Journal of Network Intelligence
Taiwan Ubiquitous Information
 ISSN 2414-8105 (Online)
Volume **9**, Number **1**, February **2024**

Critical Node Prediction and Risk Monitoring and Early Warning Algorithm for Complex Power Grid Based on Container Cloud Scheduling Technology

Peng Liu*

China Electric Power Research Institute
Beijing 100192, China
724317051@qq.com

Yinghui Xu

China Electric Power Research Institute
Beijing 100192, China
xuyinghui@epri.sgcc.com.cn

Chunyu Deng

China Electric Power Research Institute
Beijing 100192, China
dengchunyu@epri.sgcc.com.cn

*Corresponding author: Peng Liu

Received April 23, 2023, revised June 28, 2023, accepted August 27, 2023.

Abstract. *As the power system increasingly grows in scale, recent year witness frequent blackouts at home and abroad, which inflicts serious impact on people's life and social economy. In this study, the power grid results are modeled based on complex network model, and the power grid environment was scheduled and optimized by multi-objective container cloud scheduling technology. According to the K-Shell framework, the prediction algorithm was proposed for the prediction of key panel points in the electrified netting. Then, the complex power grid was monitored and early warned based on the optimal early warning parameters to optimize distributed power grid in a specific scenario of complex large-scale power grid, thus providing references for meeting cross-level and cross-center service coordination requirements in power grid business. Through the results of the experiment, it is easy to find that the model algorithm come up in this research is able to predict the key panel points of the power grid effectively and implement effective risk monitoring and early warning.*

Keywords: Container technology; Complex power grid; Key node prediction; Risk monitoring and early warning

1. **Introduction.** In this period of time, benefiting from the overall improvement in manufacturing industry, the electrified netting industry, as an engine of manufacturing industry, has created remarkable results [1]. In the domestic environment of gradually open power market, there are more prominent uncertain factors affecting programming and scheduling of power network, including the randomness of power supply, the fluctuation of load and price, the failure of power network equipment, the changeable cost, etc. The randomness of power sources refers to the randomness of the power sources that provide power support in a distributed power grid. Load fluctuation refers to the fluctuation

in the impedance of components connected to both ends of the power supply in the power grid and the power generated, converted, and consumed by electric power equipment during operation. Price fluctuations refer to the fluctuations in the fees required to use power grid resources. The failure of power grid equipment refers to the physical failure caused by some force majeure factors during the operation of the power grid, which prevents the normal supply of power grid resources in a short period of time. The above variable costs refer to the cost price changes caused by some alternative energy substances when using the power grid. These uncertainties result in random power market operation and power grid operation mode, and also make it difficult for scheduling personnel to accurately quantify the real system risks, which increases pressure for correct operation. In the meantime, the system function loss due to the failure of the grid system components greatly impairs the reliability of the power supply system and incurs immeasurable losses to the society. In the power market environment, it's of great necessity to realize static safety check of the power system, select high-risk failure scenarios as the expected fault set, search for weak system links, formulate prevention and control measures before the failure, which is an effective way to avoid the system operation risk [2]. Grid stability can be subdivided into frequency stability, voltage stability and power angle stability [3]. Frequency stability reflects the overall energy supply and demand balance of the power grid [4]. Power grid faces disturbances such as load fluctuation, line disconnection from time to time. These disturbances may lead to frequency collapse of the power grid, which in turn causes cascading failures or blackouts, etc., or even induces more severe consequences and huge losses [5]. Recent research by domestic and foreign academic researchers reveals that blackouts closely correlate with some key panel points in the electrified netting, while these key nodes are often quite significant to the safety and stability of the operation of the electrical power systems. If these critical panel points fail, it can easily trigger chain failure of the power grid, then leading to instability and collapse of the power grid [6, 7]. Hence, finding a way to identify these key nodes accurately and quickly in line with the actual condition is crucial for preventing large-scale chain failure. If these key nodes in the electrical power system are identified accurately and quickly, it can let us improve the pertinence in protection control strategy, strengthen safety and stability of the electrical power system, and take precautions against blackouts caused by cascading failures of the power system [8].

At present, steady progress is made in the containerization transformation of the existing business and equipment in the power industry. Issues such as different system functions, high complexity, and multiple coexisting systems of multiple vendors become major obstacles to the containerized migration of power grids [9, 10]. Containerization of application system can be carried out mainly from the aspects of system development, system deployment, operation and maintenance management, thereby achieving joint system cloud management [11]. In the current technical means, micro-service architecture is employed for system design and development, container cluster is adopted for micro-service deployment, and container cloud operation and maintenance management system is used for unified management of development, deployment and container cluster of all systems, thereby forming an organic container-based application cloud platform. Container technology represents a virtual server resource sharing mode for building operating system application instances on demand [12]. Docker is a typical application of container technology. In many projects developed around Docker, such as container operating system, development platform, development tools, big data, system monitoring, etc., Docker application program interface is mostly used for container management. The container management system of power grid scheduling cloud application provides containerized operation management of power grid scheduling, covering functions such as deployment and

configuration of container services, definition, configuration, startup, initialization, status management, service allocation of container service process [13]. With regard to control service management module, service unit monitoring module and configuration operation management module, power grid service unit monitoring module supports container cluster monitoring, container monitoring, container application monitoring and application process monitoring. Configuration power grid operation management module includes system configuration tool and management process configuration [14, 15].

So far, some research results have been made in the discrimination of critical panel points in the electrified netting. Some scholars suggested using the weighted average of node degree and node power as importance indexes of nodes [16]. Some scholars utilized the decline of network transmission efficiency before and after node failure to characterize node fragility [17]. Some scholars defined the weight of a node as the sum of the ratio between the branch power flow of all the edges connected to the node and the branch capacity limit. On this basis, a fragility assessment method was introduced for nodes [18]. Some scholars used the clustering degree of network nodes after shrinkage to judge node importance. The node to be solved in the network was integrated with its adjacent nodes as a new node, and a greater number of branches after the merger suggest greater importance of the original node [19]. In addition, based on quasi-steady power transfer distribution factor, some scholars put forward an improved nodal degree evaluation model combining power system tidal flow information to identify key nodes bearing an important role in power transmission [20]. From network structure, scholars found that power grid can maintain stability under most disturbances, but when the key power station node is attacked, the power grid will experience great decline in synchronization ability [21]. Some scholars carried out modeling research on cascaded failure of North American power grid. Combining the actual topology of the electrified netting and thinking about the load and overload of transmission substations, researchers found that failure in a small number of substations would trigger network cascading failures [22, 23]. Existing literatures mainly give considerations to the electrical distance of nodes, coupling connection degree, physical meaning of electrical system, and network characteristics, as a way to find key nodes of power grid more efficiently [24, 25]. In order to reflect the local and global importance of nodes, some scholars modified the node importance matrix to more appropriately rank node importance [26, 27], and proposed the use of the number of grid lines as an index for identifying fragile lines [28, 29]. Based on the topology structure and system operation status, the literature further draws on the power transmission relationship of the power grid to make up for the shortcomings of the shortest path transmission in the hypothesis of the existing power grid optimization model [30, 31].

