54.95 billion US dollars in 2020, and the Compound Annual Growth (CAGR) will be 37.1%.

LBS [3, 4] refers to providing various value-added services for mobile users based on the location information of mobile devices and the information transmission of communication networks. However, as people's demand for services increases, location service providers (LSP) may leak users' privacy to criminals for their own benefit, which will threat users' property and personal safety [5]. Therefore, protecting user privacy while providing users with convenient services has become an urgent problem to be solved [6].

In the aspect of location privacy protection, spatial generalization technology based on k-anonymity has always been a hot spot for scholars. Its core idea is to generalize the real location of users and ensure that there are at least K-1 users in the anonymous area (ASR), so that LSP cannot distinguish real users from K users. At present, there are many researches on privacy protection technology based on k-anonymity [7, 8]. For example, Li et al. [9] introduced a credit mechanism on the basis of k-anonymity and set a threshold for users. When the user's credit is higher than this threshold, they can participate in the formation of k-anonymity to obtain privacy protection.

To resist the attack against ASR, Zheng et al. [10] proposed an outlier elimination clustering algorithm based on k-anonymous model; the algorithm optimized the distribution of users in anonymous groups by taking anonymous groups as the center instead of users' positions, but the anonymous areas formed were larger than the actual needs, and in many cases, the probability of attackers identifying query requesters was much higher than $1/k$. Wang et al. [11] proposed differential private K-valued method (DPKA) combined with the concept of difference privacy and k-anonymity and proposed a method for its realization. This method, however, does not consider the effect of $l(3/4)$

diversity on k-anonymity, which is vulnerable to continuous query attacks. Literature [12] proposed a k-anonymity algorithm based on the analytic hierarchy process; in the clustering process, the method always selects the record with the smallest distance to add and individually controls the clustering according to the K value to achieve the equivalent class, but when the k-anonymity area formed in densely populated places is small, the attacker can still infer the approximate location of the user. It can be seen that the process of generating anonymous regions from the anonymous space is the most vulnerable to attack by attackers.

To solve the above problems, this paper proposes a privacy protection method of KPDR based on Dirichlet graph model, which can protect users' privacy from location and query. In the aspect of protecting location privacy, k-anonymity random location hiding method based on Dirichlet graph model is adopted to ensure the security of ASR and satisfy the diversity of location $l(3/4)$. Therefore, the probability of users being identified by attackers is less than $1/k$. In terms of protecting query privacy, due to the particularity and unreliability of LBS, attackers have a high probability of inferring the user's sensitive information according to the user's query points and causing privacy leakage. In this paper, the private information retrieval (PIR) technology with relatively high security [13, 14] is adopted, which can ensure that the trusted third-party server (TTPS) can securely retrieve the desired data from the untrusted LBS server and effectively prevent the privacy disclosure caused by the attack of LBS.

2. Propaedeutics

2.1. System Architecture. With the change of problem background and attack model, location privacy will continue to face new challenges. For example, LBS servers are vulnerable to attacks, and the risk of sensitive attribute disclosure exists objectively on the premise that LBS operators cannot be fully trusted. To ensure user privacy and service quality, this method introduces a trusted third-party server, and a trusted third-party center structure is composed of a mobile terminal, a trusted third-party server, and an LBS server, as shown in Figure 1. In the KPDR privacy protection method, the trusted third-party server and the LBS server jointly maintain a set of information points. The mobile terminal sends the query request information to the trusted third-party server, which generates Dirichlet graph according to the user query request information and its own cached information points and selects the false positions of the virtual user and the current user in $K-1$ D blocks according to the established rules to form a user anonymous set and send it to the LBS server. After obtaining the user set, the LBS server generates a relationship matrix according to the proximity relationship between the user target information points and k users. After that, the trusted third-party server retrieves the target information from the relational matrix and returns it to the user, which is a complete request. In the whole process of privacy protection, the data centralization is completed by using the trusted third party as the total carrier, and the privacy security requirements of users can be met as long as the security of the trusted third-party server is ensured.

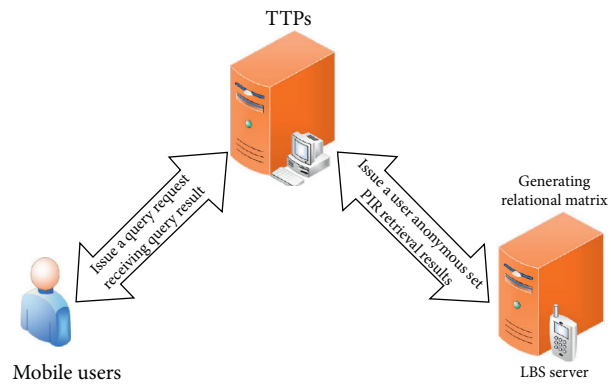


Figure 1: Communication model.

2.2. Related Definitions

Definition 1. Point of information (POI) is as follows:

$$\mathrm{POI}\left(S_{\mathrm{ig}}, C_{\mathrm{la}}, l_{\mathrm{at}}\right), \tag{1}$$

where S_{ig} stands for the unique name identification of the information point, C_{la} represents the category of the information point, and l_{at} represents the coordinate information of the information point, and the introduction of information points is to enhance the ability to query and describe the user's location and improve the query efficiency.

Definition 2. Dirichlet graph is as follows: let set $D = \{D_1, D_2, D_3, \ldots, D_n\}$ be a set of n information points on the plane, where

$$\forall D_i, D_j \in D, E_{\mathrm{ve}} \in V_i \mid \mathrm{S}\left(E_{ve}, D_i\right) < \mathrm{S}\left(E_{ve}, D_j\right), \quad i \neq j, \tag{2}$$

is the Dirichlet diagram, in which $\mathrm{S}(E_{ve}, D_j)$ is the Euclidean distance from point E_{ve} to point D_i, E_{ve} is any point in D block, and V_i is any single polygon in Dirichlet graph, which is called D-block as shown in Figure 2.

The feature of Dirichlet graph is that there is a focus in each D block, and the distance from the inner point of each D block to this focus is smaller than that from other D blocks, such as $S(E_{ve}, D_1) < S(E_{ve}, D_j)$, $j \neq 1$. The distance from the point on the boundary of block D to the focus that generates this boundary is equal; by using the characteristics of Dirichlet graph, the trusted third-party server can find the nearest information point to the user more quickly after receiving the user query request, which is more efficient than$K(3/4)$NNalgorithm.

Definition 3. Client request is as follows: $U_{\mathrm{client}}(I_{\mathrm{dent}}, L_{\mathrm{oc}}, C_{\mathrm{la}}, U_{\mathrm{time}}, \lambda)$ represents the request information sent by the user's mobile terminal. The field I_{dent} represents the unique identification number of the user; the field L_{oc} represents the position when the user initiates the

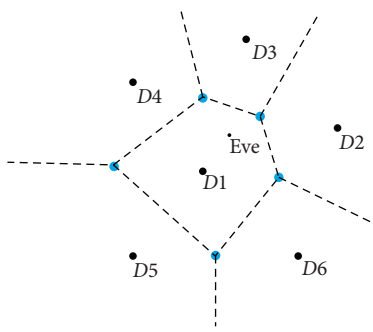


Figure 2: Block D in the Dirichlet diagram.

request; C_{la}stands for POI category; U_{time} represents the time point when the user sends the request; λrepresents a large odd prime number.

Definition 4. User anonymous set is as follows: Users$(ID_{\text{gate}}, C_{\text{la}}, C_{\text{time}}, L, X_{\text{inf}})$, where ID_{gate} represents the unique identification number of user anonymous set; C_{la}represents the information point category of user anonymous set; C_{time} indicates the time when the anonymous set of users is sent out; the query set $X_{\text{inf}} = \{x_1, x_2, \ldots, x_u, \ldots, x_k\}$ has quadratic residuals of K−1 modular$\lambda$and quadratic nonresidual x_u of one modularλ; the location set $L = \{L_1, L_2, \ldots, L_u, \ldots, L_k\}$ represents the location of each user in the user anonymous set, where L_u represents the random false location in the D block to which the real user belongs. The position parameter L must satisfy the following equation:

$$\exists L_i \in D_i, \forall L_j \notin L_i, \quad i \neq j, \tag{3}$$

where parameter D_i represents the D block in the Dirichlet diagram. In this way, when the trusted third-party server sends the user anonymity set to the LBS server, the user location in the user anonymity set must be randomly selected from different D blocks.

3. Privacy Protection Method of KPDR Based on K-Anonymous

3.1. Location Hiding Algorithm Based on Dirichlet Graph Model

3.1.1. Dirichlet Construction Based on POI. Before the privacy protection process starts, the LBS server keeps the Dirichlet graph based on the same category information points in TTPS and LBS servers synchronously. As shown in Algorithm 1, taking the information point as the base point, the Delaunay Triangulation Algorithm is used to generate the triangulation and then determine the circumscribed circle center of each triangle in the triangulation and finally connect the adjacent circle centers to construct the Dirichlet diagram model.

From the above algorithm, we can see that the algorithm complexity calculation of Algorithm 1 is divided into four parts. The first part is to construct Delaunay triangular network with the complexity of $O(n^2)$. The second part computes the center of the triangle peripheral circle, and the complexity is $O(n)$. In the third part, the complexity of finding triangles with three adjacent sides is $3O(n)$. The fourth part draws the Dirichlet diagram; the complexity is $O(n)$. Therefore, the algorithm complexity of generating Dirichlet graph focusing on POI of the same kind is $O(n^2) + O(n) + 3O(n) + O(n) = O(n^2)$.

As shown in Figure 3, each polygon represents a D block, and the focus of each D block is the information point. When the trusted third-party receives the request sent by the user, it will divide the corresponding Dirichlet graph according to the position L_{oc} in the mobile terminal request U_{client}.

3.1.2. The Processing of TTP Server to the User Sending Service Request U_{client}. When the trusted third-party server receives a user request for U_{client}, it will first determine whether the request is initiated by the same user again according to the unique user identification number ID_{gate} in U_{client}. If it is the first time, the trusted third-party server will determine the rule, generate an anonymous set of users, and send it to the LBS server. If it is not initiated for the first time, the trusted third-party server will calculate according to the two positions L in the latest service request information U_{client} sent by the user; when the latest user position L has left the last D block, it will regenerate the latest user anonymous set and send it to the LBS server. Updating user anonymous sets in time can effectively prevent joint attacks and location inference attacks. If the location L sent by the user multiple times is the same as the location sent for the first time, then the trusted third-party server will form a time series set according to the time U_{client} initiated in U_{client} each time the user requests $U_{\text{client}} = \{t_1, t_2, \ldots, t_n\}$; calculate the value of ε:

$$\varepsilon(U_{\text{time}}) = \left(\sum_{k=1}^{n} \left| t_{k+1} - t_k \right|^2 \right)^{(1/2)}. \tag{4}$$

Supposing the normal load of the trusted third-party server is∂, when $\partial \geq \varepsilon$, the trusted third-party server will regenerate the user anonymity set and send it to the LBS server; otherwise, it will directly send the last generated user anonymity set. This method reduces the load pressure of trusted third-party servers to a certain extent. At the same time, when the server load is low, updating the user anonymity set with high frequency can effectively resist continuous query attacks and associated attacks and enhance anonymity.

3.1.3. Generation Rules of TTP Server for User Anonymous Set Users. After receiving the user request, the trusted third-party server will first save all information in U_{client} according to the identification number ID_{gate} in U_{client} requested by the user and then find the nearest information point from the server cache according to the real location L of the user in U_{client} to generate the Dirichlet diagram of the same information point. Then, according to the location L of the real user, a fake location point is randomly selected from the D block to which it belongs to replace the real user location, and K−1 fake location points are selected from different D

Input: POI List
Output: Dirichlet diagram focusing on POI

```
Initialize the triangle list
Determine the super triangle
Add super triangle vertices to the end of the POI List
Add the super triangle to the triangle list
for each sample point in the POI List
  Initialize the edge buffer
  for each triangle currently in the triangle list
    Calculate the triangle circumcircle center and radius
    if the point lies in the triangle circumcircle then
      Add three triangle edges to the edge buffer
      Remove the triangle from the triangle list
    endif
  end for
  Delete all doubly specified edges from the edge buffer, this leaves the edges of the enclosing polygon only
  Add to the triangle list all triangles formed between the point and the edges of the enclosing polygon
end for
Remove any triangles from the triangle list that use the super triangle vertices
Remove the super triangle vertices from the POIList
Connect and get Dirichlet
```

Algorithm 1: Dirichlet construction based on POI.

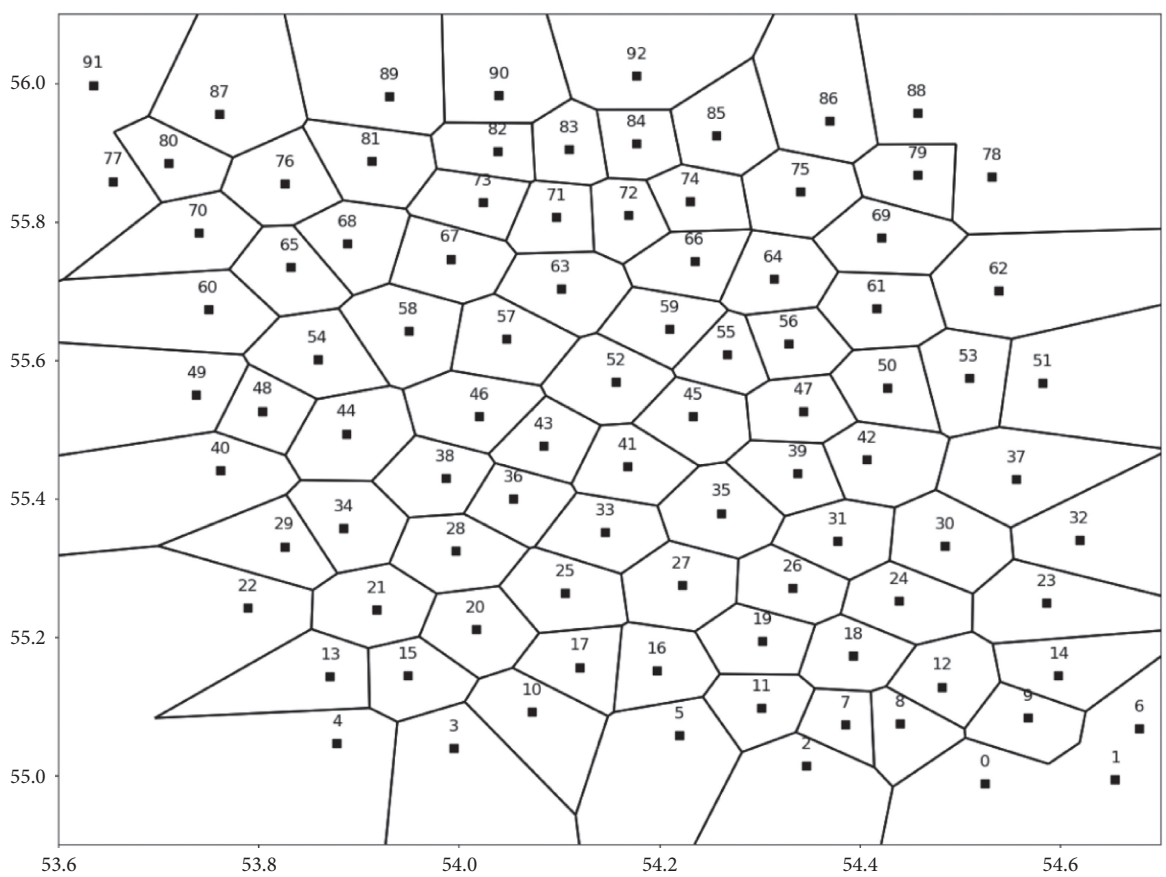


Figure 3: Dirichlet diagram.

blocks in this Dirichlet diagram in a fully random manner. A total of K false users belonging to different D blocks are generated to form a user anonymity set and sent to the LBS server, where the information point category C_{la} in the user request U_{client} is equal to the C_{la} in the user anonymity set Users.

The use of Dirichlet graph divides the continuous anonymous space into discrete D blocks; compared with the anonymous interval generated by the traditional K-anonymous method, it achieves the function of dividing the space. The advantage of this is that this method of randomly dividing the space will enhance anonymity and protect the security of ASR; the second is that it will not affect the quality of service while replacing the real location of the user with a fake location; the third is that it effectively avoids adding noise in the privacy protection method of the United States, the risk of privacy leakage caused by the impractical location of the noise.

3.2. Query Privacy Protection Algorithm Based on Private Information Retrieval. The function of the private information retrieval technology is to ensure that the private information of the retrieval initiator will not be leaked when the information retriever initiates a retrieval to the server. There are currently two mainstream private information retrieval methods: one is private information retrieval based on information theory, and the other is private information retrieval based on computational theory. The private information retrieval method based on information theory needs to send all service information from the LBS server to the mobile terminal. Although the user service quality and the absolute security of user privacy are guaranteed, the transmission cost is too large and still only stays at the theoretical level. Therefore, the current mainstream use of private information retrieval methods is based on computing power.

The problem model based on the intractable quadratic residue hypothesis is a common method of private information retrieval technology based on computational theory. In the private information retrieval protocol of quadratic residue model, the server generates a relational matrix from the data in the database, and the retrieval target of the trusted third-party server is one bit of data in the matrix. The mobile terminal initiates a query to the server according to its own private information. When the LBS server receives the query information, it performs modular multiplication on each row of elements in the matrix to obtain a query result and then returns the query result to the mobile terminal to complete a retrieval.

After the LBS server receives the user anonymity set Users, it will generate a relationship matrix according to the information point category C_{la}; this relationship matrix contains the proximity relationship between n information points and the K user positions.

Definition 5. Quadratic residue is as follows: set a large odd prime number $\lambda > 1, \mu \in Z$, and $1 \le \mu \le \lambda, (\mu, \lambda) = 1$; for the basic quadratic congruence $\Upsilon^2 \equiv \mu \,(\mathrm{mod}\lambda)$, if there is $\Upsilon \in Z$ that satisfies this congruence, then it is said that μ is a quadratic residue modulo λ; otherwise, it is called quadratic nonresidual.

Definition 6. $\lambda$is a large odd prime number, $(\mu, \lambda) = 1$, by the quadratic residue Euler discriminant conditions which are as follows:

(i) $\mu$is the necessary and sufficient condition of the quadratic residue modulo λas $\mu^{(\lambda-1/2)} \equiv 1\,(\mathrm{mod}\lambda)$.

(ii) μ is the necessary and sufficient condition of quadratic nonresidual modulo λas $\mu^{(\lambda-1/2)} \equiv -1\,(\mathrm{mod}\lambda)$.

Definition 7. The relation matrix generated in LBS server is $A_k^n = \begin{pmatrix} NR_{11} & \cdots & NR_{1k} \\ \cdots & O & \cdots \\ NR_{n1} & \cdots & NR_{nk} \end{pmatrix}$, in which $NR_{ij} \in \{0, 1\}$, the values of NR_{ij} represent the proximity relationship between the ith POI in the relation matrix and the jth user in the query set X_{inf}, 1 represents proximity, 0 represents alienation, n represents the number of information points, and k is the number of users in anonymous set.

Definition 8. $X_{inf} \otimes A_k^n = \Psi = \left\{ f(\varphi) | f(\varphi) = \prod_{m=1}^{k} x_m \cdot NR_{\varphi m}, x_m \in X_{inf}, NR_{\varphi m} \in A_k^n \right\}$, in which X_{inf} is the query set $X_{inf} = \{x_1, x_2, L, x_u, L, x_k\}$in the anonymous set Users and $\mathbf{A_k^n}$ is the relational matrix.