The cloud resource scheduling problem mainly focuses on how to achieve reasonable allocation of resources such as storage, computing, and network in the cloud computing environment, and intelligently predict and manage the usage of resources, making the scheduling process fully reasonable and effortless. The goal of this article is to construct a high-performance container cloud scheduling technology based algorithm for predicting and monitoring key nodes in complex power grids in the context of cloud resource scheduling problems. In this research, one kind of cloud resource scheduling model, a multi-objective container cloud scheduling model and a complex power grid model based on community environment were constructed to search for the optimal future state and improve the resource utilization efficiency, thereby providing a feasible operation scheme for the scheduling and allocation of complex power grid resources. Then, this paper described the classical key node prediction methods for power grid nodes. A critical panel point prediction algorithm was raised based on improved hybrid K-Shell method and optimized based on container cloud scheduling. Simulation and comparative empirical study

were performed on the improved hybrid K-Shell algorithm proposed herein and related link prediction algorithms, which verified the superiority of the algorithm which was put forward. This research took the topology structure of complex power grid as the research object, built a resource scheduling model based on container cloud scheduling technology, and introduced an efficient improved link prediction algorithm to provide theoretical basis and practical guarantee for the key node prediction, risk monitoring and early warning in complex power grid.

This article innovatively achieves many optimization and breakthroughs in models and algorithms. This article innovatively combines the multi-objective cloud container scheduling model with the complex power grid model to achieve intelligent scheduling of power grid resources in complex community structures. Moreover, the innovative algorithm proposed in this article not only achieves high-quality node clustering, but also improves the hybrid algorithm based on the neighborhood level of nodes, cloud container scheduling technology, and optimal warning parameters in real grids, proposing some efficient and stable innovative algorithms.

2. **Model construction.**

2.1. **Cloud resource scheduling model.** During the scheduling of power grid cloud resources, different cloud resources will be allocated to different users in light of user needs [32, 33]. First, the expression of cloud resource scheduling task should be calculated to build the scheduling time matrix. Suppose the set of cloud resources involved in scheduling tasks is J, P represents the set of power grid cloud resources to be allocated. Then, the scheduling task of power grid cloud resources can be described as $J = \{J_1, J_2, \ldots, J_n\}$, $P = \{P_1, P_2, \ldots, P_n\}$. J_i shows the i th scheduling target, and P_j shows the power grid cloud resources to be allocated in the j-th scheduling task. The scheduling time matrix is solved for grid cloud resource. Suppose $E_{TC,ij}$ represents the time consumed by grid cloud resource scheduling task J_i in allocation of grid cloud resource P_j, and the grid cloud resource scheduling task J_i requires shortest time $E_{TC,ij}$ when allocating cloud resource P_j. Then, the scheduling time matrix of grid cloud resource can be calculated, as shown in Equation (1).

$$E_{TC/CT,n*m} = \begin{Bmatrix} E_{TC,11} & \cdots & E_{TC,1m} & E_{CT,11} & \cdots & E_{CT,1m} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ E_{TC,n1} & \cdots & E_{TC,nm} & E_{CT,n1} & \cdots & E_{CT,nm} \end{Bmatrix} \tag{1}$$

In general, scheduling of grid cloud resources is to use the shortest time for scheduling. x_{ij} is used to describe whether cloud resource scheduling task can independently allocate grid cloud resource P_j. The objective function which belongs to the grid cloud resource scheduling tasks can be seen in Equation (2).

$$S(J) = \min \left(\sum_{i=1}^{n} \sum_{j=1}^{m} E_{TC,ij} * x_{ij} \right), x_{ij} = \{0, 1\} \tag{2}$$

To further improve the power grid resource allocation effect, this paper adopted a dual-layer programming-based dynamic allocation algorithm for power grid cloud resources [34]. The multi-objective optimization method is often to calculate the weighted sum of each objective, implement single objective optimization technique, and then optimize the objective. When this method is used for optimization, comparison is difficult due to the inconsistent units between power network objectives of different nature. Moreover, optimization objective is only the weighting of each objective and the operation progress of each objective during the optimization [35, 36]. A, B respectively represent the upper

and lower programming; D and d represent the objective function of upper and lower programming respectively. S and s respectively represent constraint conditions. Both x and y represent decision variables. The dual-layer programming model is introduced into the dynamic allocation of power grid cloud resources, and the dual-layer programming theory can be put into use to analyze this dynamic best allocation about power grid cloud resources [37]. During optimization of dynamic allocation results of power grid cloud resources, the objective function can be expressed as in Equation (3).

$$\max\left\{\varphi\left(R_k, V_j\right)\right\} = \left(\frac{\sum_{j=1}^{m} v_j^C}{\sum_{k=1}^{n} R_k^C} + \frac{\sum_{j=1}^{m} v_j^M}{\sum_{k=1}^{n} R_k^M} + \frac{\sum_{j=1}^{m} v_j^B}{\sum_{k=1}^{n} R_k^B}\right) \Big/ 3 \tag{3}$$

The above objective function is solved to construct the cloud resource scheduling function $F(\cdot)$ in the dual-layer programming mode, as shown in Equation (4).

$$F(\cdot) = \frac{\sum_{i=1}^{n} x_{ij} * \min\left(\sum_{i=1}^{n}\sum_{j=1}^{m} E_{TC,ij} * x_{ij}\right) * E_{CT,n\times m}}{P_{CT,n\times m} * \sum_{j=1}^{m} x_{ij}} \tag{4}$$

2.2. **Multi-objective container cloud scheduling model.** The multi-objective container cloud scheduling model is based on the assumption of rational resource allocation requests, by controlling the central cloud to control edge cloud clusters, edge cloud nodes, etc., to achieve the goal of reasonable allocation of cloud resources. By inputting parameters that express the amount and status of resources, a multi cluster load balancing allocation scheme is obtained, which reduces the scheduling delay of cloud resources and improves the average resource utilization rate. Owing to environmental changes, the present distribution scheme of the grid container cloud changes into possibly non-optimal present state. This paper aims to find the optimal future state on this basis [38]. This model is oriented towards multi-node and multi-platform application frameworks, with tasks between application frameworks independent of each other, so are tasks within the application frameworks [39, 40]. Suppose the grid data center can supply M physical resources, which is expressed as the matrix $a_m^{\text{prop}}, m \in [1, \ldots, M]$. For instance, a_1^{CPU} indicates the CPU rate affordable for physical machine 1. Each power grid resource shares the same instruction set, but the same instruction has different execution rates, showing heterogeneity in memory size, network bandwidth, etc. [41]. If the power grid resources are not assigned with tasks, it is in the off state; otherwise, it is in the on state. The upper application framework submits N container tasks, with each represented as the matrix $c_n^{\text{prop}}, n \in [1, N]$, prop $\in$ PROP. For example, c_1^{CPU} indicates the CPU resources required by task 1, and c_1^{JOOB} indicates the job number of task 1 . The mapping scheme between resources and tasks is indicated as o-1 matrix of $N * M$, which is expressed as$_{m,n}^{p}, m \in [1, \ldots, M]$, $n \in [1, \ldots, N]$.

The new mapping scheme $o_{m,n}^{NEW}$ is to establish new mapping from the randomly generated candidate schemes without considering the old one. The allocation of each attribute resource in the power grid can be calculated using Equation (5).

$$s_{m,1}^{\text{prop}} = \sum_{n=1}^{N}\left(c_n^{prop} o_{m,n}^{PRESENT} + c_n^{prop} o_{m,1}^{NEW}\right) \tag{5}$$

Where, c_n^{prop} indicates N container tasks submitted by the upper-layer application framework. When attempt is made to allocate resources satisfying the task requirements of n grid containers on the power grid, the allocation scheme concerns the allocation scheme of the previous $n-1$ container tasks and resource demands of the previous $n-1$

container tasks, specifically as shown in Equation (6).

$$s_{m,n}^{\text{prop}} = s_{m,n-1}^{\text{prop}} + c_{m,n-1}^{\text{prop}} o_{m,n-1}^{NEW} \tag{6}$$

Where, c_n^{prop} indicates N container tasks submitted by the upper-layer application framework. Resource requests for all container tasks must meet the power grid cloud resource constraints, $s_{m,N}^{\text{prop}} \leq a_m^{\text{prop}}$. For the grid application framework, the fairest distribution mode is to allocate required amount of resources to each upper framework. When resource supply is between the two, the fairest distribution method is to let some frameworks with less demand receive the required amount of resources, with the remaining resources equally shared in other frameworks [42]. Based on this principle, the fairness function $FA(\cdot)$ of resource and task mapping scheme is determined, as shown in Equation (7), while the average resource value u_j^{AVG} can be defined as in Equation (8).

$$FA(\cdot) = \frac{\sum_{j=1}^{J} \max\left\{1, \left(\frac{jv_j}{u_j^{IEDAL-1}}\right)^2\right\} - \sum_{m=1}^{M}\sum_{n=1}^{N} o_{m,n}^{FXED} jc_{j,n} cv_n - 1/jv_j}{\left(\frac{\min\{Cv_n | n\in[1,\ N]\}}{\max\{jv_j | j\in[1,\ J]\}}\right)^2} \tag{7}$$

$$u_j^{AVG} = \begin{cases} u_{j-1}^{AVG} + \frac{u_{j-1}^{AVG} - jv_{j-1}}{J-(j-1)}, & jv_{j-1} < u_{j-1}^{AVG} \\ u_{j-1}^{AVG}, & jv_{j-1} \geq u_{j-1}^{AVG} \end{cases} \tag{8}$$

On the premise of meeting equal-value demands, supply of power grid resource should be minimized to release excess resources, reduce resource waste and lower operating costs, thereby increasing revenue [43, 44]. The value of releasable physical machine resources without task load is used to indicate redundancy $\sum_{m=1}^{M} mv_m W.W \in \{0, 1\}$ means the grid node is on. However, after releasing the load-free grid cloud resources, the amount of grid resources supplied by the container cloud should be not less than the scale of resources which is in need for the task [45, 46]. The difference between supply and demand is spread unevenly across multiple grid nodes. Balanced load distribution favors the system stability and reliability. To this end, the power network load uniformity $B(\cdot)$ can be defined, as shown in Equation (9). Where, $\sum_{n=1}^{N} \frac{o_{m,n}^{FLXD_c} cv_n}{mv_m}$ indicates the ratio between the actual amount of resources affordable for each physical machine and the amount of supplied resources.

$$B(\cdot) = 1 - \sum_{m=1}^{M} \left(\frac{\sum_{n=1}^{N} {}^{F,n,nFIXD_{cvn}}}{mv_m} - \frac{\text{satisfication}}{\sum_{m=1}^{M} mv_m\text{-remain}} \right)^2 * \frac{W}{M} \tag{9}$$

2.3. **Construction of complex power grid model.** Complex power grid can be expressed as $G(V, E)$, in which, $V = \{v_i \mid i = 0, \ldots, n\}, E = \{e_{ij} = e_{ji} = \{v_i, v_j\} \mid v_i, v_j \in V\}$.$N$ can be the quantity of panel points in $|V| = N$, edge e_{ij} means connections between v_i, v_j. In most literatures, the adjacency matrix $A = [a_{ij}]_{V\times V} \cdot a_{ij} = 1$ is used to indicate presence of edge between nodes v_i, v_j. Otherwise, $a_{ij} = 0$. The community structure in the grid usually means that the nodes can be divided into subsets $C = \{C_1, C_2, \ldots, C_k\}$, so that the panel points C_j have a close connection when facing the same subset, while the links between various subsets can be relatively sparse. Existing studies mainly focus on disjoint power grid structures with each node belonging to only one community [47]. Nevertheless, power grids often contain overlapping community structures where nodes may have multiple community members. Suppose $C = \{C_1, C_2, \ldots, C_k\}$ is the community structure of grid $G(V, E)$. For any two communities C_i and C_j $(C_i \cap C_j = \phi)$ in C, C is a disjoint group if $i \neq j$. Otherwise, C is referred to as overlapping cluster.