Definition 9. $h(m, \varphi) = x_m a_{\varphi m}$, where $x_m \in X_{inf}, NR_{\varphi m} \in A_k^n$. If and only if $h(m, \varphi)$result is 0, it is recorded as 1.

Definition 10. λ is a large odd prime number, $(\mu, \lambda) = 1$. Legendre symbol is defined as follows:

$$\left(\frac{\mu}{\lambda}\right) = \begin{cases} 1, & \text{If } \mu \text{ is a quadratic residue of modular } \lambda, \\ -1, & \text{If } \mu \text{ is a quadratic non residue of modular } \lambda. \end{cases} \tag{5}$$

In the quadratic residue theory, the attacker cannot figure out whether μ is a quadratic residue modulo λ without a given factorization of a large odd prime number λ.The trusted third-party server calculates the quadratic residue of K-1 module $\lambda$and the quadratic nonresidual x_u of one module in advance according to the large odd prime number in the user request R to form the query set X_{inf}, send it to LBS server, and x_u correspond to the real user to be queried. After the LBS server receives the user anonymity set sent by the trusted third-party server, it generates the relationship matrix $\mathbf{A_k^n}$ according to the type of information point C_{at} and performs the $X_{inf} \otimes A_k^n$ operation. Because the LBS server cannot identify the secondary nonresidual in X_{inf}, it returns the result set Ψ to the trusted third-party server.

According to Definition 7, it can be seen that the relationship matrix $\mathbf{A_k^n}$ records the neighbor relationship between K users and n information points, and the returned result set $\Psi$is a set composed of $f(\varphi)$. When μ and ν are quadratic residuals of modulo λ, it can be seen from

Definition 5 that $\Upsilon^2 \equiv \mu(\mathrm{mod}\lambda)$ is equivalent to $\Upsilon^2 \equiv \nu(\mathrm{mod}\lambda)$, and Definition 10 has $(\mu/\lambda) = (\nu/\lambda)$. Available from Definition 6, $(\mu/\lambda) \equiv \mu^{(\lambda-1/2)}(\mathrm{mod}\lambda)(\nu/\lambda) \equiv \nu^{(\lambda-1/2)}(\mathrm{mod}\lambda)$, $(\mu\nu/\lambda) \equiv (\mu\nu)^{(\lambda-1/2)}(\mathrm{mod}\lambda)$. So, there are $(\mu\nu/\lambda) \equiv (\mu\nu)^{(\lambda-1/2)} = \mu^{(\lambda-1/2)}\nu^{(\lambda-1/2)} \equiv (\mu/\lambda)(\nu/\lambda)(\mathrm{mod}\ \lambda)$, and because the Legendre symbol in Definition 10 has a value range of ±1 and λ is a large odd prime number, there is $(\mu\nu/\lambda) = (\mu/\lambda)(\nu/\lambda)$.

To sum up, there are inferences: when λ is a large odd prime number, μ and ν are relatively prime to λ; if both μ and ν are quadratic residues modulo λ, then $\mu\nu$ is also a quadratic residue modulo λ; if one of μ and ν is a quadratic residue of modulo λ, and the other is a quadratic nonresidual of modulo λ, then $\mu\nu$ is a quadratic nonresidual modulo λ. In the result set Ψ, we have the following.

When $f(i)$ is a quadratic nonresidual of module λ, it shows that $h(u, i) = x_u \cdot NR_{iu} = x_u$; that is, $NR_{iu} = 1$; that is, the user to be queried is adjacent to the ith POI.

When $f(i)$ is still the quadratic residue of module λ, it shows that $h(u, i) = x_u \cdot NR_{iu} = 1$; that is, $NR_{iu} = 0$; that is, the user to be queried is distant from the ith POI.

According to the result set Ψ returned by the LBS server, the trusted third-party server can obtain the proximity relationship between the real user and each information point. After determining the proximity relationship, the user can be guided to the next step.

The mainstream privacy protection strategy based on an independent architecture is to send the processed data information to the LBS server to ensure that the user's private information will not be leaked. However, when the user has high requirements for service quality, the LBS server can only send processed data providing service, and such service quality is at a loss. The application of private information retrieval technology solves the problems of information loss caused by factors such as the complexity of the network environment and the uncertainty of user behavior.

4. Discussion on K Values in KPDR

In the traditional K-anonymous privacy protection method, the user's privacy protection degree and service quality are affected by the K value. When the value of K is larger, the degree of privacy protection of the user is higher, and the quality of service is lower; when the value of K is smaller, the quality of service of the user is higher, but the user is susceptible to link attacks and privacy leakage. Therefore, choosing a K value that can balance the user's service quality, and the degree of privacy protection is the key to the traditional K-anonymous privacy protection method.

In the KPDR method, the selection of the K value is slightly different. The larger the value of K, the larger the user's anonymity set, and the higher the user's privacy protection. However, because of the application of private information retrieval technology to protect query privacy, the user needs to traverse the entire relationship matrix for each query, so that the user's request service efficiency will be affected; the smaller the value of K, the smaller the user anonymity set, the faster the traversal speed of the relationship matrix, and the higher the quality of service provided to users. Therefore, the degree of user privacy protection, service request efficiency, and server computing power are all related to the value of K. With the rapid development of the computer industry, the computing power of the computer has been significantly enhanced, which is enough to cope with the calculation amount of K taking a larger value. However, if K takes a very large value or the amount of concurrent user query requests is particularly high, the server still using this query will fail because of insufficient computing power and downtime or too long computing time.

It is assumed here that the computing power of the computer is unlimited. Given μ and ν in Definition 5, when μ is the quadratic residue of modulo λ, μ takes one of the series: $(-(\lambda-1/2))^2, (-(\lambda-1/\lambda-1) \quad +1)^2, \ldots, (-1)^2, (1)^2, \ldots, ((\lambda-1/2)-1)^2, (\lambda-1/2)^2$; the simplified residue system with the smallest absolute value of modulo λ is $-(\lambda-1/2), -(\lambda-1/2)+1, \ldots, -1, 1, \ldots, (\lambda-1/2)-1, (\lambda-1/2)$. Because $(-\alpha)^2 = \alpha^2$, μ is the quadratic residue of modulo λ if and only if the value is one of $(1)^2, \ldots, ((\lambda-1/2)-1)^2, (\lambda-1/2)^2$. And because when $1 \le \alpha \le \beta \le (\lambda-1/2)$, $\alpha^2 \equiv \beta^2(\mathrm{mod}\lambda)$, so all quadratic residuals of modulo λ are $(1)^2, \ldots ((\lambda-1/2)-1)^2, (\lambda-1/2)^2$, a total of $(\lambda-1/2)$. Thus, there are $(\lambda-1)-(\lambda-1/2) = (\lambda-1/2)$ quadratic nonresidues of module λ.

Because the trusted third-party needs to choose K-1 quadratic residue of modulo λ and a quadratic nonresidual of modulo λ, the value of K needs to satisfy $K \le (\lambda + 1/2)$; because each query needs a quadratic nonresidual of modulo λ, $2 \le K$ needs to be satisfied.

To sum up, the value of K is related to the computing power of the computer and positively correlated with the degree of privacy protection, and the theoretical value of K is $2 \le K \le (\lambda + 1/2)$.

5. Security Analysis

With the increasing number of users using LBS service, criminals have increased attacks on users' privacy. This section will analyze the security of KPDR method in the face of various attacks.

5.1. Resist Attacks Based on Geographic Location Information. The attack based on geographic location information is mainly due to the incompleteness of privacy protection technology, which leads to many unrealistic false positions in the generated ASR. When criminals find that a large number of false locations are distributed among lakes and cliffs, these locations can be easily excluded, which increases the probability of the user's true location leaking. The KPDR method based on the ASR generated by the actual POI can resist this attack method, because the actual POI position will not be in the lake or cliff, and if the V block generated based on the POI contains similar lakes, the KPDR method only one false location will be distributed in the area, avoiding the generation of a large number of invalid false locations, and the impact on the leakage probability of the user's true location is almost zero.

5.2. Resist Inference Attacks Based on User Background Knowledge. Attacks based on users' information background knowledge refer to privacy attacks launched by attackers on the basis of mastering users' basic information, such as interests and habits. When the KPDR method responds to the request service initiated by the user, the user request is divided into a Dirichlet graph each time, and the type of user request is different, and the generated Dirichlet graph will be different. In the entire privacy protection process, the user's basic information is never exposed, the attacker cannot infer the user's requested service information, and the KPDR method can resist such attacks very well.

5.3. Resist Continuous Multiquery Attack. Continuous multiquery attack means that when a user continuously requests a service for a period of time, the attacker infers the next position of the user according to the current moving speed of the user and the generated ASR results. In KPDR, when the user makes a continuous query, TTPs will judge the user's position every time. Every time the user initiates a query, new false information will be regenerated according to the new V block. Every false information and ASR update make it impossible for the attacker to analyze any information of the user in time. Therefore, KPDR method has a good effect on the attack of continuous multiquery and effectively protects the privacy of the user.

5.4. Resist Monitor Attacks by Attackers. Monitoring attacks are mainly aimed at privacy protection methods using distributed point-to-point architecture. In this architecture, users spontaneously form anonymous groups through P2P protocol, and attackers can impersonate ordinary users to participate in anonymous group construction. If attackers monitor users' requests in anonymous groups, they can monitor users' private information by analyzing the returned results. The difference is that KPDR adopts the trusted third-party center architecture and TTPs as the overall carrier to complete centralized data processing. Users do not communicate or interfere with each other when requesting services, and attackers cannot listen to any request information from other users.

6. Experimental Results Analysis

The experiment makes a detailed comparison between the KPDR method proposed in this paper and the privacy protection method (GRAM), which is also based on the principle of K- anonymity. The GRAM [15] method constructs a protection graph that satisfies the anonymity requirements of (k, l) identifies vertices in the protection graph, satisfies users' privacy requirements by constantly adding vertices and edges, alleviates the contradiction between privacy protection and quality of service, and has some advantages over traditional k-anonymity methods, but because the GRAM method cannot rule out all redundant edges in the process of adding vertices. It not only reduces the efficiency of anonymity, but also has some shortcomings. This paper will analyze the difference, advantages, and disadvantages between KPDR and GRAM through experiments. Because GRAM has carried out data experiments with the traditional k-anonymity method in terms of security and efficiency, this paper will not repeat it in the data comparison but will explain it in the theoretical analysis.

6.1. Analysis of Computing and Communication Overhead

6.1.1. Computational Overhead Analysis of KPDR Method. In the KPDR method, based on the POI data in the geographic information system, different kinds of POIs are generated into Dirichlet diagrams by using Delaunay Triangulation Algorithm. Considering that the update of POI data in real life is not frequent, the strategy of sacrificing storage space is adopted to reduce the computing overhead of the server. The Dirichlet diagrams divided by different kinds of POI are stored in TTPs in advance, and when updating the POI data, only the Dirichlet diagrams generated by the corresponding POI categories need to be recalculated, which greatly reduces the computational overhead of TTPs. In the process of privacy protection, using Dirichlet graph to segment the interval, TTPs need to traverse the proximity relationship between POI and users, and the complexity is $O(n)$. Although the computational overhead increases linearly, combined with the classification of POI before, the search cardinality has been greatly reduced, which improves the computational efficiency of TTPs and reduces the computational overhead on the premise of ensuring security. When selecting the false position of the user, the calculation cost is related to the value of K; because of the characteristics of the Dirichlet graph, the nearest neighbor calculation is not required. Although the calculation cost will increase with the increase of K, the overall cost will not be generated with excessive changes.

6.1.2. Analysis of Communication Cost of KPDR Method. In the KPDR method, Dirichlet graphs divided according to different types of POI are jointly maintained by LBS and TTPs. LBS accepts user anonymity data packets and responds to user requests. Therefore, the size of communication overhead is related to the speed of LBS processing user requests, especially in the face in the case of multiple users and high concurrency; the throughput of LBS directly affects the quality of service for users. In the process of forming an anonymous set, the communication overhead increases with the increase of the size of the anonymous set. Due to the use of the quadratic residue hypothesis model in this paper, LBS accepts the generation of the relation matrix of the user anonymous set, although the proximity relationship between two POI and K users is recorded in the relation matrix; the TTPs does not need to index all proximity relationships; it only needs to retrieve a neighbor relationship between the user and the POI. This not only reduces the communication overhead to a certain extent, but also ensures that the overall communication overhead will

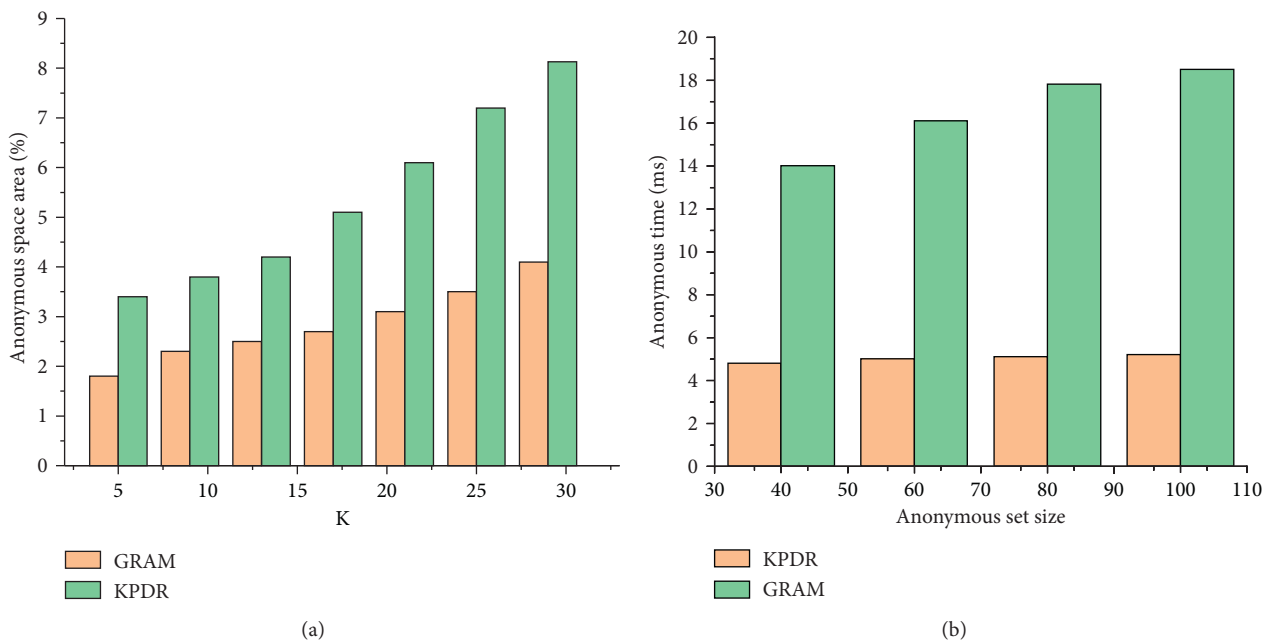


Figure 4: Comparison of anonymous space and anonymous time. (a) Anonymous space area comparison. (b) Anonymous time comparison.

not increase significantly with the increase of the size of user anonymous sets.

6.2. Experimental Comparative Analysis. The experiment uses a dataset of Beijing's POI category for catering services to verify the performance of the KPDR algorithm. The data comes from the POI set of AutoNavi Map, which contains about 10,821 POIs. The algorithm is implemented using Python 3.8.3 programming. The environment is configured as processor Intel (*R*) Core (TM) i7-4710HQ CPU @ 2.50 GHz (8 CPUs), ~2.5 GHz, memory 4 GB, graphics card NVIDIA GeForce GTX 850M, operating system Windows 10 Professional Edition. The Forbidden City Museum is the center of the circle and the distribution of POIs within a 5,000-meter radius after being scaled down.

6.2.1. Comparison between Anonymous Space and Anonymous Time. As shown in Figure 4(a), the number of POIs in the KPDR method is fixed, and the value of K is continuously increased. As the area of the anonymous space becomes larger, the degree of privacy protection of users will be higher, but no matter what value K takes, the area of anonymous space of KPDR is always larger than that of GRAM, and as the value of K becomes larger, the area of anonymous space that differs between the two methods increases. As shown in Figure 4(b), the number of POIs is fixed to ASR; the KPDR method uses the characteristics of Dirichlet graph model and does not need to judge by the algorithm of the nearest distance. The LBS server stores the Dirichlet graph under the current POI division, which is updated only when the POI is changed, while the anonymous time of GRAM method increases significantly because it needs to meet the (k, l) mechanism. Therefore, when the scale of anonymous set is increased, the difference of anonymous time between the two methods will be greater.

6.2.2. Comparison of Anonymous Success Rate and Communication Overhead. As shown in Figure 5(a), the ASR is fixed. With the continuous increase of K value, the anonymous success rate of the two methods remains at a relatively high level, but the anonymous success rate of the KPDR method is still higher than that of GRAM method. The GRAM method needs to continuously add edges to the base map to protect user privacy. Each edge addition must be recalculated and K integrations are required, so the anonymous success rate will be lower. As shown in Figure 5(b), the average communication cost of both methods increases with the increase of K value, and the increase of KPDR method is relatively slow, because the increase of K value indicates that users need more location information to construct anonymous areas when requesting services; when the GRAM method increases the value of K and when K reaches a certain node value, it will add a vertex corresponding to the edge on the protection graph, so the GRAM algorithm increases gently and jumps. From the results, the average communication cost of this method is lower than that of GRAM method.

6.2.3. Comparison of Influence of Different Values of POI and K on KPDR Method. As shown in Figure 6(a), taking the number of POIs as 1500, 3000, 4500, 6000, 7500, and 9000, you can see that the anonymous time increases with the increase in the number of POIs; at the same time, keeping the POI value unchanged and increasing K value, the anonymous time will also increase slightly. Although the anonymity time of this scheme will increase with the

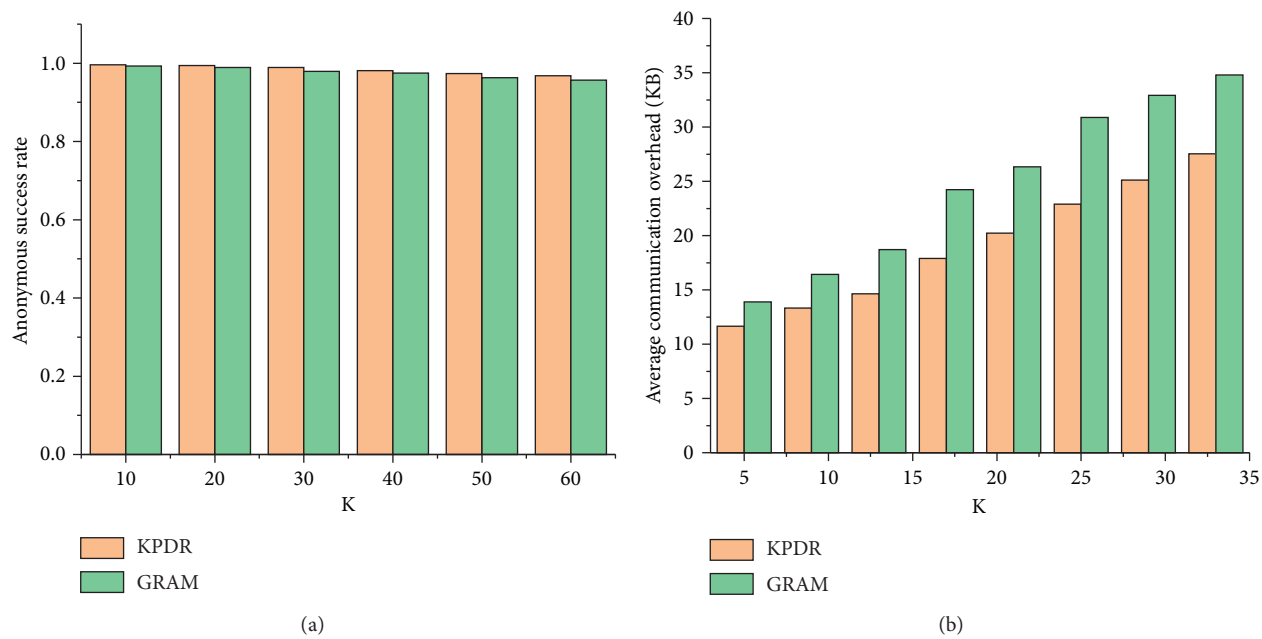


Figure 5: Average communication overhead. (a) Anonymous success ratio comparison. (b) Average communication overhead comparison.