In this study, V is viewed as a set of randomly generated labels, which has nothing to do with the original features of grid nodes. Also, it is hypothesized that the set V can be arbitrary traversal, and ordered subset of node is expressed as $S \subseteq V = \{v_1, v_2, \ldots, v_t\}$. Where, $v_i \in S$ means $i \in \{1, \ldots, t\}$. Ordered set of damage nodes in a given graph G is expressed as $= \{s_1, \ldots, s_t\}$. Literature [48] proposed the use of distance from damage node to simulate opponent vector $r_G(u \mid S)$ with regard to node $u \in V$, as shown in Equation (10).

$$r_G(u \mid S) = (d_G(u, s_1), \ldots, (d_G(u, s_t)) \tag{10}$$

Where, the distance $d_G(u, v)$ can be calculated by the quantity of edges which belong to the path with the least distance connecting u and v. This vector is known as the metric representation of u about S. It can be hypothesized that the system can re-identify these nodes, and these nodes have a unique metric identifier relative to the set of damage nodes under their control [49]. Thus, the grid manager must ensure that each node in the released social network graph is at least indistinguishable from minimum number of other nodes. This property can be defined by the concept of k inverse solution set as follows. Suppose there is a maximum positive integer k in the graph $G = (V, E)$ that makes each $v \in V(G) \backslash S$ have nodes $\omega_1, \omega_2, \ldots, \omega_{k-1} \in V(G) \backslash S$, and $\omega_1, \omega_2, \ldots, \omega_{k-1}$ are mutually different and meet the constraints shown in Equation (11).

$$r_G(v \mid S) = (r_G(\omega_1 \mid S) = \cdots = r_G(\omega_{k-1} \mid S) \tag{11}$$

If the damage node set S is compelled to be the inverse solution of k, then for every victim node v, at least $k - 1$ other nodes cannot be distinguished from v by merely examining the metric representation about S.

3. Algorithm design.

3.1. Classical key node prediction method.

With the surging number of grid nodes and the continuous optimization of the grid risk control technology in the industry, the node-set identification of grid node influence has turned into a significant problem of academic attention [50]. The existing methods simply fall into two categories. In centrality measurement, scalability of one panel point depends on the location of this meshwork. The panel point with higher degree figure in degree centrality [51] is more influential. In closeness centrality [52], nodes with smaller total distances from other nodes have greater propagation capability. In intermediate centrality [53], the first is to determine the path with the least distance between the two panel points in the meshwork. The quantity of paths where panel point v_i is located depends on the centrality of panel point v_i. In k-shell centrality [54], the closeness between node v_i and graphics core depends on its propagation level. This measure depends on a k-shell figure to each panel point v_i, so those panel points near graph core have higher k-shell figure. k-shell iteration factor centrality [55] is an improved k-shell centrality, which tries to divide panel points with various propagation capability figures into one single shell according to the iteration process in each k-shell stage. In clustering arrangement [56], panel points with smaller relevance between adjacent nodes are deemed as panel points which have influence. The centrality value of node v_i can be calculated as shown in Equation (12).

$$CR_i = f(cc_i) \sum_{v_j \in N_i} \left(d_j^{\text{out}} + 1\right) \tag{12}$$

Where, d_j^{out} can be the out-degree of panel point v_j, cc_i can be the clustering coefficient of panel point v_i, N_i represents the neighborhood set of node v_i, with function f being defined as 10^{-cc_i}. In hierarchical k-shell centrality [57], one hierarchical standard can be first put forward to describe the topological position of panel points relative to the graph

core in an increasingly accurate way , and the impact degree of each panel point can be calculated according to the location of adjacent panel points. In entropy centrality [58], the concept of entropy can be described to test the neighbor effection distribution of panel points. For entropy centrality, nodes with greater neighbor influence are viewed as influence nodes.

The seed set selection method is to search for a network node subset to solve the node-set identification problem of power grid influence, which can be roughly divided into two subcategories: (1) Greedy algorithm [59], wherein the propagation model is used to determine the optimal subset, which is invalid from the perspective of time if the network scale keeps expanding; (2) Method of using the centrality measure and thinking about the distance from one panel point to another [60]. It searchers for the best seed set to ensure all the selected set parts have minimal overlap. Providing that panel pointing v_i is joined into the seed set as an effective panel point, the centrality of its adjacent nodes will be lowered by 1 , so that the seed set is selected within an appropriate distance. The process continues its iterations until the first k panel points can be selected. Degree discount [61] is similar to single discount, but different from the neighbor centrality reduction of node v_i, as shown in Equation (13).

$$dd_i = d_j - 2t_j - (d_j - t_j)\, t_j p \tag{13}$$

Where, d_j can be the degree of panel point v_j, t_j ca the part of neighboring nodes of panel point v_j which can be selected as a seed set member, and p can be the probability of information propagation from panel point v_i to v_j. In degree distance [28], the panel point which owns the highest order can be taken as the candidate panel point of the seed set member in each stage. Providing that the distance from the candidate panel point to the present seed set member exceeds the threshold, then, the panel point can be joined into the set which is composed with seed. Then, Initial Multi-Spreader Nodes selection (IMSN) [62] firstly ranks nodes using degree values, shells, and hybrid measure of neighbor difference in various shells. The panel point with the biggest degree can be selected in each stage. Providing that the coincidence from the candidate panel point to the present seed set is a bit little, the panel point can be joined into the seed set as the part which is new. In degree penalty [63], firstly, panel points can be arranged in order by degree. In each period, a panel point is selected from the list of netting panel points and joined into the subset v_l. Providing that a panel point can be joined into the seed set as the part which is new, then panel point v_j can be located nearby, while node v_l is located near v_j. According to the penalty function shown in Equation (14), its centrality value decreases respectively. The sorted list is updated with changes in the neighbor centrality value of node v_i. This process keeps until the members with the quantity of k are selected.

$$\{c_j(t+1), c_l(t+1)\} = \{c_j(t) - d_i \cdot w, c_l(t) - d_j.w^2\} \tag{14}$$

Where, $c_j(t)$ shows the centrality of panel point v_j at the period t, and w shows the factor which is used to implement penalties. In the coloring means which depend on the distance [63], panel points can be sorted for the first according to the centrality measurement method. Each panel point can be then assigned with a ranking based index. Hence, the distance from one panel point to another whose same index exceeds the threshold. Panel points which have the same index can get simultaneous sorting. Each group of panel points can be then ranked in order depending on the centrality. Additionally, the k members which own the highest ranking in the group can be regarded as the seed set members. In heuristic clustering [64], the similarity measure between the two nodes v_i and v_j is first introduced to perform clustering on nodes according to the similarity degree. Firstly, k panel points can be selected in a random way which comes

from the meshwork as clustering centers, and other panel points can be divided into various clusters according to the similarity between nodes and clustering centers. The power grid node with the highest similarity which belongs to each cluster can be deemed as a clustering center which is new, while other panel points can be reassigned to the cluster according to the similarity of cluster central nodes until the algorithm realizes its convergence. Finally, the cluster center can be described as a seed set member, and the similarity about the two panel points v_i and v_j can be calculated as shown in Equation (15).

$$\mathrm{Sim}_{ij} = \lambda * A^{(2)} * A^{(3)} \tag{15}$$

Where, $A^{(r)}$ shows the quantity which belongs to the paths with the length of r about the double panel points v_i and v_j, and λ can be an adjustable parameter which is within the interval of $(0, 1)$.

3.2. **Key node prediction algorithm based on improved hybrid K-Shell method.** It is easy to know that the classical hybrid K-Shell means can be used to reduce monotonicity. Node degree, node distance measurement and node K-Shell index are used to calculate the KSH (K-shell-hybrid) figure of all panel points in a complex meshwork, with panel points ranked in order depending on this, specifically as shown in Equation (16).

$$k\,\mathrm{sh}\,(v_i) = \sum_{v_j \in \phi(v_i)} \frac{\alpha\,(v_i, v_j) + \mu * k\,(v_j)}{d^2\,(v_i, v_j)} \tag{16}$$

Where, $\phi\,(v_i)$ is the neighborhood r-magnitude of node v_i. $\alpha(v_i, v_j)$ is known as the K-Shell power and can get the definition of $\sqrt{(ks(v_i) + ks(v_j))} \cdot ks(v_i)$ is the K-Shell exponent of panel point v_i. $ks\,(v_j)$ can describe the K-Shell index of adjacent panel point v_j. $k(v_j)$ represents the degree of panel point v_j, and $d(v_i, v_j)$ is the distance which is shortest between panel points v_i and v_j.μ can be a free parameter which owns a figure within the interval from o to 1 . Moreover, this method can be extended to incorporate the ksh value of adjacent nodes, thus yielding the extended K-Shell mixed value (ksh_+). In this way, (ksh_+)can show more positive information than (ksh), and its improved expression is shown in Equation (17).

$$k\,\mathrm{sh}_+\,(v_i) = \sum_{v_j \in \psi(v_i)} k\,\mathrm{sh}\,(v_j) \tag{17}$$

The improvement of the hybrid K-Shell method in the complex grid scenario requires the condition that node degree influence of the neighborhood decreases with the decrease of the neighborhood level (l). That is, nearby neighbor has greater contribution than the distant neighbor. In the hybrid K-Shell method, $0 < \mu < 1$ is used to reduce the node-degree influence of panel points which are neighboring on the importance of the selected panel point. The paper introduces a mathematical expression based on the network parameter μ. The network parameter r gives consideration to the maximum neighborhood level. For instance, the network diameter is d and $r \leq d$. In studies with different r values, Namtirtha et al. [41] observed that for most practical networks, node-ranking performance will not be further improved if $r > 3$. This study also verified this, with $r = 3$ set in all experiments. Hence, a function expressing μ as l is shown in Equation (18).

$$\mu(l) = \frac{2 * (r - l + 1)}{r * (r + 1)} \tag{18}$$

Where, $\sum_{l=1}^{r} \mu(l) = 1, l$ represents neighborhood level, r is the maximum level of the considered neighborhood. As a matter of fact, the nearest $(l = 1)$ neighborhood has

the highest μ value. When $l = r$, that is, if the farthest neighbor is considered, the μ value becomes smaller. Moreover, according to the hypothesis, the outermost core node ($ks = 1$) has lower importance than the inner core node. By multiplying the K-Shell exponents between two nodes and then dividing them, it can more effectively demonstrate the mutual influence between the two nodes. so the improved K-Shell can be defined as shown in Equation (19).

$$\alpha_I(v_i, v_j) = \sqrt{(ks(v_i) * ks(v_j))} \tag{19}$$

When both nodes in the above function derive from the inner core, its importance will be greater, but if any one or two nodes belong to the outermost layer, its importance will be smaller. Obviously, the original K-Shell power function is increasingly important for panel points with $ks = 1$, but the improved K-Shell power function has bigger weight on the inner shell panel points. In particular, when the complex network has incomplete global structure, the reliable local proxy shows the sum of the nearest neighbors when facing the measurement impact of panel points [65]. Based on this, this study incorporated node degree k in the equation, and used parametrized μ as a basic discount for geodesic. So, in the proposed improved hybrid K-Shell, the function $IS(v_i)$ of node v_i with respect to parameters α and μ can be expressed as shown in Equation (20).

$$IS(v_i) = \sum_{l=1}^{r} \sum_{v_j \in \phi(v_i, l)} \frac{\alpha_l(v_i, v_j) + \mu(l) * k(v_j)}{d^2(v_i, v_j)} \tag{20}$$

Where, v_j is magnitude l of the neighborhood ($\phi(v_i, l)$) with respect to node v_i. Magnitude l rises to the maximum level r under consideration. Algorithm 1 reports the internal mechanism of the improved hybrid K-Shell means which is proposed herein.

Algorithm 1 Improved hybrid K-Shell algorithm in the power grid scenario

Input: $G(V, E)$, k, ks, r, m
Output: $Rank[v_i, IS(v_i)]$

```
The list Rank contains the nodes with corresponding IS(v_i) value assigned
for v_i ∈ V do
    for level l = 1 : r do
        Compute μ(l)
        for v_j ∈ (v_i, l) do
            Compute formula (19)
            Compute the shortest distance between node v_i and v_j as d(v_i, v_j) = l
            IS(v_i, l)_+ = (α_I(v_i, v_j) + μ(l) * k(v_j)) / d^2(v_i, v_j)
        end for
    end for
    Update the node entry in Rank [v_i, IS(v_i)]
end for
```

3.3. **Algorithm optimization based on container cloud scheduling.** Grid container cloud scheduling is a new algorithm system for fitting and simulation of complex grid system. Grid container is compatible with uniform distribution. The set gets the definition of $Z_{l,b}(t) = \{c_{l,b}(t), v_{l,b}(t)\}$. According to the construction level, power grid containers can be divided into two categories: primary level (with no subdivision downwards) and parent level, as shown in Equation (21).

$$\{Z_{0,b}(t); Z_{l,u}(t)\} = \{c_{0,b}(t), v_{0,b}(t); Z_{l-1,q}(t), q \in Q\} \tag{21}$$

Where, Q is the number set of the grid subcontainers, in other words, the parent container is a set composed of containers with a smaller level. Concerning about the time parameter t, the process which is used to realize an input/output can get the description of an evolutionary process: overflow occurs when the level of content added to the grid container exceeds the capacity which belongs to the container itself. Hence, the input process about the original container, and the input process of the parent container can get the definition in Equation (22) and Equation (23), respectively.