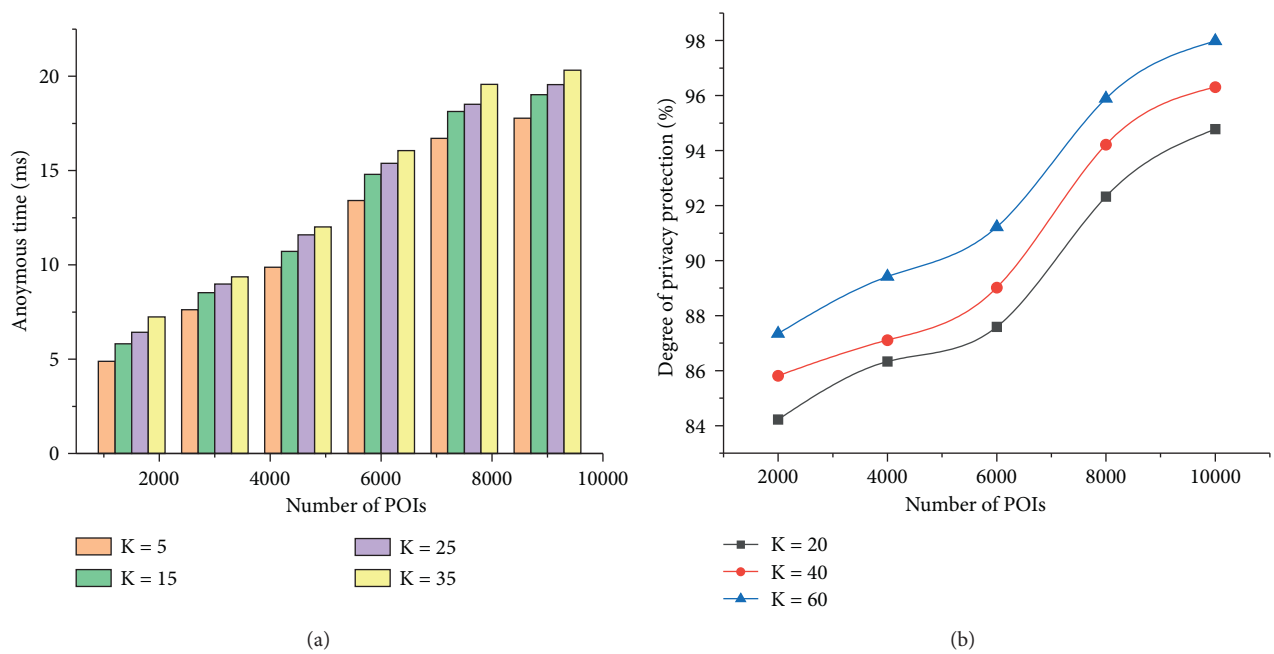


Figure 6: Comparison of the influence of POI and K on this scheme. (a) Influence of POI and K on anonymous time. (b) Influence of POI and K on privacy protection degree.

increase of the number of POI and K values, the overall anonymity efficiency is still controlled at a good level. In real life, when there are so many similar POIs, the coverage area is large enough, and such anonymity efficiency is enough to ensure the quality of service for users, which also proves the superiority of this method. As shown in Figure 6(b), take the number of POIs as 2000, 4000, 6000, 8000, and 10000 to test the degree of privacy protection. It can be seen from the experiment that when the K value is equal; the more POIs are generated, the larger the coverage area of the Dirichlet graph is, the higher the dispersion when constructing user anonymity sets, and the higher the degree of privacy protection of users; when the number of POIs is equal, the K value becomes greater, the degree of privacy protection of users is higher, and the overall degree of privacy protection is maintained at a relatively high level.

The GRAM method has proved its superiority compared to the traditional privacy protection method. The experimental results show that the KPDR method proposed in this paper has better security performance and anonymity efficiency than the GRAM method. By storing the Dirichlet graph on the LBS server, the space is exchanged in time, which avoids service congestion due to a large number of user requests, further improves the user's privacy security, and increases the practicability of the method.

7. Conclusion

In this paper, a KPDR method based on K-anonymity mechanism is proposed. By using Dirichlet graph model and quadratic residue theory model, it can effectively resist link attacks and continuous query attacks and solve the problem that anonymous areas are vulnerable to attacks. With the advent of the era of big data as a service provider, we must fully consider the possibility of a large number of requests from users, so the next step will be to improve the query efficiency of users when the concurrent amount of service requests is high.

Data Availability

The data come from the POI set of AutoNavi Map, which contains about 10,821 POIs.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant no. 61300216.

References

[1] L. Zhang, J. Li, S. Yang, B. Wang, and X. Bian, "A novel attributes anonymity scheme in continuous query," *Wireless Personal Communications*, vol. 101, pp. 943–961, 2018.

[2] Location Based Services (LBS) and Real-Time Location Systems (RTLS) Market-Global Forecast to 2020, https://www.digitaljournal.com/pr/2758079.

[3] Y. Sun, M. Chen, L. Hu, Y. Qian, and M. M. Hassan, "ASA: against statistical attacks for privacy-aware users in location based service," *Future Generation Computer Systems*, vol. 70, pp. 48–58, 2016.

[4] W. He, "Research on LBS privacy protection technology in mobile social networks," in *Proceedings of the 2017 IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*, Chongqing, China, 2017.

[5] Z. Lei, H. Lili, L. Desheng, L. Jing, J. Qingfeng, and Y. Qi, "An attribute generalization mix-zone without privacy leakage," *IEEE Access*, vol. 7, pp. 57088–57099, 2019.

[6] Y. Zhang, Q. Chen, and S. Zhong, "Privacy-preserving data aggregation in mobile phone sensing," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 5, pp. 980–992, 2016.

[7] R. Wang, X. Jie, Z. Lin, and R. Si, "An improved algorithm of individuation k-anonymity for multiple sensitive attributes," *Wireless Personal Communications an Internaional Journal*, vol. 95, pp. 2003–2020, 2017.

[8] Y. Yuji and I. Kouichi, "K-presence-secrecy: practical privacy model as extension of k-anonymity," *Ice Transactions on Information & Systems*, vol. 100, pp. 730–740, 2017.

[9] X. Li, M. Miao, H. Liu, J. Ma, and K.-C. Li, "An incentive mechanism for k-anonymity in LBS privacy protection based on credit mechanism," *Soft Computing*, vol. 21, no. 14, pp. 3907–3917, 2017.

[10] L. Zheng, H. Yue, Z. Li, X. Pan, M. Wu, and F. Yang, "K-Anonymity location privacy algorithm based on clustering," *IEEE Access*, vol. 6, pp. 28328–28338, 2018.

[11] J. Wang, Z. Cai, Y. Li, D. Yang, J. Li, and H. Gao, "Protecting query privacy with differentially private k-anonymity in location-based services," *Personal & Ubiquitous Computing*, vol. 22, pp. 453–469, 2018.

[12] K. Wang, W. Zhao, J. Cui, Y. Cui, and J. Hu, "A K-anonymous clustering algorithm based on the analytic hierarchy process," *Journal of Visual Communication and Image Representation*, vol. 59, pp. 76–83, 2019.

[13] H. Sun and S. A. Jafar, "The capacity of private information retrieval with colluding databases," in *Proceedings of the IEEE Global Conference on Signal & Information Processing (GlobalSIP)*, pp. 941–946, Washington, DC, USA, November 2017.

[14] H. Sun and S. A. Jafar, "The capacity of private information retrieval," *IEEE Transactions on Information Theory*, vol. 63, no. 7, pp. 4075–4088, 2017.

[15] R. Mortazavi and S. H. Erfani, "GRAM: An Efficient (K, l) Graph anonymization method," *Expert Systems with Applications*, vol. 153, pp. 113454–113463, 2020.

Hindawi
Complexity
Volume 2021, Article ID 6687463, 8 pages
https://doi.org/10.1155/2021/6687463

Research Article

Enhance the Transfer Capacity of Multiplex Networks

Fei Shao,[1,2] **Wei Zhao**,[1,2] **and Binghua Cheng**[1]

[1]*School of Computer Engineering, Jinling Institute of Technology, Nanjing 211169, China*
[2]*Jiangsu Key Laboratory of Data Science & Smart Software, Jinling Institute of Technology, Nanjing 211169, China*

Correspondence should be addressed to Fei Shao; shaofei@jit.edu.cn

Received 20 November 2020; Revised 22 December 2020; Accepted 28 December 2020; Published 13 January 2021

Academic Editor: Yongsheng Hao

Most complex real systems are found to have multiple layers of connectivity and required to be modelled as multiplex networks. One of the extremely critical problems is to reduce the congestion and enhance the transfer capacity, especially in real communication networks with a big data environment. A novel and effective strategy to improve traffic and control congestion is proposed by adding edges according to their weights which are defined by the topology structural properties. Furthermore, which layer is more effective when our strategy is applied is discussed based on its topology structure. Adding edges between nodes whose product of multiplex network betweenness is the highest is confirmed to be more effective, particularly in the layer with stronger community structure. Simulation experiments on both computer-generated and real-world networks demonstrate that our strategy can enhance the transfer capacity of multiplex networks significantly, which is in good agreement with our analysis.

1. Introduction

The complex real systems can be depicted as complex networks [1, 2] with small-world effects [1] or scale-free properties [2]. These complex systems, such as online social networks, biological systems, and the Internet, are abstracted as a graph, while nodes represent individuals and edges represent the interactions among them. In recent decades, due to the ever-increasing amount of information transfer on the Internet and online social networks, it is of paramount interest to explore the transfer capacity of complex networks [3–7]. Models of packet transfer process in computer networks are presented in many previous studies. In most of the models presented, all nodes in the network are both hosts and routers. Hosts generate packets to be sent to destination addresses and receive packets from other hosts or routers, while routers forward the packets to their destinations according to the specified routing strategy. At each time step, every node in the network generates a packet with a constant packet generated rate λ, with randomly chosen source and destination addresses. Once a packet is generated, it is placed at the tail of the queue with packets generated by itself or transferred from neighbours in the previous time steps. Meanwhile, the first C_i packets at the head of the queue of each node i are forwarded one step to their destinations and placed at the tail of the queues of the selected nodes. When a packet arrives at its final destination, it will be expelled from the network. The transfer capacity of each node is defined in different ways. In some models, it is considered to be the same in all nodes while to be proportional to its topology metrics such as degree or betweenness in other models. These former studies are focused on the critical value λ_c of the packet-generated rate [3, 5, 8]. We can obtain a phase transition from free state to congestion at the critical point λ_c along with the increase of the packet-generated rate. For $\lambda < \lambda_c$, the amount of generated and transferred packets is balanced and the whole network is in a steady free state. For $\lambda > \lambda_c$, it turns into a jammed state because the node cannot transfer the packet beyond its limited transfer capacity. This critical packet-generated rate λ_c can best reflect the maximum packet transfer capacity of a network. Therefore, it is used to estimate which routing strategy is of more validity. It may have a certain reference value for the real-world networks.

Along with the development of studies on the topology of complex networks, a mesoscopic description, community structure, is found in many networks [9–14]. It is presented that there is a tendency for nodes to divide into communities

within which node-node connections are dense but between which connections are sparse. The modularity Q [12] is defined as a measure of the community structure which can specify a certain mesoscopic description of the network in terms of communities being more or less accurate. The larger value of Q means a more accurate division of community. The influence of community structure on packet transport and how to enhance the transfer capacity based on community structure are also investigated [4, 15, 16].

Traditional studies of complex networks usually assume that all nodes are linked to each other by a certain type of edge to produce a single-layer network. Recently, it has been recognized that lots of complex real systems are not composed by single network primitively, but by multiplex network [17–24]. They consist of a series of N nodes linked by L different kinds of interactions. All interactions of the same kind determine a unique layer/network of the multiplex network. Nodes have diverse neighbours in each layer. For example, the multiplex network of relationships existing among employees of the Computer Science Department of Aarhus University [23] contains 61 nodes (employees) and five different layers (five kinds of online or offline relationships): Facebook, Leisure, Work, Co-authorship, and Lunch. Generally, the routing strategy is based on the shortest-path algorithm in the single-layer network. However, in a multiplex network, there are two distinct kinds of shortest paths: paths that only use a single layer and paths that use more than one layer. The dynamic process in multiplex networks is becoming a hot spot of current research [17–20, 25–27].

Adjusting the network topology structure, such as adding or deleting some edges, is effective to improve the network transfer capability of the single-layer network [4, 28, 29]. The multiplex network consists of two or more single-layer networks with different community structures. How to enhance the transfer capacity based on the interior topology characteristics attracts our attention. By adding edges in the light of the different definitions of edge weight, we present strategies to enhance the transfer capacity of the multiplex network. And the impacts of the nodes number N and the modularity Q of different layers on our strategies will be discussed.

The rest of this paper is organized as follows. Section 2 presents the proposed routing strategies method. Extensive simulation experiments are conducted to validate our routing strategies and make comparisons, and the results are reported in Section 3. Eventually, conclusions are made in Section 4.

2. Materials and Methods

The node betweenness b_i is widely used to access the possible traffic through node i under a specified routing strategy, in general, the shortest-path routing algorithm [30]. It is usually defined as follows:

$$b_i = \sum_{s,t} \frac{\sigma(s,i,t)}{\sigma(s,t)}, \tag{1}$$

where $\sigma(s,i,t)$ is the amount of shortest paths connected nodes s and t while passing through node i and $\sigma(s,t)$ is the total amount of shortest paths connected nodes s and t. The likelihood that a generated packet will travel through node i is $b_i/\sum_{j=1}^{N} b_j$. Accordingly, the mean number of packets that node i receives at a certain time step is $N * \lambda * b_i/(N * (N-1)) = \lambda * b_i/(N-1)$. When the number of incoming packets is no less than the packet transfer capacity of node i, that is, $\lambda * b_i/(N-1) \geq C_i$, the node is unable to forward all packets to their destination and it will produce a congestion in the network gradually. The critical packet generated rate λ_c is [3, 4]

$$\lambda_c = \min \frac{C_i * (N-1)}{b_i}. \tag{2}$$

When it turns to a multiplex network, there are two different types of shortest paths: intralayer paths and interlayer paths. When we investigate the dynamics of multiplex network, the number of shortest paths must be considered together with the intralayer paths and interlayer paths through a certain node. The critical packet injection rate of the multiplex network is as follows [19, 25, 27]:

$$\rho_c = \min \frac{\tau_i * (N-1)}{L * B_i}, \tag{3}$$

where τ_i is the max node processing rate of node i, which is similar to the packet delivery capability C_i in the previous studies. In general, we suppose that all nodes have the same maximum processing rate τ_i and we set $\tau_i = 1$ for simplicity. L is the layer number and B_i is the multiplex betweenness of node i. The critical injection rate ρ_c is used to estimate the maximum transfer capacity of a multiplex network.

In a system divided into m communities, a $m \times m$ symmetric matrix $\mathbf{E}$ is used to calculate the modularity whose element e_{ij} is the portion of all edges in the system that connect nodes in two different communities, community i and community j. The modularity measure, Q, is as follows [12, 13]:

$$Q = \sum_i \left(e_{ii} - a_i^2\right), \tag{4}$$

where $a_i = \sum_j e_{ij}$.

Different divisions of the network result in different Q, and the maximum is symbolized by $Q_{\max}$. The greater modularity $Q_{\max}$ implies the stronger community structure inside the system. Multiplex networks always contain two or more single-layer networks with different community structures. Our strategies are as follows:

(1) In a certain single-layer network of multiplex network, we calculate the weight of all node pairs, W_{ij}, between nodes i and j, in different ways according to different definitions. In the MBP strategy, W_{ij} is equal to the product of their multiplex betweenness $B_i * B_j$. In the SBP strategy, W_{ij} is equal to the product of their node betweenness $b_i * b_j$ using the shortest-path routing algorithm. In the DDP strategy, W_{ij} is equal to the product of their degree $k_i * k_j$.

(2) We sort the weights in decreasing order and add the edge between the node pair whose weight is at the top. If adding an edge will cause multiple edges between the node pair, we will not add the edge, but cope with the node pair rank next.

(3) We renew the weight W_{ij} and duplicate step 2 until a certain f_e of edges are added.

In a simple network with N nodes, $N*(N-1)/2$ edges are the maximum. If the mean degree is $<k>$, there exist $N * <k>/2$ edges approximately. The number of edges we can add is $N * (N-1)/2 - N * <k>/2$. We only add 10 percent of the edges we can add at most. A fraction $f_e = 1$ means $0.1 * (N * (N-1)/2 - N * <k>/2)$ edges are added. When each edge is added, we calculate the critical packet injection rate ρ_c of the new multiplex system and record the enhancement as ρ_c/ρ_{c0} where ρ_{c0} is the critical packet injection rate of the original multiplex system.

For comparison, we define the RAN strategy as adding edges randomly (also without multiple edges). And, the validity of our strategies on different layers will be discussed by adapting our strategies in different layers.

3. Results and Discussion

We generate a multiplex network that only consists of two Erdős–Rényi single-layer networks. In each single-layer network, we utilize a series of pseudorandom networks with N nodes and separate these nodes into m communities. Each node has the average number of edges, Z_{in}, connected to the nodes in the same community and Z_{out} connected to the nodes of any other communities, while the average degree $<k> = Z_{\text{in}} + Z_{\text{out}}$ is constant. We can adjust Z_{in} for different community structures. In all simulations, we generate 100 multiplex networks randomly to calculate the average.

Firstly, we generate a multiplex network with 128 nodes which are separated into 4 equal communities while the average degree $<k>$ is fixed to be 16. In layer 1, Z_{in} is fixed to be 8, which means the single-layer network in layer 1 is a totally random network. In layer 2, we change Z_{in} from 8 to 11 and 14 to get increasingly pronounced community structure. The results are exhibited in Figure 1.

As shown in Figure 1, when f_e is larger than zero, ρ_c/ρ_{c0} is greater than 1. It means that when some edges are added, the critical packet injection rate of the new multiplex network is greater than that of the original multiplex network. Therefore, our strategies can enhance the transfer capacity of the multiplex network. In Figures 1(a)–1(c), the MBP strategy, adding edges with the highest product of multiplex betweenness, is the most effective one, while the enhancement as ρ_c/ρ_{c0} of the MBP strategy is the highest. The modularity $Q_{\max}$ of the single network in layer 1 is about 0.2245. In layer 2, when Z_{in} is 8, the modularity $Q_{\max}$ is close to 0.2245. When Z_{in} increases to 11, the modularity $Q_{\max}$ is about 0.4404 and 0.6288 for $Z_{\text{in}} = 14$. By comparing Figures 1(a) with 1(b) and 1(c), we can discover that they are more effective in the network with stronger community structure, especially the MBP strategy.

Then, we apply our strategies in different layers to check the impact of community structure on our strategies. In layer 1, we set Z_{in} as 14, which means the single-layer network in layer 1 has a strong community structure. In layer 2, we change Z_{in} from 8 to 14 to obtain the results presented in Figure 2.