$$o_{0,b}(t) = \begin{cases} 0 & c_{0,b}(t) \leq v_{0,b}(t) \\ c_{0,b}(t) - \alpha_{0,b}(t) * o_{0,b}(t) - v_{0,b}(t) & c_{0,b}(t) > v_{0,b}(t) \end{cases} \tag{22}$$

$$Z_{l,u}(t) = \{A(c_{l,u}(t-1), x(l,u,t), Q) + c_{l,u}(t-1), v_{l,u}(t)\} \tag{23}$$

The parent container level has some basic similarities about the original level, and the reduction function defines the way that the input from the parent container acts on the child container. Thus, the parent container output process can get the definition and can be seen in Equation (24).

$$c_{l,u}(t) = c_{l,u}(t-1) + \sum_{q \in Q} o_{l-1,q}(t) \tag{24}$$

Due to the intrinsic consistency between container and uniform distribution, different container combinations can be replaced by one or more parallel or upward equivalent container combinations. In this study, by encoding the output values of the central container, the discrete state sequence $S = \{s_0, s_1, \ldots, s_u\}$ is established, and the known combination of peripheral container coefficients are traversed to build a probabilistic structure based on input and output, thus obtaining various peripheral output coefficients which owns the uniformity to each central output state, and it can be seen in Equation (25).

$$ST(t) = s_u : [[Q_{u0}, q_{u0}], [Q_{u1}, q_{u1}], \ldots, [Q_{uk_u}, q_{uk_u}]] \tag{25}$$

Where, q_{uv} is the occurrence times of different coefficient combination $Q_{a,b}$ corresponding to the current state s_u. Pattern recognition based on container framework yields peripheral output coefficient table mainly by fitting. The resulting evolution data can be used for pattern classification. Seen from the nature of the method, the container evolution itself is based on the peripheral output coefficient table, with the fitting process equivalent to the training process, so the training method is also referred to as direct fitting method. The core construction idea of the training method is to construct the combination of feature sequences as the input, and there are two strategies for constructing related output: Strategy one is the container fitting system with category as the output, which yields the relevant peripheral output coefficient table through fitting and can be directly used for pattern classification; Strategy two is to realize the mapping evolutionary outputs to intervals of o-1 and output the results through the CUP evolution system. Finally, multi-classification mapping is performed, as shown in Equation (26).

$$S_i = \sum_{j \in J}^{e^{V_i}} e^{V_j} \tag{26}$$

In the real grid environment, the negative impact on the damage node often cannot exceed a certain limit, so a constraint is applied in this study, namely, the upper bound value c of Ψ_2, in which $c > 0$ is the design parameter. In optimal design, the optimal early warning problem can be solved as shown in Equation (27), and the optimal parameters can be calculated as shown in Equation (28). Finally, the fixed point equation of the

optimal early warning parameters can be established as shown in Equation (29).

$$\min_{w,b} \Psi_1(w,b) \text{ s.t. } \Psi_2(w,b) \le c; 0 \le w \le 1; b \ge 0 \tag{27}$$

$$\frac{\partial b(w)}{\partial w} = \frac{c * \bar{\alpha}_c^R}{(1-c) * (c - \epsilon \bar{\alpha}_C^R)} \tag{28}$$

$$\frac{\partial \beta^*}{\partial w} = (1-\beta^*) \frac{\partial q_R^u(\beta^*)}{\partial w} + \beta^* \frac{\partial q_F^u(\beta^*)}{\partial w} + \frac{\partial \beta^*}{\partial w} \left(q_F^u(\beta^*) + \beta^* \frac{\partial q_F^u(\beta^*)}{\partial \beta^*} - q_R^u(\beta^*) + (1-\beta^*) \frac{\partial q_R^u(\beta^*)}{\partial \beta^*} \right) \tag{29}$$

Based on this, the optimized early warning parameters can be solved by the improved algorithm proposed herein, as shown in Equation (30).

$$w_{l+1} = \left[\left| w_l - \kappa_l \frac{\partial \Psi_1}{\partial w} \right|_{(w,b)=(w_l, b(w_t))} \right]_{[0,1] \cap \{b(w) \ge 0\}} \tag{30}$$

Where, $[\cdot]_{\mathcal{A}}$ is the projection of $\mathcal{A}$, and $\{\kappa_l\}$ is the decreasing sequence of step size.

4. **Simulation and empirical research.**

4.1. **Experimental design.** In this study, the three power grid structures used in literature [15, 16, 17] were taken to access the performance of the means which has been put forward. This study realized timing simulation depending on MATLAB software, implemented the process on Python framework through the two servers with Linux operating system (Intel Xeon processor (34GHZ) 64GB memory). Each machine had 6 CPUs. There were two Titan X GPU and 100 GB RAM of NVIDIA. Since the public opinion control model in the experiment may have different results in each run, the evaluation result was set as the average value after 200 iteration runs. The research team used Tensorflow1.5.1 to implement the algorithm proposed herein and make comparison with the correlated algorithms. The experiment was conducted by grouping, dividing the data into 10 groups. The means of cross validation could be adopted, that is, the data set was divided into 10 equal parts, with a group of data selected as the test set each time and the other groups used as the training set. Finally, the average value was calculated, resulting in three cases of 2-fold, 4-fold and 10-fold respectively. All data was saved in CSV format in MySQL database for data processing. Table 1 describes the features of the power grid data set.

Table 1. Feature description of the electrified netting data set.

Network No.	Data set No.	Type	Node number	Number of node edges	Average degree	Node average path	Clustering coefficient
1	Data set 1	Directed	2414	64959	3.19	2.12	0.429
2	Data set 2	Undirected	2310	79610	4.18	3.41	0.120
3	Data set 3	Directed	4285	75592	3.12	2.24	0.245

In this study, the proposed improved Container Hybrid K-Shell (CH-KS) algorithm was compared with reference algorithms including Degree Centrality (DC), Intermediate Centrality(IC), Closeness Centrality (CNC), Ant Colony Optimization (ACO), Swarm Optimization (SWO), K-Shell Centrality (KSC), and Weighted K-Shell Degree Neighborhood (WKS-DN). Based on literature [30], two precision functions were fully utilized:

TABLE 2. Area under curves of each data set in different methods.