From Figure 2, we can observe that the MBP strategy is also the most effective among these strategies. And, when the layer where our strategies are applied has stronger community structure, our strategies are more effective. Comparing Figure 2(a) with Figure 1(c), we can discover that in a multiplex network consisting of two layers with different community structures, applying our strategies in the one with pronounced community structure yields better results.

Afterwards, we double the average degree $<k>$ to 32 to check the influence of average degree $<k>$. Results are illustrated in Figure 3.

Due to the increase of the average degree $<k>$, the absolute number of edges we can add is rising. However, we keep the relative proportion of added edges fixed to 10 percent. The change of average degree almost has no influence on our strategies. Figure 3 also proves that the MBP strategy works better than the other strategies, particularly in the layer with strong community structure.

The impact of communities number m is presented in Figure 4.

From Figures 4(a) and 4(b), we can detect the enhancement of the MBP strategy is still the highest. The increase of communities number results in more accurate division of the network and the stronger community structure. In Figure 4(a), the modularity of layer 2 is about 0.3234, while in Figure 4(b), it is 0.6982. Hence, the enhancement in Figure 4(b) is nearly 268.48%, which is greater than 198.37% in Figure 4(a).

Finally, we check the influence of network size. We generate two-layer multiplex network with $N = 256$ nodes in $m = 8$ communities. The average degree $<k>$ and the adjusting parameter of community structure Z_{in} are also increased accordingly. Results are shown in Figure 5.

The enhancement ρ_c/ρ_{c0} of the MBP strategy is the highest as usual regardless of the expansion of network scale. The increase of adjusting parameter of community structure Z_{in} leads to stronger community structure. Figure 5 also validates that our strategies work more efficiently in the network with distinct community structures.

The incessant expansion of the network poses a new challenge to our strategies. Since most of the real networks are rather large with massive number of nodes and edges, it is of vital importance to consider the computational cost of the algorithm required to compute the multiplex network betweenness. The computational complexity of our strategies is mainly dominated by the calculation of the multiplex betweenness. The time complexity of

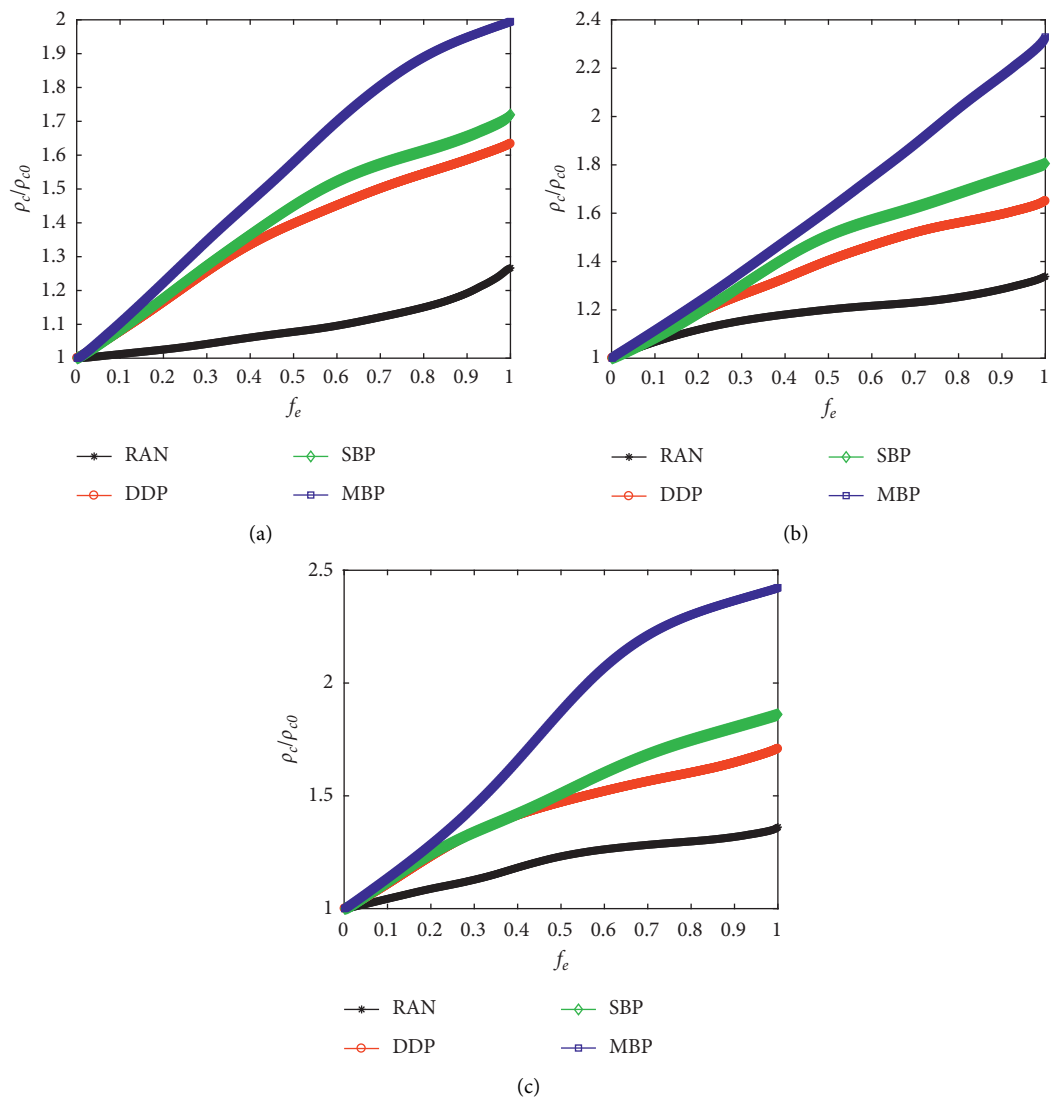


Figure 1: The enhancement of our strategies in multiplex network: (N) = 128, (m) = 4, <(k)> = 16, and $(Z)_{in} = 8$ in layer 1; in layer 2, (a) $(Z)_{in} = 8$, (b) $(Z)_{in} = 11$, and (c) $(Z)_{in} = 14$.

computing betweenness centrality of unweighted multiplex networks is $O(L*N*M)$ by using Breadth First Search [17] (M is the number of edges) which is acceptable under current circumstances.

The above simulations are run in computer-generated multiplex networks. Then, we verify our three strategies on real-world systems. We apply our strategies to the multiplex network of relationships among employees of the Computer Science Department of Aarhus University [23]. We choose two layers with Giant Components: the "Lunch" network and the "Work" network. The "Lunch" network has 193 edges and the modularity is 0.6548, while the "Work" network has 194 edges and the modularity is 0.4587. We apply our strategies in each layer to test them and get the simulations shown in Figure 6.

As shown in Figure 6(a), the MBP strategy can almost triple the transfer capacity when it is applied in the "Work" network whose modularity is 0.4587. In Figure 6(b), the enhancement is up to 333.4% when the MBP strategy is used in the "Lunch" network with stronger modularity of 0.6548. It indicates that our strategies are of practical value in enhancing transfer capacity in real multiplex networks.

Adding edges between nodes with the highest product of multiplex betweenness will make the two nodes to connect directly to each other. In general, it will reduce the load of those nodes with high multiplex betweenness. Stronger community structure will result in more nodes with high multiplex betweenness. That is why our strategies are more effective in the network with pronounced community structure. To uncover how our strategies work, we conduct further research on the initial and final multiplex betweenness. We employ the MBP strategy on the computer-generated multiplex network with $N = 128$, $m = 4$, $<k> = 16$, and $Z_{in} = 8$ in layer 1 and $Z_{in} = 14$ in layer 2 (the same simulation environment as Figure 1(c)) and the real CS-Aarhus multiplex network (the same simulation

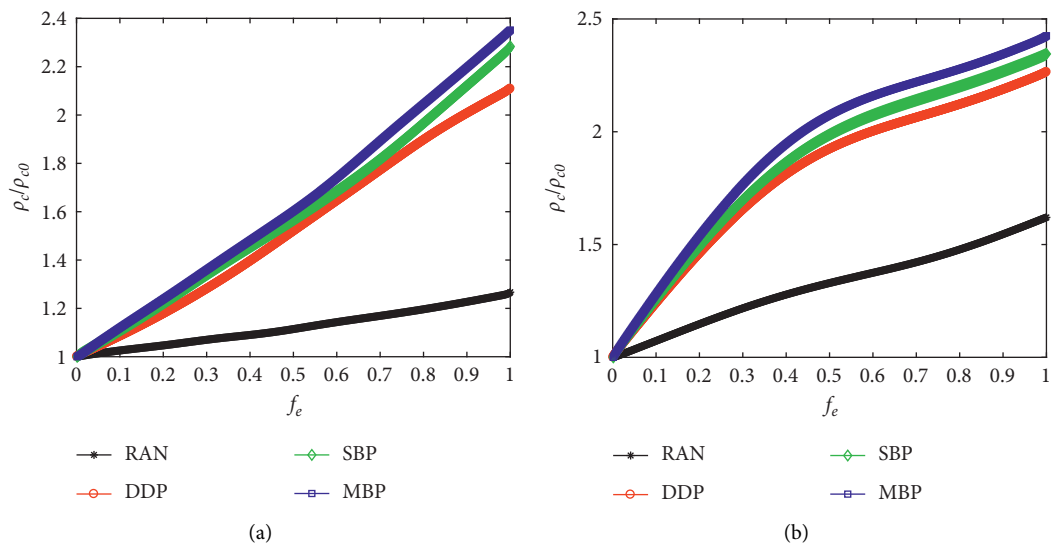


FIGURE 2: The enhancement of our strategies in multiplex network: (N) = 128, (m) = 4, <(k)> = 16, and $(Z)_{in}$ = 14 in layer 1; in layer 2, (a) $(Z)_{in}$ = 8 and (b) $(Z)_{in}$ = 14.

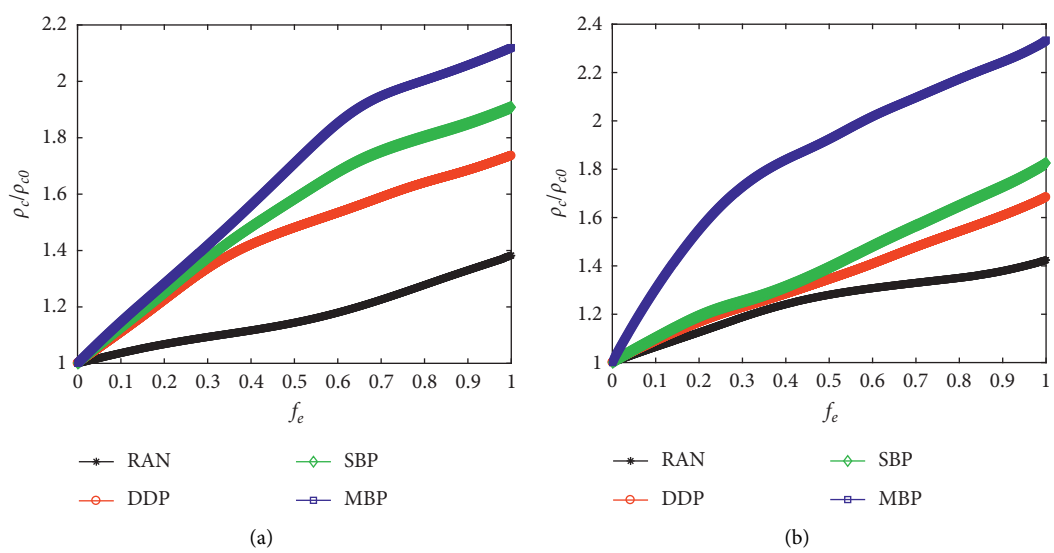


FIGURE 3: The enhancement of our strategies in multiplex network: (N) = 128, (m) = 4, <(k)> = 32, and $(Z)_{in}$ = 16 in layer 1; in layer 2, (a) $(Z)_{in}$ = 16 and (b) $(Z)_{in}$ = 28.

environment as Figure 6(b)). The multiplex betweennesses plot against the node index is presented in Figure 7.

Initially, in the original multiplex network, the multiplex betweennesses are distributed over a pretty wide range (see the square in Figure 7). After applying the MBP strategy, in the ultimate multiplex network, the multiplex betweennesses are restricted to an extremely narrow region (see the plus sign in Figure 7). There is a significant decrease in the highest multiplex betweenness of all nodes. The multiplex betweennesses are more uniformly distributed after our

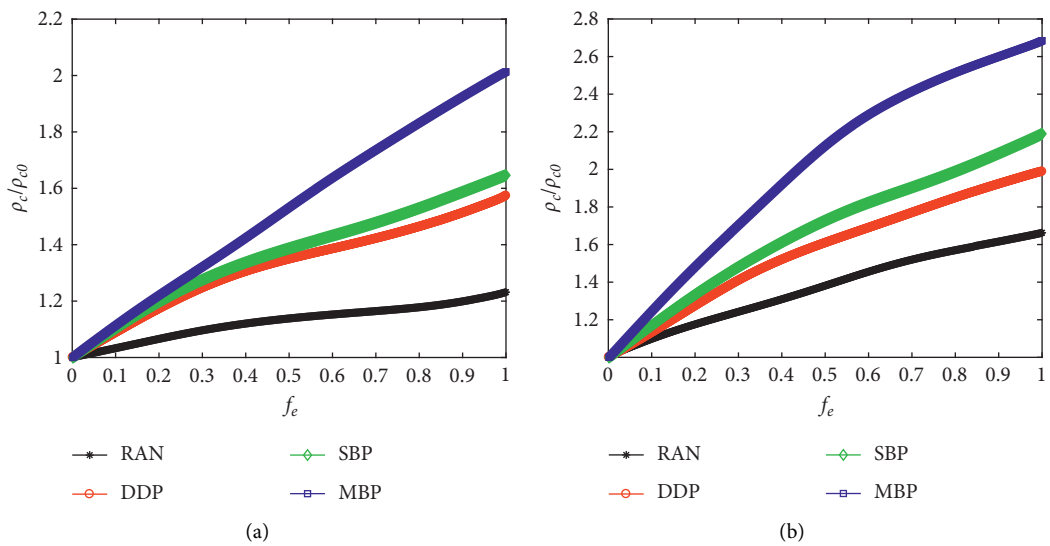


FIGURE 4: The enhancement of our strategies in multiplex network: (N) = 128, (m) = 8, <(k)> = 16, and $(Z)_{in} = 8$ in layer 1; in layer 2, (a) $(Z)_{in} = 8$ and (b) $(Z)_{in} = 14$.

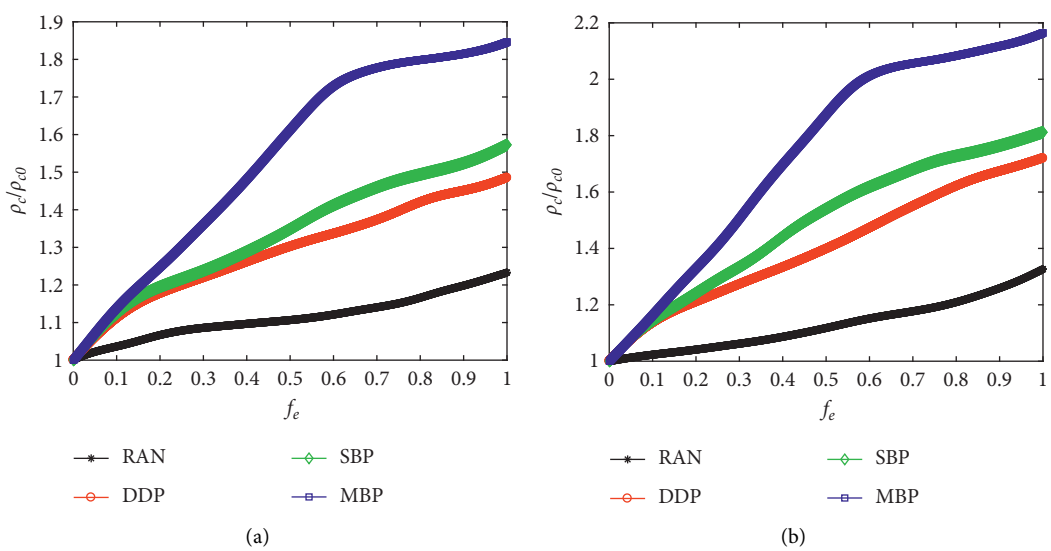


FIGURE 5: The enhancement of our strategies in multiplex network: (N) = 256, (m) = 8, <(k)> = 32, and $(Z)_{in} = 16$ in layer 1; in layer 2, (a) $(Z)_{in} = 16$ and (b) $(Z)_{in} = 28$.

strategies are applied. In Figure 7(b), there are a few nodes with comparatively high multiplex betweenness. The sharp declines of these nodes lead to a higher increase of network transfer capacity.

In our strategies, we add edges between nodes which will result in shortcuts between nodes. Therefore, the average path length will be reduced and the small-world phenomenon is still maintained.

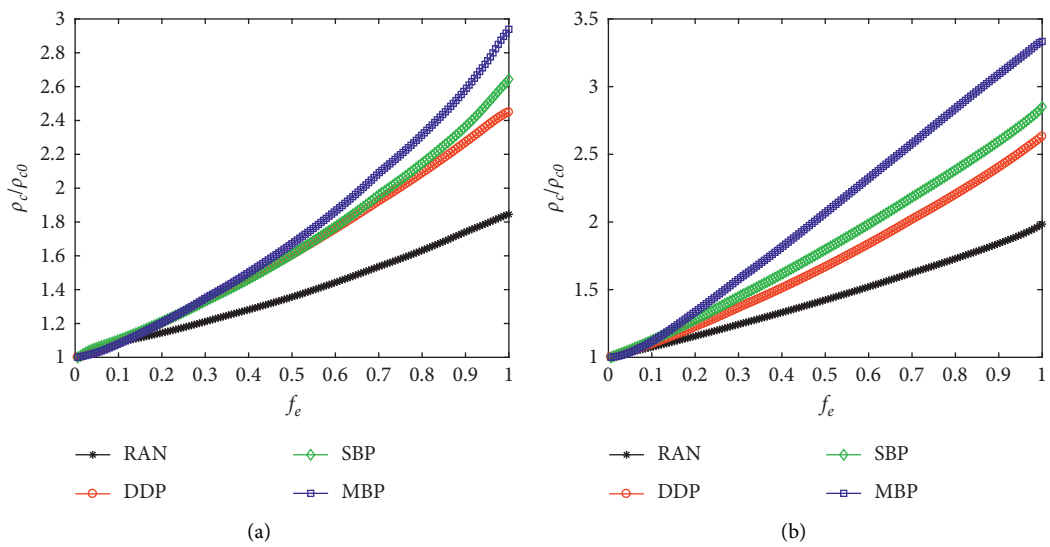


Figure 6: The enhancement of our strategies in a real multiplex network. (a) In "Work" network. (b) In "Lunch" network.

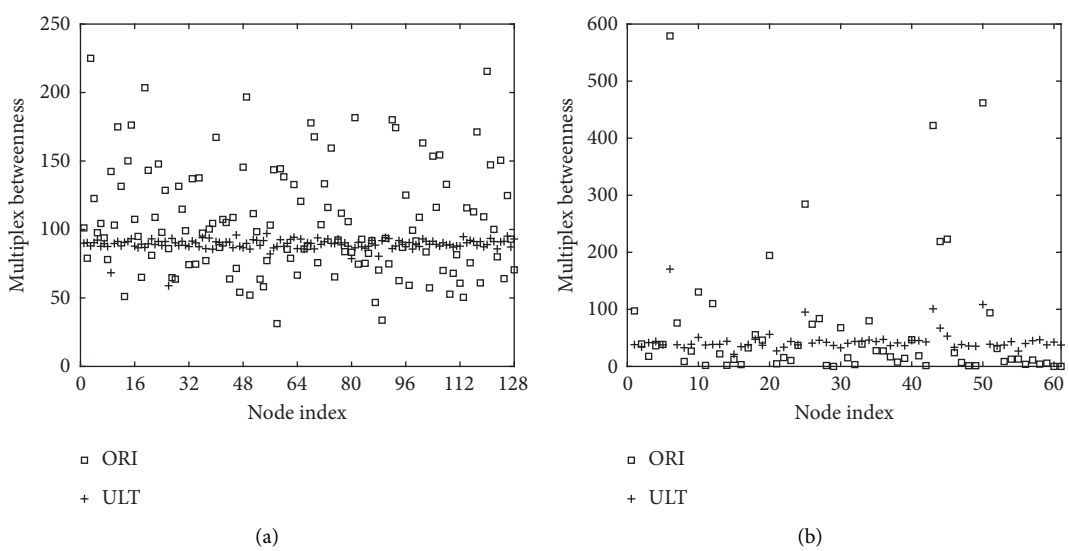


Figure 7: The multiplex betweenness of each node before and after applying MBP strategy. (a) The computer-generated multiplex network with (N) = 128, (m) = 4, <(k)> = 16, and $(Z)_{in} = 8$ in layer 1 and $(Z)_{in} = 14$ in layer 2. (b) The real CS-Aarhus multiplex network.

4. Conclusions

In order to enhance the transfer capacity of multiplex systems, we propose some strategies by adding edges according to different weights. By checking the critical packet injection rate of the multiplex network, we discover that our strategies are capable of enhancing the transfer capacity significantly. The MBP which adds edges with the highest product of multiplex betweenness is more effective than the others. The impacts of different topology characteristics, such as the community structure, the average degree, the communities number, and the nodes number, are explored to find that our strategies are more effective in multiplex networks with pronounced community structure. And in a two-layer network with different community structures, applying our strategies in the layer with the stronger community structure can yield better results. Our strategies are proved to be very effective by simulation results of computer-generated networks and real-world systems. The time complexity of our strategies is also acceptable which means our strategies might be helpful in developing more efficient transfer networks and routing strategies.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

F. SHAO thanks Dr. Albert Solé Ribalta of the Complex Systems group at IN3 for sharing the source code of references [17] and [25] to validate the betweenness of the multiplex network. This research was funded by the Natural Science Foundation of the Jiangsu Higher Education Institutions of China (Grant nos. 17KJA520001, 18KJA520003, and 19KJA510004) and the National Natural Science Foundation of China (No. 61375121). Also, the APC was funded by Jinling Institute of Technology.

References

[1] D. J. Watts and S. H. Strogatz, "Collective dynamics of "small-world" networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[2] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[3] L. Zhao, Y. C. Lai, K. Park, and N. Ye, "Onset of traffic congestion in complex networks," *Physical Review E*, vol. 71, no. 2, Article ID 026125, 2005.

[4] R. Guimerà, A. Díaz-Guilera, F. Vega-Redondo et al., "Optimal network topologies for local search with congestion," *Physical Review Letters*, vol. 89, no. 24, pp. 248–701, 2002.

[5] G. Yan, T. Zhou, B. Hu et al., "Efficient routing on complex networks," *Physical Review E*, vol. 73, no. 4, Article ID 046108, 2006.

[6] Z. Toroczkai and K. E. Bassler, "Jamming is limited in scale-free systems," *Nature*, vol. 428, no. 6984, p. 716, 2004.

[7] R. Ding, "The complex network theory-based urban land-use and transport interaction studies," *Complexity*, vol. 2019, 2019.

[8] G.-Q. Zhang, S. Zhou, D. Wang, G. Yan, and G.-Q. Zhang, "Enhancing network transmission capacity by efficiently allocating node capability," *Physica A: Statistical Mechanics and Its Applications*, vol. 390, no. 2, pp. 387–391, 2011.

[9] A. Solé-Ribalta, C. Granell, S. Gómez, and A. Arenas, "Information transfer in community structured multiplex networks," *Frontiers in Physics*, vol. 3, no. 61, 2015.

[10] M. E. J. Newman, "Communities, modules and large-scale structure in networks," *Nature Physics*, vol. 8, no. 1, pp. 25–31, 2012.

[11] S. Fortunato, "Community detection in graphs," *Physics Reports*, vol. 486, no. 3-5, pp. 75–174, 2010.

[12] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, no. 2, Article ID 026113, 2004.

[13] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Physical Review E*, vol. 69, no. 6, Article ID 066133, 2004.

[14] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002.

[15] L. Danon, A. Arenas, and A. Díaz-Guilera, "Impact of community structure on information transfer," *Physical Review E*, vol. 77, no. 3, Article ID 036103, 2008.

[16] J. Cai, Y. Wang, Y. Liu, J.-Z. Luo, W. Wei, and X. Xu, "Enhancing network capacity by weakening community structure in scale-free network," *Future Generation Computer Systems*, vol. 87, pp. 765–771, 2018.

[17] A. Solé-Ribalta, M. D. Domenico, S. Gómez, and A. Arenas, "Centrality rankings in multiplex networks," in *Proceedings of the 2014 ACM Conference on Web Science*, ACM, Bloomington, IN, USA, June 2014.

[18] G. F. d. Arruda, F. A. Rodrigues, and Y. Moreno, "Fundamentals of spreading processes in single and multilayer complex networks," *Physics Reports*, vol. 756, 2018.

[19] A. Solé-Ribalta, S. Gómez, and A. Arenas, "Congestion induced by the structure of multiplex networks," *Physical Review Letters*, vol. 116, no. 10, Article ID 108701, 2016.

[20] M. D. Domenico, C. Granell, M. A. Porter, and A. Arenas, "The physics of spreading processes in multilayer networks," *Nature Physics*, vol. 12, no. 10, p. 901, 2016.

[21] E. Cozzo and Y. Moreno, "Characterization of multiple topological scales in multiplex networks through supra-Laplacian eigengaps," *Physical Review E*, vol. 94, no. 5, 2016.

[22] M. D. Domenico, A. Solé-Ribalta, E. Cozzo et al., "Mathematical formulation of multilayer networks," *Physical Review X*, vol. 3, no. 4, Article ID 041022, 2013.

[23] L. Rossi and M. Magnani, "Towards effective visual analytics on multiplex and multilayer networks," *Chaos, Solitons & Fractals*, vol. 72, pp. 68–76, 2015.

[24] X. Zhu, J. Ma, X. Su et al., "Information spreading on weighted multiplex social network," *Complexity*, vol. 2019, no. 15, Article ID 5920187, 2019.

[25] A. Solé-Ribalta, A. Arenas, and S. Gómez, "Effect of shortest path multiplicity on congestion of multiplex networks," *New Journal of Physics*, vol. 21, no. 3, Article ID 035003, 2019.

[26] C. Rahmede, J. Iacovacci, A. Arenas, and G. Bianconi, "Centralities of nodes and influences of layers in large multiplex networks," *Journal of Complex Networks*, vol. 6, no. 5, pp. 733–752, 2017.

[27] A. Solé-Ribalta, M. De Domenico, S. Gómez, and A. Arenas, "Random walk centrality in interconnected multilayer networks," *Physica D: Nonlinear Phenomena*, vol. 323-324, pp. 73–79, 2016.

[28] G. Q. Zhang, D. Wang, and G. J. Li, "Enhancing the transmission efficiency by edge deletion in scale-free networks," *Physical Review E*, vol. 76, no. 1, Article ID 017101, 2007.

[29] A. Arenas, A. Díaz-Guilera, and R. Guimerà, "Communication in networks with hierarchical branching," *Physical Review Letters*, vol. 86, no. 14, pp. 3196–3199, 2001.

[30] L. C. Freeman, "A set of measures of centrality based on betweenness," *Sociometry*, vol. 40, no. 1, pp. 35–41, 1977.

Hindawi
Complexity
Volume 2020, Article ID 6660117, 13 pages
https://doi.org/10.1155/2020/6660117

Research Article

A Cluster-Head Rotating Election Routing Protocol for Energy Consumption Optimization in Wireless Sensor Networks

Jun Wang,[1] **Zhuangzhuang Du**,[2] **Zhengkun He**,[3] **and Xunyang Wang**[4,5]

[1]*School of Electrical Engineering, Henan University of Science and Technology, Luoyang, Henan 471000, China*
[2]*School of Agricultural Equipment Engineering, Henan University of Science and Technology, Luoyang, Henan 471003, China*
[3]*School of Computer Science and Engineering, Central South University, Changsha, Hunan 410000, China*
[4]*Department of Applied Mathematics, Lanzhou University of Technology, Lanzhou, Gansu 730050, China*
[5]*Postdoctoral Research Station in Gansu Electric Power Research Institute, Lanzhou, Gansu 730000, China*

Correspondence should be addressed to Xunyang Wang; 12198114@163.com

Received 31 October 2020; Revised 2 December 2020; Accepted 10 December 2020; Published 21 December 2020

Academic Editor: Yongsheng Hao

Balancing energy consumption using the clustering routing algorithms is one of the most practical solutions for prolonging the lifetime of resource-limited wireless sensor networks (WSNs). However, existing protocols cannot adequately minimize and balance the total network energy dissipation due to the additional tasks of data acquisition and transmission of cluster heads. In this paper, a cluster-head rotating election routing protocol is proposed to alleviate the problem. We discovered that the regular hierarchical clustering method and the scheme of cluster-head election area division had positive effects on reducing the energy consumption of cluster head election and intracluster communication. The election criterion composed of location and residual energy factor was proved to lower the probability of premature death of cluster heads. The chain multihop path of intercluster communication was performed to save the energy of data aggregation to the base station. The simulation results showed that the network lifetime can be efficiently extended by regulating the adjustment parameters of the protocol. Compared with LEACH, I-LEACH, EEUC, and DDEEC, the algorithm demonstrated significant performance advantages by using the number of active nodes and residual energy of nodes as the evaluation indicators. On the basis of these results, the proposed routing protocols can be utilized to increase the capability of WSNs against energy constraints.

1. Introduction

As the cornerstone of the system of the Internet of Things, wireless sensor networks (WSNs) are distributed network systems in which numerous microsensor nodes cooperate to detect, process, and transmit various information of interest in the way of wireless [1, 2]. Due to the characteristics of low cost, rapid deployment, self-organization, and high fault tolerance, WSNs have been widely used in numerous fields, such as military reconnaissance, environmental detection, agricultural production, and medical treatment [3–5]. Generally, the nodes powered by limited energy resource (batteries) are deployed in an unattended harsh environment, and it is virtually impracticable to replace or charge the depleted batteries after a long run [6, 7]. Therefore, in views of sustainability and quality of data acquisition, energy consumption reduction has become an essential issue for WSNs to lengthen the network lifetime.

Energy-efficient transmission and data aggregation mechanisms are critical subjects that cannot be ignored for energy saving in WSNs [8–10]. The primary aims to be achieved involve reducing the total energy consumption, decreasing the number of data communications, enhancing the number of active nodes over a certain period of operations, and balancing the energy dissipation of nodes [11–13]. Hierarchical cluster-based routing protocols have been considered as the most effective network organization scheme in improving the energy-efficiency for WSNs [14–16]. Recently, a variety of cluster routing algorithms of this type have been introduced to cope with the problems of

uneven load distribution among nodes and strict energy constraints [17, 18]. LEACH (low-energy adaptive cluster hierarchical) is the most classic clustering protocol [19]. However, with the increase of deployment scale, the efficiency of the protocol declines dramatically due to the single-hop communication from cluster heads (CHs) to the base station (BS) and the possibility of low-power nodes being repeated as CHs [20, 21]. Several dynamic CH role rotation algorithms have been suggested to eliminate the deficiencies of LEACH by multihops and energy awareness, including I-LEACH (improved low-energy adaptive cluster hierarchical) [22], EEUC (energy-efficient uneven clustering) [23], HEED (hybrid energy-efficient distribution) [24], DEEC (distributed energy-efficient clustering) [25], and DDEEC (developed distributed energy-efficient clustering) [26]. These block clustering-based protocols can alleviate unbalanced energy consumption through CH selection based on residual energy and more relevant criteria. Meanwhile, the mechanism of time-driven CH candidate is verified to be effective, easy to implement, and of low complexity [27, 28]. Nevertheless, the participation of each node in the CH election phase will unavoidably induce unnecessary energy loss, and it is remarkably challenging to obtain a satisfactory energy-efficiency of intracluster communication for the chosen CHs individually regarding their own location, energy level, or other related information [29–31]. Moreover, the irregular cluster distribution causes the intercluster communication path difficult to be optimal.

In addition, a series of chain clustering-based algorithms are exploited to increase the lifespan of the network along with sustainable scalability, such as PEGASIS (power-efficient gathering in sensor information systems) [32], CCM (chain cluster-based mixed) routing [33], and CCMAR (cluster-chain mobile agent routing) [34]. The use of chain-based routing protocols can significantly prolong network lifetime by minimizing transfer distance between nodes and avoiding the energy overhead of periodic head voting with a chain topology [35]. However, these algorithms suffer from colossal data delay and are not proper for large scale networks [36]. Through the above analysis, it can be inferred that the combination of advantages of both block clustering-based and chain clustering-based protocols is undoubtedly practicable and reliable option for energy-efficiency maximization in WSNs. Thus, we propose a cluster-head rotating election routing protocol (CHRERP) to efficiently manage energy consumption in this study. The main contributions of this study are summarized as follows:

(1) To reduce the number of nodes competing for CHs and the energy overhead in intracluster communication, the sensing area is segmented into multiple clusters by regular hierarchical pattern, and the central region in each cluster is utilized as the CH election area.

(2) The periodic time-based rotation of clusters and CH candidate areas is used to change cluster member composition and regulate node energy distribution dynamically.

(3) The node score evaluated by location and residual energy is adopted to select the CHs, and the chain shortest path with optimized intercluster communication dissipation is applied for data aggregation from clusters to the BS.

The remainder of this paper is arranged as follows. The network model and the energy consumption model are given in Section 2. The details of the presented routing algorithm are displayed in Section 3. The evaluations on the performances of the routing protocol and the impacts of key parameters are exhibited in Section 4. Finally, conclusions are shown in Section 5.

2. Preliminaries

2.1. Network Model. The network model used in this study is a WSN model in which N nodes are randomly deployed in a circular sensing area centered on a BS. The BS has strong computing and network management capabilities and is equipped with more battery power or can be self-replenished through energy harvesting. Hence, the BS can keep on working until all nodes are dead. On this basis, the following assumptions are made about the WSN.

(1) All nodes are homogeneous, stationary, and energy-constrained. The initial energy of each node is equal and expressed as E_0. Each node is assigned a unique identifier (ID) and can collect data packets from the cluster members when acting as a CH. CHs transmit data packets to the BS in single or multiple hops. In addition, the data packets are considered to be successfully transmitted upon arrival at the BS.

(2) The BS is aware of the location of every node after the network deployment. Each node stores the locations of other nodes and organization information in its database at the initial stage of the network through the BS's flooding broadcast.

(3) Proper medium access control methods (e.g., CDMA-based or contention window-based technologies) are applied to accomplish multiple simultaneous wireless transmissions.

2.2. Energy Consumption Model. According to the actual transmission distance from the CHs to the BS, the free space model and the multipath fading channel model both need to be comprehensively investigated; therefore, the extended model proposed in [37] is adopted in our study for representing communication energy consumption in consideration of path loss. Either the free space (d^2 power loss) or the multipath fading (d^4 power loss) channel models are included. The required energy for transmitting a m-bit packet is expressed as

$$E_T(m,d) = \begin{cases} E_{\text{elec}} \times m + m \times \varepsilon_{\text{fs}} d^2, & d < d_0, \\ E_{\text{elec}} \times m + m \times \varepsilon_{\text{mp}} d^4, & d \geq d_0, \end{cases} \quad (1)$$

where E_{elec} is the energy consumed per bit by the transmitter or receiver circuit, which depends on factors such as the

digital coding, modulation, filtering, and propagation of the radio signal. d is the distance between the transmitter and the receiver, and d_0 is utilized as the distance threshold. Moreover, in the case that $d < d_0$, we apply the free space model and ε_{fs} indicates the energy coefficient per bit. Otherwise, the multipath fading channel model is used, and ε_{mp} depicts the energy coefficient per bit.

The energy required to receive the data information of m bits is defined as follows:

$$E_R(m) = E_{elec} \times m. \tag{2}$$

3. Cluster-Head Rotating Election Routing Strategy

3.1. Cluster-Head Election Area. At the initialization stage of network, the BS evenly divides the sensing area into z sector districts and stratifies the sensing area with itself as the center according to different radiuses. The quantity of z directly reflects the density of clusters, and the greater the z value, the more clusters there are in the network. The region formed by the concentric rings of adjacent layers and the sector radius lines is regarded as a cluster area. Furthermore, within each sector district, a sector with a central angle of α is subdivided as the district's CH election area, and the part surrounded by the sector and each cluster area is further designated as the CH election area for the cluster. The position of CH election area is in the middle of the corresponding district (Figure 1).

For balancing energy consumption among nodes in a cluster, the CH rotating election method is adopted to decrease the possibility of an individual node being repeatedly selected as a CH. Specifically, all the districts and CH election areas rotate synchronically. Since each node can determine the cluster members after each rotation by the locations of the other nodes in the network and the rotation angle, non-extra-energy consumption of new cluster formation will generate. β is used to represent the counterclockwise rotation angle after the preset rounds of data collection (Figure 2).

Thus, the centerline angle of the nth district, the center angle range of the nth CH selection area, and the position of the nth CH selection area after the p-round rotations can be computed by the following equations, respectively:

$$\theta_{Middle}(n) = \frac{2\pi n}{z} - \frac{\pi}{z}, \quad n \in (1, 2, 3, \ldots, z), \tag{3}$$

$$\theta_{Election}(n) = \left[\frac{2\pi n}{z} - \frac{\pi}{z} - \frac{\alpha}{2}, \frac{2\pi n}{z} - \frac{\pi}{z} + \frac{\alpha}{2}\right], \tag{4}$$

$$\theta_{Election}(p) = \left[\frac{2\pi n}{z} - \frac{\pi}{z} - \frac{\alpha}{2} + p\beta, \frac{2\pi n}{z} - \frac{\pi}{z} + \frac{\alpha}{2} + p\beta\right], \quad p \in z^+. \tag{5}$$

3.2. Hierarchical Pattern. Nodes close to the BS inevitably need to undertake more forwarding tasks. In this study, the competitive range of CH election of each layer is adjusted using the circular layered interval proportional to the distance from the BS (Figure 3), making up for the excessive energy consumption of CHs near to the BS. The hierarchical way can be expressed as follows:

$$\begin{cases} R_1 = r_0, & i = 1, \\ R_{i-0} = \varepsilon i r_0, & i \geq 2, \\ R_i = \varepsilon i r_0 - \varepsilon(i-1) r_0, & i \geq 2, \end{cases} \tag{6}$$

where R_1 denotes the radius of the first layer of the network, R_{i-0} represents the distance from the ith layer to the BS, R_i depicts the width of the circular area of the ith layer, R_0 implies the initial radius of the first layer, and ε $(0.5 \leq \varepsilon \leq 1.5)$ signifies the radius coefficient.