Cross validation level	Data set name	Optimization framework of grid container cloud scheduling			
		DC	IC	CNC	ACO
2-fold	Type 1 data set	0.4282	0.3532	0.3893	0.5355
	Type 2 data set	0.4461	0.2411	0.4048	0.4985
	Type 3 data set	0.3019	0.3199	0.3002	0.4810
4-fold	Type 1 data set	0.4318	0.3849	0.4012	0.5838
	Type 2 data set	0.4579	0.3191	0.4127	0.5492
	Type 3 data set	0.3628	0.4292	0.5228	0.5051
10-fold	Type 1 data set	0.4583	0.4055	0.4828	0.6638
	Type 2 data set	0.4910	0.3525	0.5485	0.5729
	Type 3 data set	0.4739	0.4660	0.5829	0.5391
Cross validation level	Data set name	Optimization framework of grid container cloud scheduling			
		SWO	KSC	WKS-DN	CH-KS
2-fold	Type 1 data set	0.6010	0.5939	0.6584	**0.8472**
	Type 2 data set	0.6239	0.5328	0.6749	**0.8329**
	Type 3 data set	0.5029	0.5930	0.7381	**0.8493**
4-fold	Type 1 data set	0.6122	0.6638	0.6692	**0.8938**
	Type 2 data set	0.6283	0.6573	0.6839	**0.8492**
	Type 3 data set	0.6650	0.6019	0.7680	**0.8204**
10-fold	Type 1 data set	0.5383	0.6782	0.6949	**0.8930**
	Type 2 data set	0.5839	0.6839	0.7382	**0.8575**
	Type 3 data set	0.5291	0.6371	0.7029	**0.8957**

Note: All the values in bold indicate that the corresponding algorithm has good performance.

Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). Additionally, their calculation means can be seen in Equations (31) and (32) respectively.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|f_i - y_i| \tag{31}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(f_i - y_i)^2} \tag{32}$$

The power grid optimization problem is usually considered as the task of binary classification. In the evaluation confusion matrix of binary classification tasks with two categories [11], True Positive (TP) illustrates that the quantity of links is predicted in a correct way. True Negative (TN) illustrates the correct quantity of unpredicted links. False Positive (FP) illustrates that the quantity of links is predicted in an incorrect way. False Negative (FN) illustrates the incorrect quantity of unpredicted links. Based on this, the following indexes can be determined, such as true positive rate/recall rate/sensitivity, etc. Where, True Positive Rate (TPR), False Positive Rate (FPR), True Negative Rate (TNR) and Precision can be calculated with reference to literature [26]. Evaluation was performed based on the following two indexes, Area Under the Receiver Operating Characteristics curve (AUROC) [18] and Average Precision (AP) [19]. The ROC curve is the curve between the true positive rate (sensitivity) which locates on the Y-axis and the false

positive rate (1-specificity) which locates on the X-axis. Additionally, the area beneath the ROC curve can be the single point summary statistics between $[0, 1]$, which can be calculated using the trapezoid rule of summarizing all trapezoids under the curve. The link prediction method has a AUROC value greater than 0.5 , and a higher AUROC value indicates better performance of the link prediction method. The average precision is the single point summary value calculated based on different recall thresholds, which is the average precision of the recall value in the interval $[0, 1]$, specifically as shown in Equation (33). Where, p represents the precision under different thresholds of recall rate r, and R is the set of different thresholds.

$$AP = \sum_{k=1}^{R} p(k)\Delta r(k) \tag{33}$$

4.2. **Experimental results.** Table 2 reports the area under the curve of the proposed CH-KS improved hybrid K-Shell algorithm and other reference methods in the power grid data set. It was found in this study that the proposed CH-KS improved hybrid K-Shell algorithm had better experimental results in all data sets. Table 3 shows that the figures of average precision of the proposed CH-KS improved hybrid K-Shell algorithm and other reference algorithms in power grid data sets. The results reveal that the CH-KS improved hybrid K-Shell algorithm proposed herein has a high average precision value in all experimental data sets of power grids. By comparison with the running time which belongs to the algorithm which can be single can be found easily in Table 4, it can be found that compared with other existing algorithms, the CH-KS improved hybrid K-Shell algorithm which is put forward in the research has significantly improved the efficiency when operating in the power grid data set. Table 5 reports that the proposed CH-KS improved hybrid K-Shell algorithm owns lower costs than other compared algorithms when facing topological power grids.

With reference to literature [36], two precision functions were fully utilized: Mean Absolute Error (MAE) and Root Mean Square Error (RMSE). In addition, Table 4 reports MAE and RMSE values of the proposed CH-KS improved hybrid K-Shell algorithm and other reference algorithms in different power grid topologies. The higher the MAE and RMSE figures are, the smaller the precision of the predictive optimization algorithm is. According to Figure 1, the proposed power grid optimization algorithm is generally superior to other methods, because the proposed CH-KS improved hybrid K-Shell algorithm features rapid response, real-time adjustment and optimization of the electrified netting, which can minimize the loss of the electrified netting too.

5. **Conclusion.** Information system is an important component of power grid and an important technical means for differentiated competition of the electrified netting. Nonetheless, with this information system being constructed, there are problems such as increasing system scale and growing need for resources. Through mathematical algorithms and computer simulation operations, better cloud resource allocation solutions can be provided [65, 66, 67, 68, 69, 70]. In order to get the solutions of the issues above, in the beginning, this paper constructed the cloud resource scheduling model, the multi-objective container cloud scheduling model and the complex power grid model to represent the relevant research variables in the power grid cloud resource scheduling and community environment. Then, this paper described the classical critical panel point prediction method in power grid risk control, proposed a critical panel point prediction algorithm depending on improved hybrid K-Shell method, and carried out algorithm optimization based on container cloud scheduling algorithm. By comparing the algorithm model proposed in this article with numerical experiments of similar algorithms, it can be found that

Table 3. Average precision of each data set in different methods.

Cross validation level	Data set name	Optimization framework of grid container cloud scheduling			
		DC	IC	CNC	ACO
2-fold	Type 1 data set	0.2859	0.2729	0.3382	0.3291
	Type 2 data set	0.2039	0.2049	0.4818	0.4293
	Type 3 data set	0.2371	0.2385	0.3919	0.3527
4-fold	Type 1 data set	0.3271	0.3291	0.3418	0.3472
	Type 2 data set	0.3391	0.2410	0.3204	0.4839
	Type 3 data set	0.3204	0.2394	0.4491	0.4311
10-fold	Type 1 data set	0.2395	0.3417	0.4372	0.4638
	Type 2 data set	0.3581	0.3921	0.3340	0.4953
	Type 3 data set	0.3849	0.3100	0.4839	0.4552
Cross validation level	Data set name	Optimization framework of grid container cloud scheduling			
		SWO	KSC	WKS-DN	CH-KS
2-fold	Type 1 data set	0.5395	0.5873	**0.6497**	0.6240
	Type 2 data set	0.5856	0.5443	0.7354	**0.8063**
	Type 3 data set	0.5748	0.6054	0.7703	**0.8352**
4-fold	Type 1 data set	0.5663	0.6281	**0.6617**	0.6512
	Type 2 data set	0.6438	0.5985	0.6794	**0.7445**
	Type 3 data set	0.6192	0.5474	0.7592	**0.8186**
10-fold	Type 1 data set	0.6479	0.5847	0.6873	**0.7299**
	Type 2 data set	0.6753	0.6012	0.6944	**0.7742**
	Type 3 data set	0.6191	0.6126	0.8205	**0.8742**

Note: All the values in bold indicate that the corresponding algorithm has good performance.