3.3. Cluster-Head Election Criterion. Following the network partitioned into clusters, it becomes a critical issue to select a suitable CH for each election area. Choosing the node near the centerline of the election area as the CH is beneficial to balance the transmission energy consumption within the cluster and reduce the communication distance between clusters. Nonetheless, considering only the location factor and ignoring the residual energy of the selected CHs, it is easy to cause premature failure for the low-energy CHs. In this study, the distance between the node and the corresponding centerline of the election area and the node residual energy are used as the candidate parameters for CH. The node score function for CH selection criteria can be depicted as follows:

$$CH_j = (1 - \chi)\frac{D_{max} - D_{jm}}{D_{max}} + \chi\frac{E_j}{E_0}, \tag{7}$$

where CH_j is the score of node j, χ $(0 \leq \chi \leq 1)$ is the weight adjustment coefficient, D_{max} is the half of the outer boundary length of the CH election area where node j is located, D_{jm} is the vertical distance from node j to the centerline of the election area, E_j is the residual energy of node j, and E_0 is the initial energy of node j. The node with the highest score for each election area is selected as the CH of the corresponding cluster.

3.4. Communication Process. After each rotation of districts and CH election areas, the nodes in each election area compute their scores associated with location and residual energy to decide whether they can become a CH. The chosen CHs in the same district then establish a chain communication link from outside to inside to deliver data packets collected from each cluster to the BS. The overall process of network communication is shown in Algorithm 1. Figure 4 illustrates the simulation result of the scenario ($N = 200$, $z = 4$, $i = 5$, and $\beta = 15°$). As demonstrated in Figure 4(a), in the primary nonrotation phase, the selected CH of each

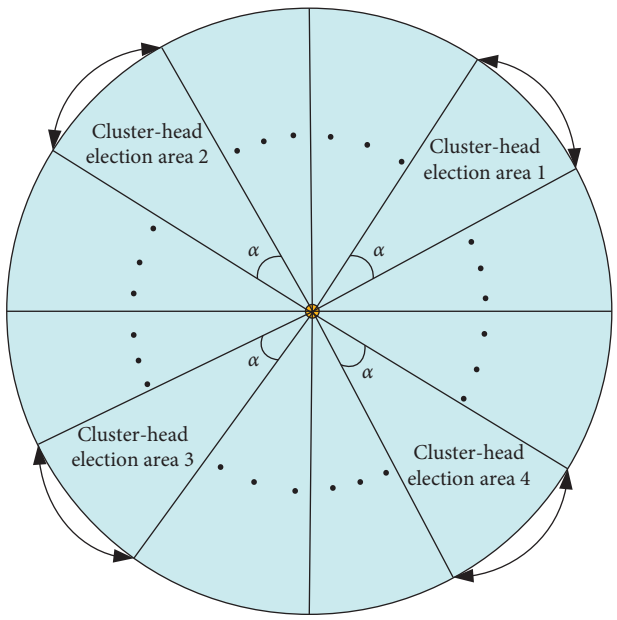


Figure 1: Initial distribution of CH election areas ($z = 4$).

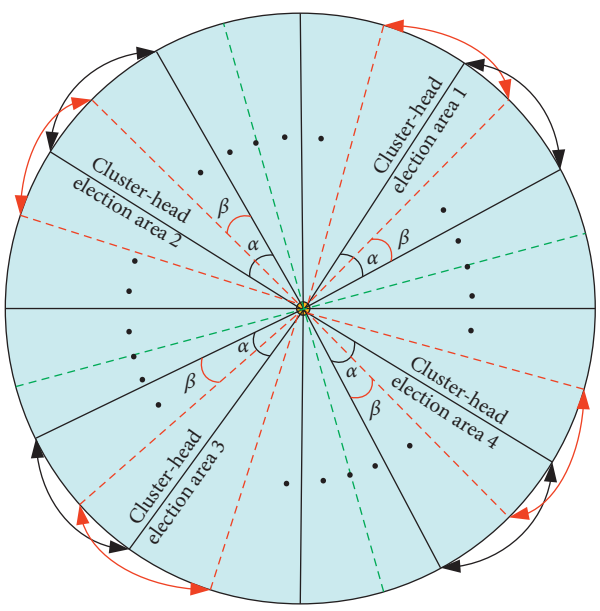


Figure 2: Rotation of CH election areas ($z = 4$, $\beta = 5^{\circ}$).

cluster is basically distributed near the centerline of the election area. After 45 rotations, the elected CHs obviously deviate from the centerlines with the change of node residual energy (Figure 4(b)).

4. Simulation Results and Analysis

To assess the performance of CHRERP in terms of network lifetime, relevant experiments are carried out with the help of MATLAB R2019a. The comparative tests with LEACH, I-LEACH, EEUC, and DDEEC are conducted under the same conditions, which are four cluster-based routing protocols in WSNs. The values of the experimental parameters are shown in Table 1.

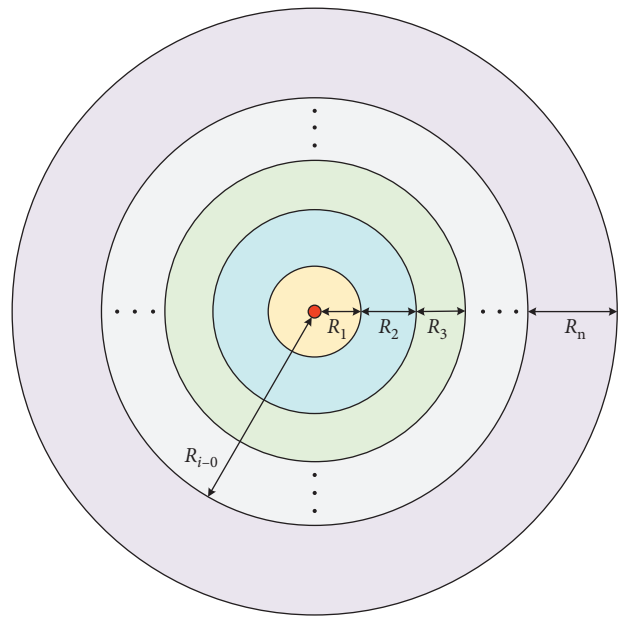


Figure 3: Heterogeneous multilayered network model.

In the simulations, the influences of parameters ε, z, χ, α, and β on network lifetime are analyzed. The indexes, including the death round of the first node, the death round of half nodes, the death round of 80% nodes, and the residual energy after 80% node death, are used as assessment metrics.

Figure 5 depicts the assessment metrics with varying ε in the case of $z = 3$, $\chi = 0.9$, $\alpha = 30^{\circ}$, and $\beta = 5^{\circ}$. As manifested in Figure 5(a), with the increase of ε, the death round of the first node presents a decreasing trend, and the death round of both half and 80% nodes first increases and then decreases. When the death round of half and 80% nodes reaches the peak, ε is 1 and 2, respectively. The results show that, in the case that CH election areas do not rotate, the death round of the first node is inversely proportional to the deployment area of WSNs, which is consistent with most conventional algorithms. With the increase of simulation rounds, CHRERP can dramatically decrease the communication burden induced by the increase of the area, and an optimal interval of ε exists for a certain number of nodes. Moreover, the maximum of the weighted sum of death rounds is achieved when $\varepsilon = 2$, in the circumstance of the weight of the death round of the first node, half nodes, and 80% nodes is conventionally set to 0.1, 0.3, and 0.6, respectively. As exhibited in Figure 5(b), as ε goes from 0 to 4, the residual energy after 80% node death first increases and then decreases. The maximum residual energy is obtained in the case that ε is equal to 3, denoting that when ε is within the range [0, 3], the impact of ε on the residual energy is more significant than that of the growth of the sensing area. Nevertheless, as ε continuously grows, the energy consumption of nodes in the same cluster gradually increases with the rise of the sensing area, and the residual energy after 80% node death presents a downward trend. The size of the sensing area, interlayer distance, and clustering size can be

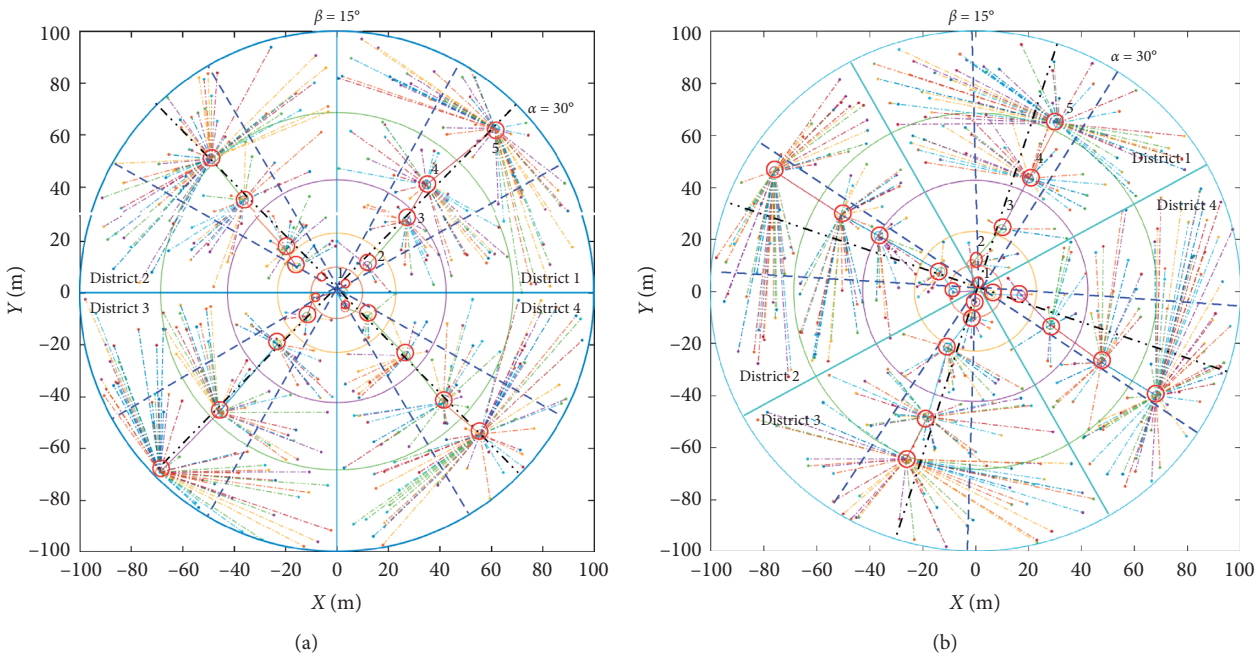


Figure 4: Illustration of network communication. (a) Nonrotation and (b) 45th rotation.

Input: Number of sector districts z, central angle of each CH election area α, weight adjustment coefficient χ, radius of the first layer r_0, number of layers i, radius coefficient ε, rotation angle β, location of each node, and number of data acquisition rounds per rotation, node morality threshold.
Output: Number of active nodes, residual energy.
Initialization: BS broadcasts the hierarchical clustering instruction. According to location, all the nodes are divided into a total of iz clusters numbered counterclockwise in ascending order.

```
While Node death rate is below the set threshold do
    CH election, intra-cluster communication, inter-cluster communication.
  for β ⟵ 0 < θ ≤ 2π/z do
    repeat
      for j ⟵ Nodes in each CH election area do
        Calculate the score of nodes through equation (7);
      end for
    until Choose the node with the maximum score as the CH for each cluster; Execute pre-determined rounds of data
acquisition; Rotate districts and CH election areas β degrees counterclockwise;
    repeat
      for Nodes within each cluster do
        Nodes transmit the data packets to the corresponding CH by single hop;
      end for
    until Each CH receives data packets from every active node in the cluster;
    repeat
      for CHs in the same district do
        CHs transfer data packets from outside to inside;
      end for
    until BS obtains data packets from the CHs in the first layer;
  end for
end
```

Algorithm 1: Network communication procedure.

modified by regulating ε. It can be noticed that for the node number arranged in the simulation, ε between 1 and 2.5 has a positive effect on the number of simulation rounds, and ε varying from 2.0 to 3.0 can visibly enhance the residual energy. Furthermore, it can be inferred that an appropriate ε can adequately balance the node energy consumption of the whole network and extend the network lifetime. Hence, in the following simulation analysis, we choose ε as 2 hereafter

to determine the size of the sensing area and interlayer distance.

Figure 6 presents the assessment metrics with varying z under the case that $\varepsilon = 2$, $\chi = 0.9$, $\alpha = 30^\circ$, and $\beta = 5^\circ$. From Figure 6(a), It can be noted that with the rise of z, the number of death rounds declines noticeably. When $z = 5$, compared with the case of $z = 3$, the death round of the first node, half nodes, and 80% nodes reduces by 35.3%, 39.6%, and 57.5%, respectively. The growth of z will lead to the generation of more CHs. When the total number of nodes remains unchanged, the number of nodes in each cluster decreases relatively. As the number of simulation rounds rises, the probability of node being repeatedly selected as CH increases. Due to excessive energy consumption, the CHs will die prematurely while in the condition that z is small, the number of CHs in the network shrinks, and the number of nodes in each cluster increases, which can dramatically lower the chance of node becoming CH multiple times and boost the number of death rounds. It can be seen from Figure 6(b) that when z shifts from 3 to 8, the residual energy after 80% node death exhibits the characteristic of fluctuation, and when z is 3, the residual energy reaches the maximum. Therefore, for $z = 3$, the performance of assessment metrics is optimal, indicating that the CHRERP proposed in this paper are more suitable for the case with smaller z. Therefore, to optimize the network lifetime, we should reasonably adjust the network partition scale and avoid the premature death of partial nodes prompted by redundant CHs.

Figure 7 represents the effect of χ on the assessment metrics when $z = 3$, $\varepsilon = 2$, $\alpha = 30^\circ$, and $\beta = 5^\circ$. As shown in Figure 7(a), when χ changes between 0.3 and 0.7, the number of death rounds fluctuated slightly. While χ rises from 0.1 to 0.2 and 0.8 to 0.9, the death round of the first node, half nodes, and 80% nodes increases by −9, 918.4, 2087.6 and 22.2, 60.4, 419.4, respectively. Compared to $\chi = 0.9$, when $\chi = 0.2$, the number of the three death rounds grows by −73.8%, 30.6%, and 32.5%, respectively. It can be perceived that the case that χ equals 0.2 can adequately balance the weight relationship between node residual energy and the distance from node to the centerline of the candidate area and remarkably prolong the network lifetime. As presented in Figure 7(b), when χ is within the range of [0, 0.2], the residual energy after 80% node death is between 70 and 85 J. The residual energy declines drastically when χ varies from 0.3 to 1, proving that the smaller χ is more beneficial to improve the network energy-efficiency. It suggests that in the process of network operation, the effect of location factor on CH selection should be reinforced as much as possible.

Figure 8 expresses the impact of α on the assessment metrics in the circumstance that $\varepsilon = 2$, $z = 3$, $\chi = 0.9$, and $\beta = 5^\circ$. As seen in Figure 8(a), with the rise of α from 30° to 45°, the number of death rounds exhibits a downward trend as a whole. The death round of the first node, half nodes, and 80% nodes decreases by 1.9, 18.1, and 51.7 per degree, respectively. This decrease may be caused by the increase of energy consumption from the increment of participating nodes in the CH election and the path extension of intercluster communication. It can be seen from Figure 8(b) that when α increases from 30° to 45°, the residual energy after 80% node death remains between 35 and 40 J, inferring that α has little influence on the energy consumption of nodes. The simulation results show that a small α is in favor of balancing the network energy dissipation. In contrast, a larger α denotes the rise of each CH election area, definitely leading to the energy consumption increment of CH candidate process.

Table 1: Parameter values.

Parameter	Value
E_0	0.5 J
E_{elec}	50 nJ/bit
ε_{fs}	10 pJ/bit/m^2
ε_{mp}	0.0013 pJ/bit/m^4
d_0	87 m
r_0	30 m
I	5
Packet size	4000 bits
α	$30^\circ \le \alpha \le 45^\circ$
β	$5^\circ \le \beta \le 15^\circ$
z	$3 \le z \le 8$
ε	$0.5 \le \varepsilon \le 4$
χ	$0 \le \chi \le 1$
Number of nodes	1500
Number of data acquisition rounds per rotation	10

Figure 9 exhibits the relationships between β and the assessment metrics when $\varepsilon = 2$, $z = 3$, $\chi = 0.9$, and $\alpha = 30^\circ$. As observed in Figure 9(a), the variations of the three death rounds present notable differences. The death round of the first node performs an overall downward trend, and the number of death rounds reduces by 3.2 for per degree increment. The fluctuation of the death round of half nodes is stable, and the value changes around 1300 rounds. The death round of 80% nodes lowers sharply under the condition that β grows from 13° to 15°. The results indicate that a continuous increase in β will ultimately cause a drastic reduction in the number of simulation rounds. It can be recognized from Figure 9(b) that, in the case that β is within the range of 5° to 13°, the residual energy after 80% node death maintains between 35 and 40 J, while β jumps to 15°, and the residual energy falls below 35 J. To sum up, $\beta = 5^\circ$ can lengthen the network lifetime compared with a larger β. It may be caused by the fact that with the enlargement of β, the intersection of candidate areas decreases before and after each rotation, then the selected CH will unavoidably deviate from the centerline region due to the inconspicuous residual-energy advantage of nodes near the centerline for the updated election area.

Figure 10 displays the combination effect of χ and α on assessment metrics in the condition that $\varepsilon = 2$, $z = 3$, and $\beta = 5^\circ$. From Figure 10(a), when χ and α grow from 0.1 to 0.7 and from 30° to 45°, respectively, the fluctuation scale of each death round of the first node is little and has approximate waveform characteristics. When χ constantly rises from 0.7 and α is in the range of 30° to 45°, the death round enhances visibly, and [0.9, 30°] is the optimal combination of χ and α.

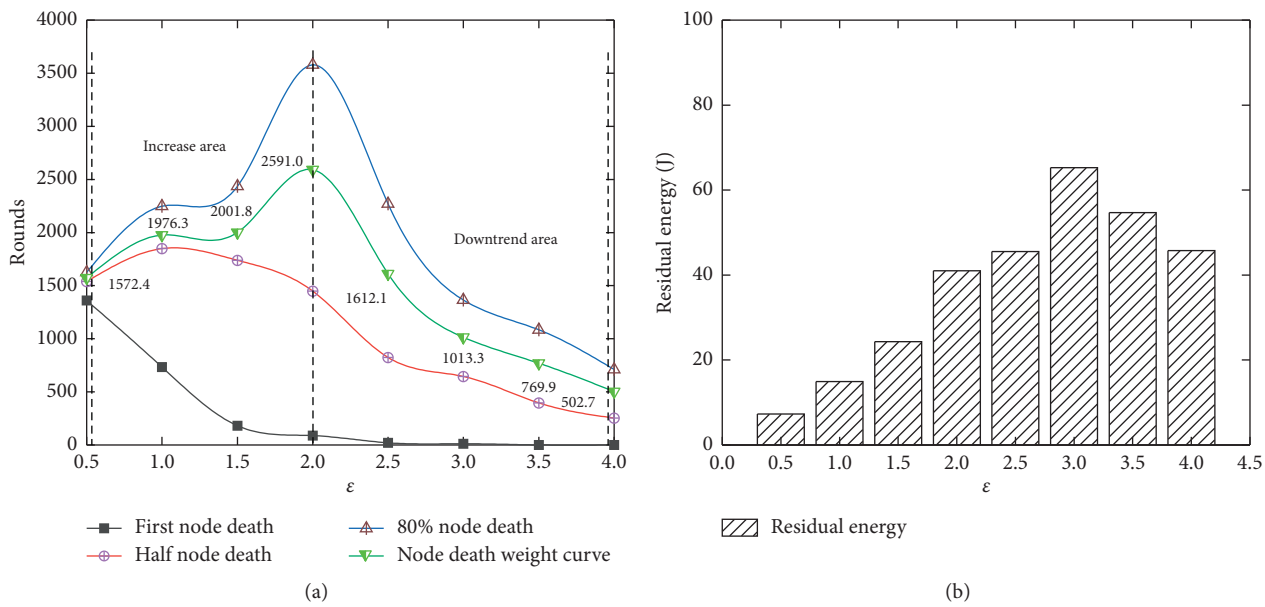


Figure 5: Assessment metrics with varying ε: (a) death round with varying ε and (b) residual energy with varying ε.

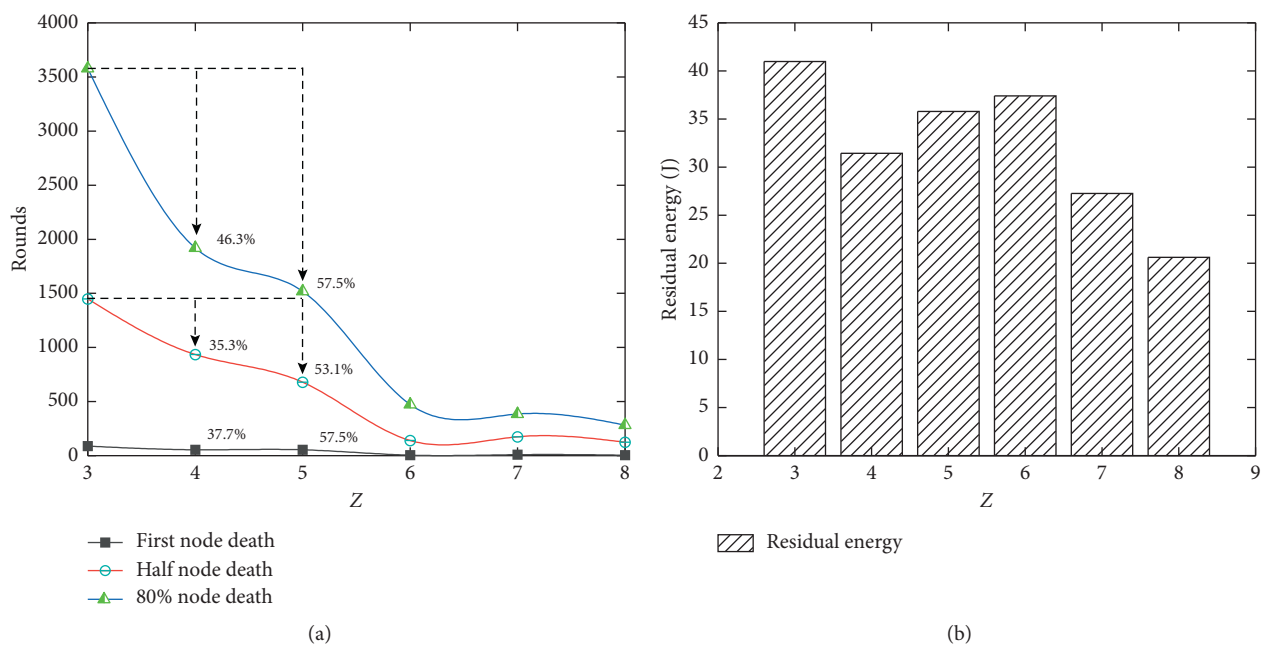


Figure 6: Relationships between z and assessment metrics: (a) influence of z on death round and (b) influence of z on residual energy.

In Figure 10(b), while χ changes from 0.3 to 0.7 and α rises from 30° to 45°, the death round of half nodes varies slightly. When χ increases from 0.7 to 0.9, the death round begins to ascend. However, compared with $\chi = 0.2$, the number of death rounds drops greatly, and the combination of [0.2, 30°] manifests the best performance. The identical fluctuation characteristics with Figure 10(b) can be discovered in Figure 10(c), and [0.2, 30°] is also the combination with the most extended network lifetime. As presented in Figure 10(d), when χ belongs to the range of [0.2, 0.9], the residual energy after 80% node death overall decreases and the most excellent performance is achieved at [0.2, 30°]. Obviously, the larger χ can increase the death round of the first node, while the smaller χ will unavoidably improve the death round of half nodes and 80% nodes. It can be easily concluded that the combination of χ and α with small values is more valuable for the extension of network lifetime.

Figure 11 manifests the combination influence of χ and β for assessment metrics when $\varepsilon = 2$, $z = 3$, and $\alpha = 30°$. It can be seen from Figure 11(a) that the death round of the first node declines by about 2.5 rounds on average per degree increment in β. When χ grades from 0.1 to 0.7, the death

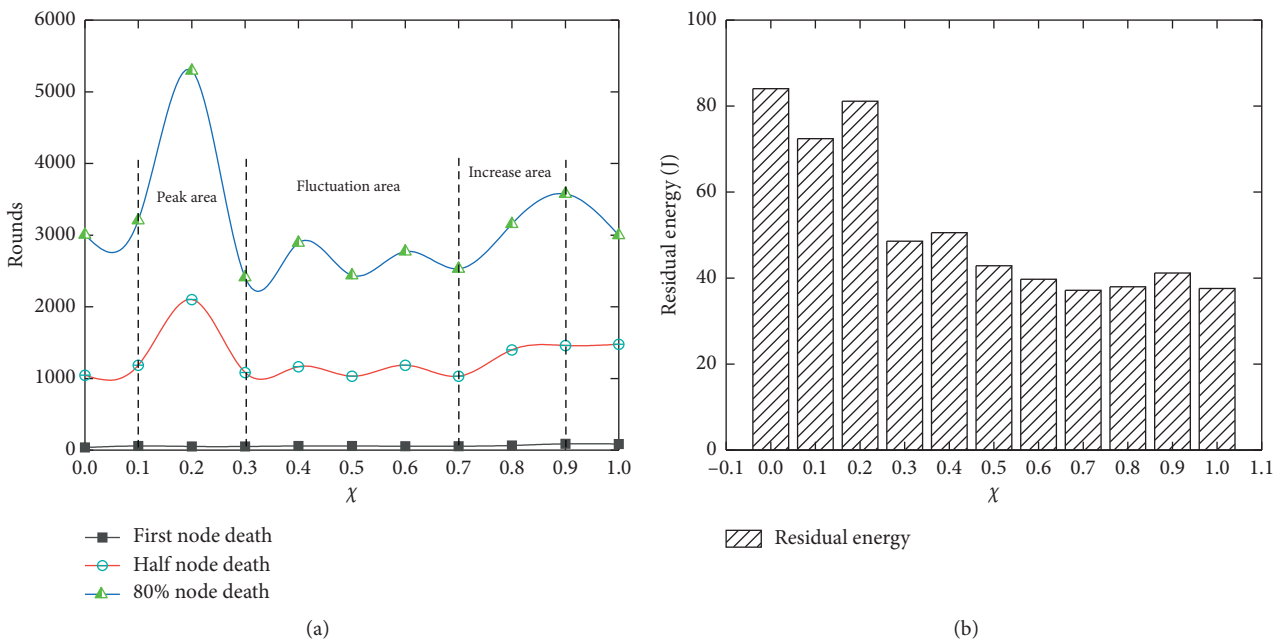


Figure 7: Assessment metrics with varying χ: (a) impact of χ on death round and (b) impact of χ on residual energy.

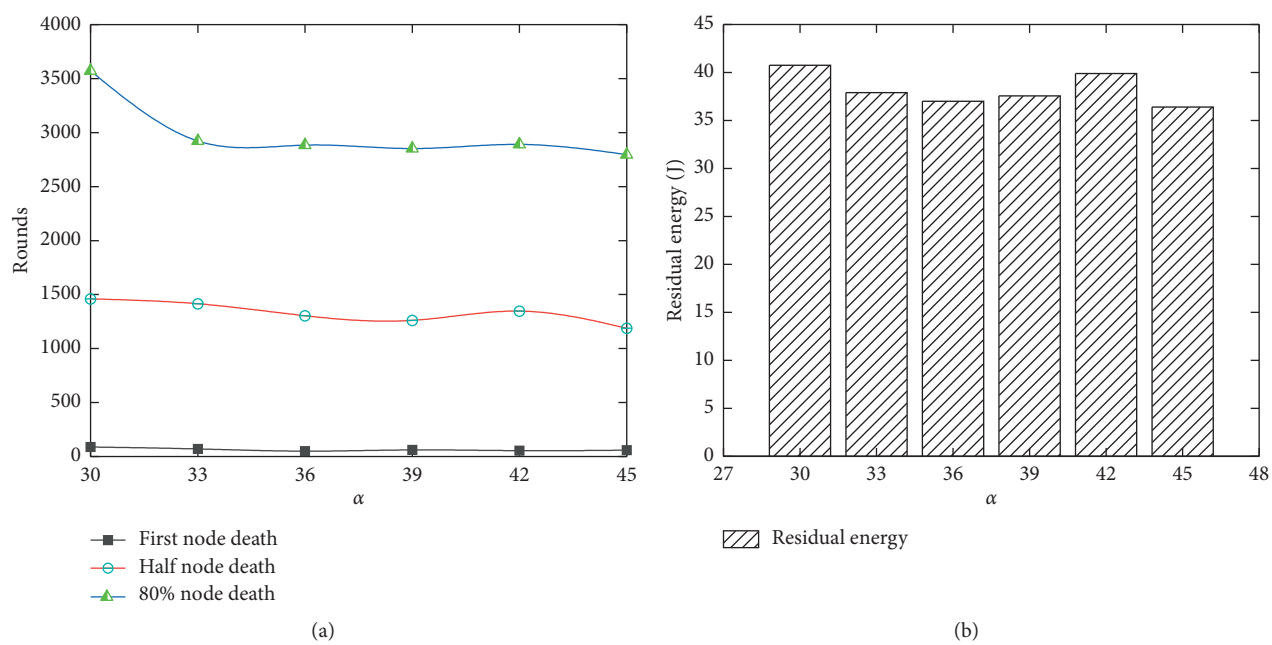


Figure 8: Affecting relations between α and assessment metrics: (a) death round with varying α and (b) residual energy with varying α.

round is stable. In addition, χ increases from 0.7 to 0.9, the number of death rounds for different β increases by 58.8%, 62.5%, 60.7%, 75.2%, 79.6%, and 103.3%, respectively, the combination parameter of [0.9, 5°] is the most proper selection. Figures 11(b) and 11(c) exhibited a similar variation pattern. When $\chi = 0.2$, the death round reaches the maximum. Only slight fluctuations in death round occur under the situation that χ is within the range of [0.3, 0.7]. Furthermore, the number of death rounds boosts with the increase of χ from 0.7 to 0.9, and the optimal combination is obtained at [0.2, 5°]. From Figure 11(d), it can be noticed that the residual energy after 80% node death gains the maximum when $\chi = 0.2$ and $\beta = 5°$. It can be inferred that the [0.2, 5°] can effectively mitigate the unbalanced energy consumption of nodes. Further, we can summarize that the optimized combination of χ and β has a profound impact on improving the number of death rounds and node residual energy.

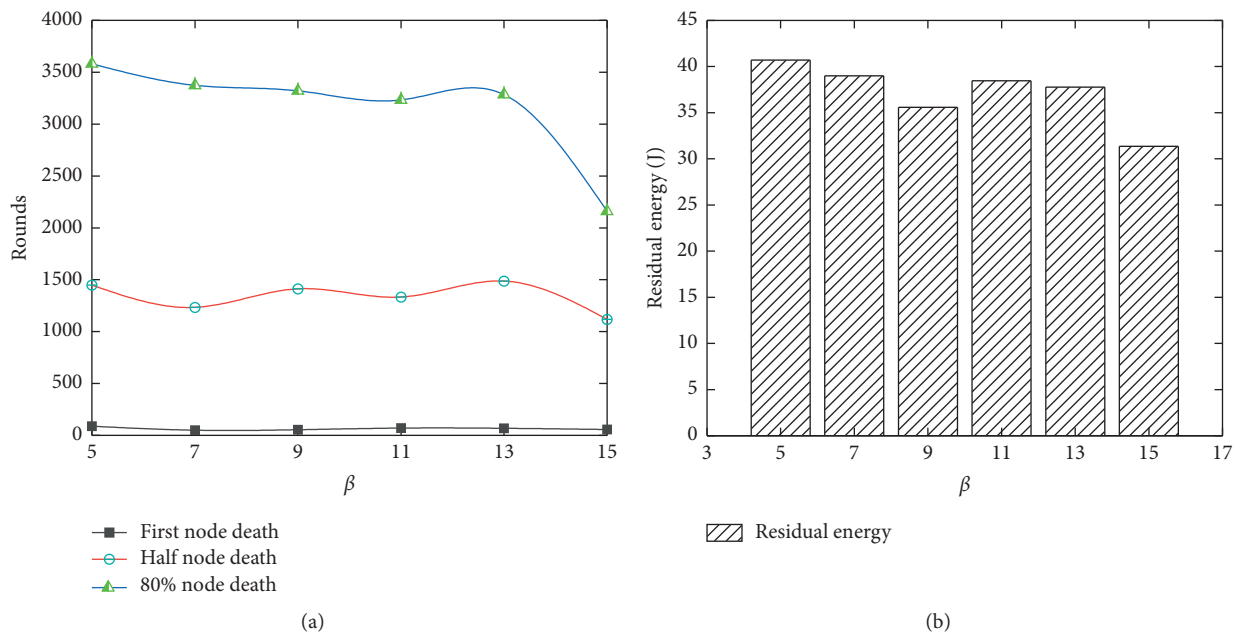


Figure 9: Relevance evaluation between β and assessment metrics: (a) effect of β on death round and (b) effect of β on residual energy.

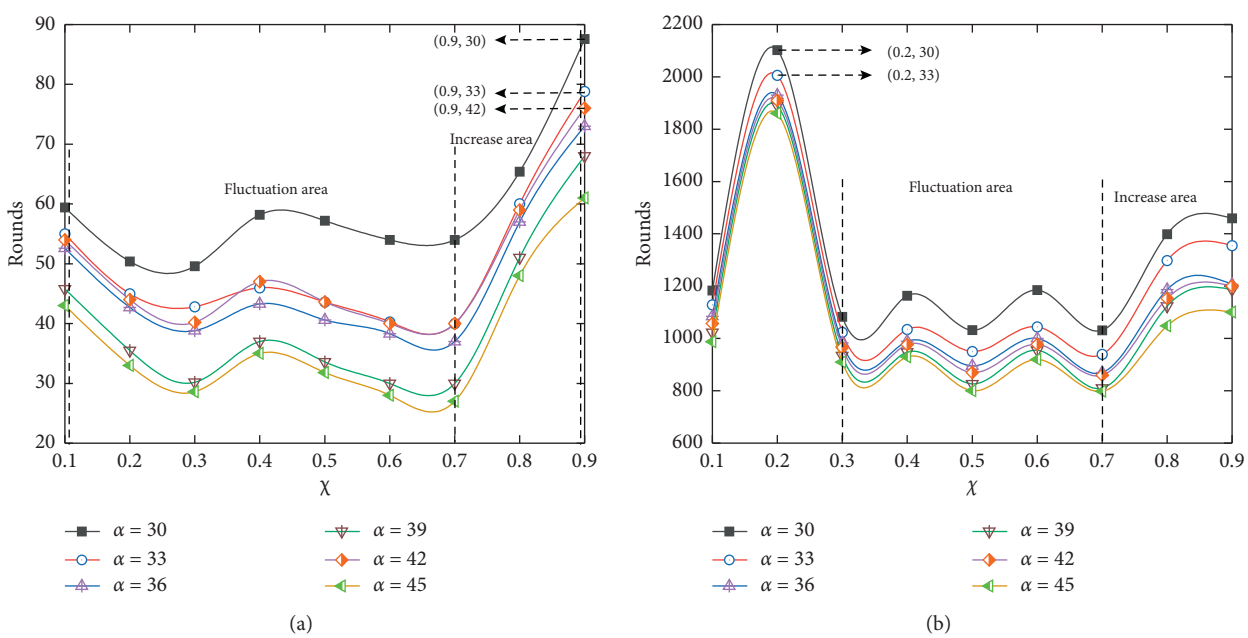


Figure 10: Continued.

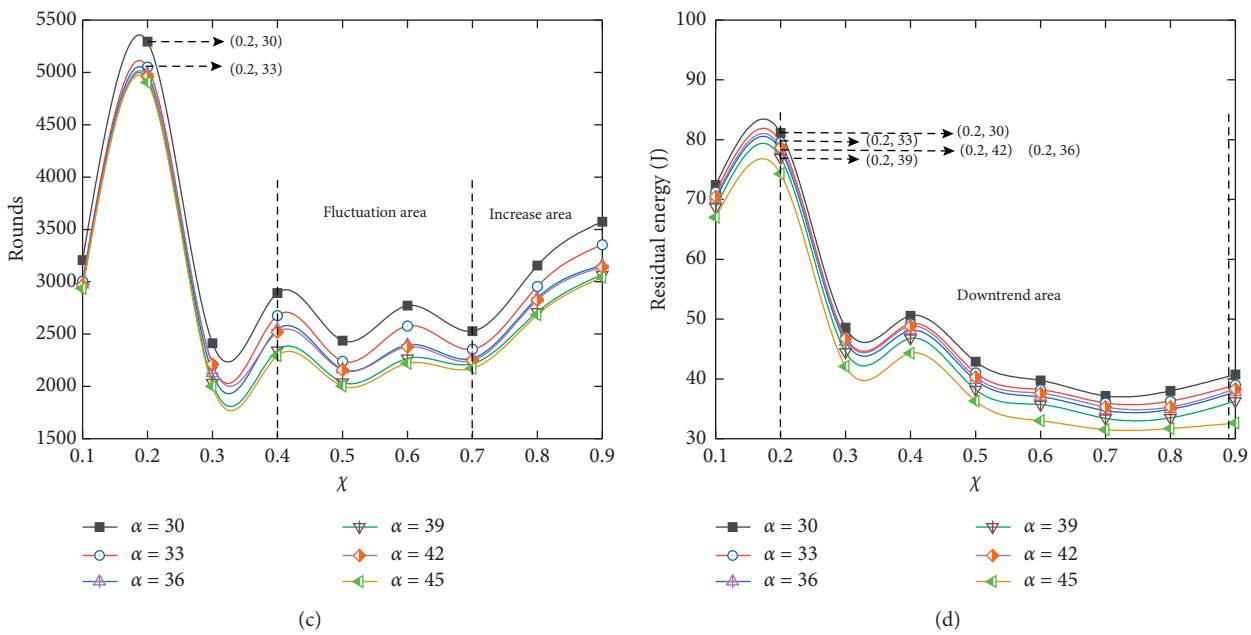


Figure 10: Combination impact of χ and α on assessment metrics, (a) death round of the first node with varying χ and α: (b) death round of half nodes with varying χ and α, (c) death round of 80% nodes with varying χ and α, and (d) residual energy with varying χ and α.

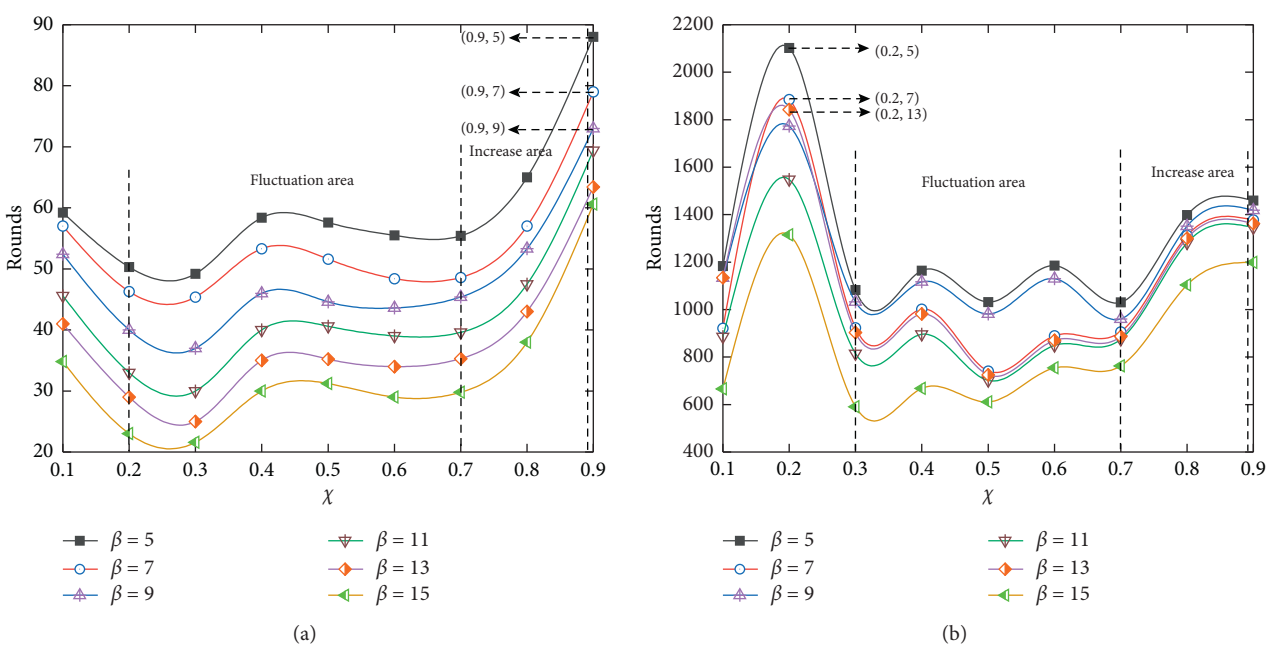


Figure 11: Continued.

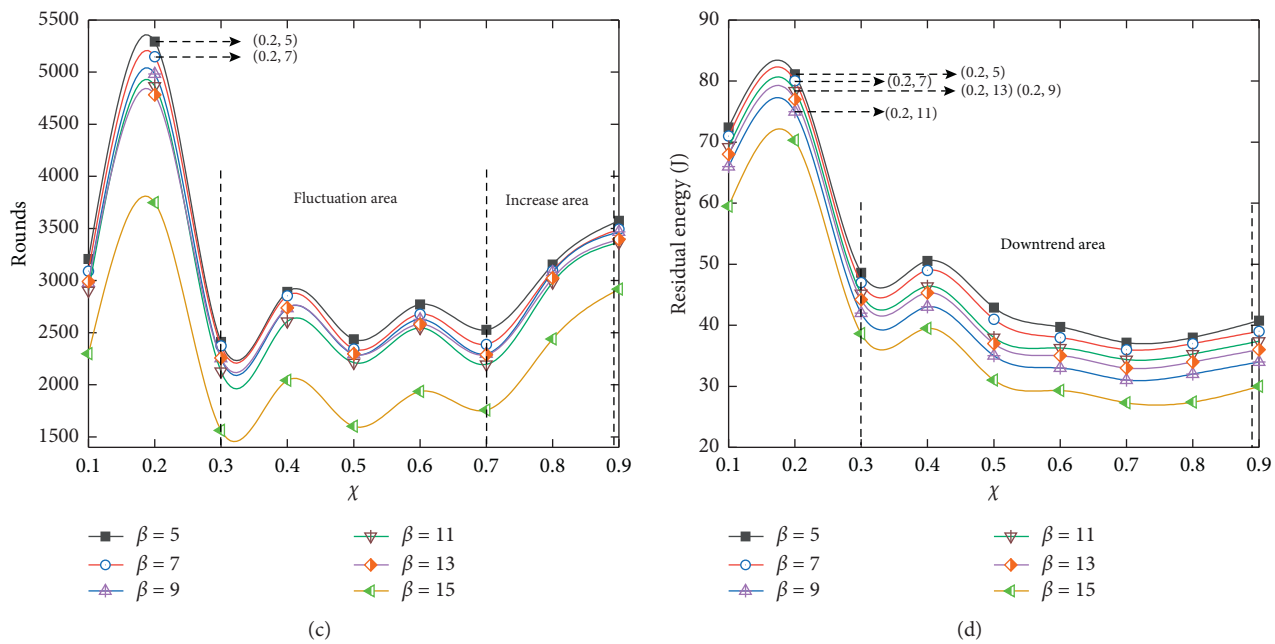


Figure 11: Relationships between [χ, β] and assessment metrics: (a) impact of [χ, β] on the death round of the first node, (b) impact of [χ, β] on the death round of half nodes, (c) impact of [χ, β] on the death round of 80% nodes, and (d) impact of [χ, β] on residual energy.

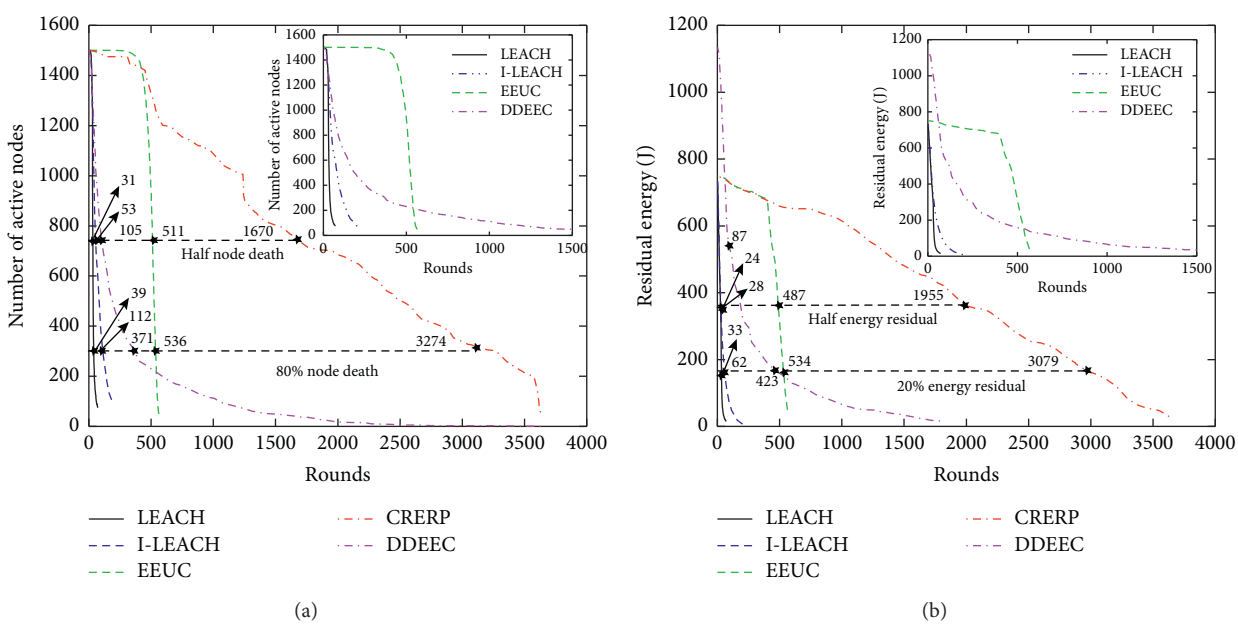


Figure 12: Performance comparisons of CHRERP, LEACH, I-LEACH, EEUC, and DDEEC: (a) number of active nodes with varying simulation rounds and (b) residual energy with varying simulation rounds.

Figure 12 shows the comparison results of CHRERP proposed in this study with LEACH, I-LEACH, EEUC, and DDEEC, by using the number of active nodes and residual energy of nodes as the evaluation indicators. As shown in Figure 12(a), in the case that the number of simulation rounds is less than 400, the number of active nodes of EEUC and CHRERP decreases mildly and keeps above 1400 nodes. However, when the number of simulation rounds continues to increase higher than 400 rounds, the number of active nodes of EEUC declines precipitously, and CHRERP presents a slow decline. Meanwhile, the death round of half nodes for LEACH, I-Leach, EEUC, DDEEC, and CHRERP is 31, 53, 511, 105, and 1670, and the death round of 80% nodes is 39, 112, 536, 371, and 3274, respectively. Compared with the other four protocols, CHRERP can hugely increase the number of active

nodes for the same simulation rounds. It is easily recognized that CHRERP can optimize the clustering specification of the network sensing area and rotate to elect CHs considering the location and residual energy, resulting in reducing the possibility of node being repeatedly selected as the CH and the intracluster communication distance. In addition, the "radial path" with the minimum number of hops is adopted for data transmission between clusters. All the factors mentioned above may contribute to the excellent performance of CHRERP. It can be seen from Figure 12(b) that the number of simulation rounds of LEACH, I-LEACH, EEUC, DDEEC, and CHRERP with half residual energy is 24, 28, 487, 87, and 1955, and the number of simulation rounds with 20% residual energy is 33, 62, 534, 423, and 3079, respectively. Moreover, the initial energy difference originates from the fact that LEACH, I-Leach, EEUC, and CHRERP are all for homogeneous WSNs, while DDEEC is used in heterogeneous WSNs comprising normal nodes and advanced nodes with higher energy. We can conclude that CHRERP has notable advantages in energy saving, due to selecting nodes with higher energy as CHs and establishing the shortest communication path between clusters.

5. Conclusions

In this study, based on the CH rotating election scheme, a novel hierarchical clustering routing protocol was proposed for WSNs to improve the network lifetime. The weighted ratio sum of location and residual energy information of nodes in each CH candidate area was used as the election criterion. We validated the performance of the developed protocol by applying the death round of the first node, the death round of half nodes, the death round of 80% nodes, and the residual energy after 80% node death as assessment metrics. The results indicated that a moderate radius coefficient was confirmed for the balancing effect on both operation rounds and total residual energy. Meanwhile, a fit number of sector districts could effectively prolong the network lifetime by regulating the number of CHs. Little χ, α, and β should be advised for reducing the network energy consumption. In addition, the influence of combined parameters [χ, α] and [χ, β] on the energy-efficiency of the network was consistent with that of an individual parameter. Compared with LEACH, I-LEACH, EEUC, and DDEEC, the addressed protocol exhibited an overwhelming advantage in terms of the number of active nodes and residual energy. Our simulation results suggest that CHRERP is a feasible method for mitigating the unbalanced energy consumption for intracluster and intercluster communication in WSNs. In future work, we will investigate the application of this protocol in a real-world wireless communication scenario to enhance the reliability and practicality of the protocol.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this article.

Authors' Contributions

J.W. conceptualized the study; J.W. and Z.D. developed methodology; Z.H. provided software; Z.D. and X.W. performed validation; J.W. performed formal analysis; J.W. and Z.D. performed investigation; J.W. provided resources; Z.D. and Z.H. performed data curation; J.W., Z.D., and X.W. prepared the original draft; Z.D. and X.W. reviewed and edited the article; Z.D. performed visualization; X.W. supervised the study; J.W. performed project administration; and J.W. was responsible for funding acquisition. All authors have read and agreed to the published version of the manuscript.

Acknowledgments

This research was funded by National Natural Science Foundation of China (Grant no. 61771184), Program for Science & Technology Innovation Talents in Universities of Henan Province (Grant no. 20HASTIT029), Program for Innovative Research Team (in Science and Technology) in University of Henan Province (Grant no. 19IRTSTHN021), and Science and Technology Major Project of Henan Province (Grant no. 181100110100).

References

[1] W. Wen, S. Zhao, C. Shang, and C.-Y. Chang, "EAPC: energy-aware path construction for data collection using mobile sink in wireless sensor networks," *IEEE Sensors Journal*, vol. 18, no. 2, pp. 890–901, 2018.

[2] J. Huang, Y. Hong, Z. Zhao, and Y. Yuan, "An energy-efficient multi-hop routing protocol based on grid clustering for wireless sensor networks," *Cluster Computing*, vol. 20, no. 3, pp. 3071–3083, 2017.

[3] Z. Zhao, K. Xu, G. Hui, and L. Hu, "An energy-efficient clustering routing protocol for wireless sensor networks based on AGNES with balanced energy consumption optimization," *Sensors*, vol. 18, no. 11, Article ID 3938, 2018.

[4] Y. Du, J. Gong, Z. Wang, and N. Xu, "A distributed energy-balanced topology control algorithm based on a noncooperative game for wireless sensor networks," *Sensors*, vol. 18, no. 12, Article ID 4454, 2018.

[5] D. Agrawal and S. Pandey, "FUCA: fuzzy-based unequal clustering algorithm to prolong the lifetime of wireless sensor networks," *International Journal of Communication Systems*, vol. 31, no. 4, Article ID e3448, 2018.

[6] R. Logambigai, S. Ganapathy, and A. Kannan, "Energy-efficient grid-based routing algorithm using intelligent fuzzy rules for wireless sensor networks," *Computers & Electrical Engineering*, vol. 68, pp. 62–75, 2018.

[7] L. Li and D. Li, "An energy-balanced routing protocol for a wireless sensor network," *Journal of Sensors*, vol. 2018, Article ID 8505616, 12 pages, 2018.

[8] B. S. Mostafa and B. A. Massoud, "HEEC: a hybrid unequal energy efficient clustering for wireless sensor networks," *Wireless Networks*, vol. 25, no. 8, pp. 4751–4772, 2018.

[9] H. Rhim, K. Tamine, R. Abassi, S. Damien, and G. Sihem, "A multi-hop graph-based approach for an energy-efficient routing protocol in wireless sensor networks," *Human-centric Computing and Information Sciences*, vol. 8, Article ID 30, 2018.

[10] M. Abbasi and N. Fisal, "Noncooperative game-based energy welfare topology control for wireless sensor networks," *IEEE Sensors Journal*, vol. 15, no. 4, pp. 2344–2355, 2015.

[11] Y. Zhang, M. Liu, and Q. Liu, "An energy-balanced clustering protocol based on an improved CFSFDP algorithm for wireless sensor networks," *Sensors*, vol. 18, no. 3, Article ID 881, 2018.

[12] G. Han and L. Zhang, "WPO-EECRP: energy-efficient clustering routing protocol based on weighting and parameter optimization in WSN," *Wireless Personal Communications*, vol. 98, no. 1, pp. 1171–1205, 2017.

[13] K. Guravaiah and R. Leela Velusamy, "Energy efficient clustering algorithm using RFD based multi-hop communication in wireless sensor networks," *Wireless Personal Communications*, vol. 95, no. 4, pp. 3557–3584, 2017.

[14] A. Emad and M. Ion, "New energy efficient multi-hop routing techniques for wireless sensor networks: static and dynamic techniques," *Sensors*, vol. 18, no. 6, pp. 3557–3584, 2018.

[15] B. Na, G. Han, L. Li, J. Xu, and S. Lei, "An unequal clustering routing protocol for energy-heterogeneous wireless sensor networks," in *Proceedings of the 2015 IEEE/CIC International Conference on Communications in China—Workshops (CIC/ICCC)*, Shenzhen, China, June 2017.

[16] K. Wang, Y. Ou, H. Ji, H. Zhang, and X. Li, "Energy aware hierarchical cluster-based routing protocol for WSNs," *The Journal of China Universities of Posts and Telecommunications*, vol. 23, no. 4, pp. 46–52, 2016.

[17] D. C. Hoang, P. Yadav, R. Kumar, and S. K. Panda, "Real-time implementation of a harmony search algorithm-based clustering protocol for energy-efficient wireless sensor networks," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 774–783, 2013.

[18] A. Mehmood, Z. Lv, J. Lloret, and M. M. Umar, "ELDC: an artificial neural network based energy-efficient and robust routing scheme for pollution monitoring in WSNs," *IEEE Transactions on Emerging Topics in Computing*, vol. 8, no. 1, pp. 106–114, 2017.

[19] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the 33rd Hawaii International Conference on System Sciences*, pp. 1–10, NW Washington, DC, USA, January 2000.

[20] D. Izadi, J. Abawajy, and S. Ghanavati, "An alternative clustering scheme in WSN," *IEEE Sensors Journal*, vol. 15, no. 7, pp. 4148–4155, 2015.

[21] V. Pal, G. Singh, and R. P. Yadav, "Balanced cluster size solution to extend lifetime of wireless sensor networks," *IEEE Internet of Things Journal*, vol. 2, no. 5, pp. 399–401, 2015.

[22] Z. Beiranvand, A. Patooghy, and M. Fazeli, "I-LEACH: an efficient routing algorithm to improve performance & to reduce energy consumption in wireless sensor networks," in *Proceedings of the 5th Conference on Information and Knowledge Technology*, pp. 13–18, Shiraz, Iran, May 2013.

[23] Z. Shen, F. Liu, B. Hou, and C. Zhang, "Energy-efficient uneven clustering routing protocol for wireless sensor networks," *Transducer and Microsystem Technologies*, vol. 32, no. 12, pp. 60–67, 2013.

[24] O. Younis and S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 4, pp. 366–379, 2004.

[25] N. Javaid, T. N. Qureshi, A. H. Khan et al., "EDDEEC: enhanced developed distributed energy-efficient clustering for heterogeneous wireless sensor networks," *Procedia Computer Science*, vol. 19, pp. 914–919, 2013.

[26] A. Rathee, I. Kashyap, and K. Choudhary, "Developed distributed energy-efficient clustering (DDEEC) algorithm based on fuzzy logic approach for optimizing energy management in heterogeneous WSNs," *International Journal of Computer Applications*, vol. 115, no. 17, pp. 14–19, 2015.

[27] W. Mardini, M. B. Yassein, Y. Khamayseh, and B. A. Ghaleb, "Rotated hybrid, energy-efficient and distributed (R-HEED) clustering protocol in WSN," *WSEAS Transactions on Communications*, vol. 10, no. 20, pp. 16416–16420, 2015.

[28] G. Ma and Z. Tao, "A hybrid energy- and time-driven cluster head rotation strategy for distributed wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2013, no. 6, pp. 89–108, 2013.

[29] H. W. Ferng and J. S. Chuang, "Area-partitioned clustering and cluster head rotation for wireless sensor networks," in *Proceedings of the 2017 International Conference on Machine Learning and Cybernetics (ICMLC)*, Ningbo, China, July 2017.

[30] D. Lin, Q. Wang, D. Lin, and Y. Deng, "An energy-efficient clustering routing protocol based on evolutionary game theory in wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2015, pp. 1–12, 2015.

[31] O. A. Amodu and R. A. Raja Mahmood, "Impact of the energy-based and location-based LEACH secondary cluster aggregation on WSN lifetime," *Wireless Networks*, vol. 24, no. 5, pp. 1403–1408, 2013.

[32] S. Lindsey, "PEGASIS: power-efficient gathering in sensor information systems," in *Proceedings of the IEEE Aerospace Conference Proceedings*, pp. 1125–1130, Big Sky, MT, USA, February 2002.

[33] H. K. Farhan, "Enhanced chain-cluster based mixed routing algorithm for wireless sensor networks," *University of Baghdad Engineering Journal*, vol. 22, no. 1, pp. 103–117, 2016.

[34] S. Sasirekha and S. Swamynathan, "Cluster-chain mobile agent routing algorithm for efficient data aggregation in wireless sensor network," *Journal of Communications and Networks*, vol. 19, no. 4, pp. 392–401, 2017.

[35] F. Tang, I. You, S. Guo, M. Guo, and Y. Ma, "A chain-cluster based routing algorithm for wireless sensor networks," *Journal of Intelligent Manufacturing*, vol. 23, no. 4, pp. 1305–1313, 2012.

[36] S. Rani, S. H. Ahmed, J. Malhotra, and R. Talwar, "Energy efficient chain based routing protocol for underwater wireless sensor networks," *Journal of Network and Computer Applications*, vol. 92, pp. 42–50, 2017.

[37] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions on Wireless Communications*, vol. 1, no. 4, pp. 660–670, 2002.