Table 4. Comparison results of algorithms in power grid topology.

	Index	DC	IC	CNC	ACO
Actual value	MAE	0.6383	0.6738	0.6239	0.5371
	RMSE	0.7362	0.6988	0.6582	0.5481
Optimal value	MAE	0.5353	0.6074	0.6248	0.5743
	RMSE	0.6353	0.6428	0.6739	0.5938
	Indicator	SWO	KSC	WKS-DN	CH-KS
Actual value	MAE	0.5382	0.5644	0.4272	**0.2371**
	RMSE	0.5463	0.5738	0.4371	**0.2492**
Optimal value	MAE	0.4739	0.5192	0.4472	**0.1281**
	RMSE	0.5291	0.5332	0.4738	**0.2455**

Note: The bold indicates that this algorithm is relatively optimal under this parameter condition.

the complex power grid key node prediction and risk monitoring early warning algorithm based on container cloud scheduling technology constructed in this article has excellent performance.

Future research directions for the innovative algorithm model proposed in this article are as follows. First, in the prediction of key power grid nodes, uncertainty of both nodes and edges can be considered to further improve the ability which belongs to the algorithm

TABLE 5. Comparison results of algorithms in power grid topology for cost.

Index	DC	IC	CNC	ACO
Actual cost	143.53	153.90	110.42	105.53
Optimal cost	135.41	135.45	105.33	119.33
Indicator	SWO	KSC	WKS-DN	CH-KS
Actual cost	115.33	95.45	126.82	**48.31**
Optimal cost	135.86	84.30	93.59	**38.01**

Note: The bold indicates that this algorithm is relatively optimal under this parameter condition.

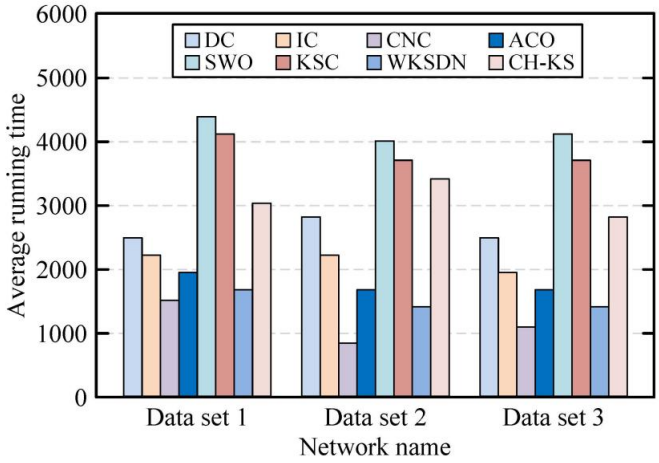

FIGURE 1. Time complexity comparison of algorithms

which can be proposed to resist uncertainty. Second, attempt can be made to optimize the cloud application container management system of power grid scheduling based on Kubernetes framework for the seek of further improving the efficiency and precision of the proposed algorithm. Finally, advanced technologies such as depth autoencoder can be used to get further improvement of the precision and efficiency which belongs to the critical panel points prediction method for power grids.

REFERENCES

[1] J. Zhao, C. Li, and L. Wang, "Hadoop-based power grid data quality verification and monitoring method," *Journal of Electrical Engineering & Technology*, vol. 18, no. 1, pp. 89–97, 2023.

[2] Z. Li, H. Jin, D. Zou, and B. Yuan, "Exploring new opportunities to defeat low-rate ddos attack in container-based cloud environment," *IEEE Transactions on Parallel and Distributed Systems*, vol. 31, no. 3, pp. 695–706, 2019.

[3] S. Raju and M. Chandrasekaran, "An improved neighbor cooperation using adjacent node cooperation in peer to peer networks," *Cluster Computing*, vol. 22, pp. 12 263–12 274, 2019.

[4] S. Choudhury, S. Maheshwari, I. Seskar, and D. Raychaudhuri, "Shareon: Shared resource dynamic container migration framework for real-time support in mobile edge clouds," *IEEE Access*, vol. 10, pp. 66 045–66 060, 2022.

[5] R. Noorian Talouki, M. Hosseini Shirvani, and H. Motameni, "A hybrid meta-heuristic scheduler algorithm for optimization of workflow scheduling in cloud heterogeneous computing environment," *Journal of Engineering, Design and Technology*, vol. 20, no. 6, pp. 1581–1605, 2021.

[6] H. Sun, M. Liu, Z. Qing, X. Li, and L. Li, "Energy consumption optimisation based on mobile edge computing in power grid internet of things nodes," *International Journal of Web and Grid Services*, vol. 16, no. 3, pp. 238–253, 2020.

[7] H. Cao, "The analysis of edge computing combined with cloud computing in strategy optimization of music educational resource scheduling," *International Journal of System Assurance Engineering and Management*, vol. 14, no. 1, pp. 165–175, 2023.

[8] O. Abuzeid, A. Daoud, and M. Barghash, "Optimal off-grid hybrid renewable energy system for residential applications using particle swarm optimization." *Jordan Journal of Mechanical & Industrial Engineering*, vol. 13, no. 2, pp. 117–124, 2019.

[9] J. Li, L. Zhang, and X. Gu, "A composite particle swarm optimization algorithm for hospital equipment management risk control optimization and prediction," *Journal of Environmental and Public Health*, vol. 2022, p. 5268887, 2022.

[10] R. Chackochan, A. Sunny, and S. Dhanasekaran, "Approximate aggregate utility maximization using greedy maximal scheduling," *IEEE/ACM Transactions on Networking*, vol. 30, no. 6, pp. 2521–2530, 2022.

[11] J. Liu and C. Zhu, "No user left behind: dynamic bottleneck-aware allocation of multiple resources," *Cluster Computing*, vol. 22, pp. 10 219–10 227, 2019.

[12] R. Zhu, S. Li, P. Wang, M. Xu, and S. Yu, "Energy-efficient deep reinforced traffic grooming in elastic optical networks for cloud–fog computing," *IEEE Internet of Things Journal*, vol. 8, no. 15, pp. 12 410–12 421, 2021.

[13] K. Sreenu and M. Sreelatha, "W-scheduler: whale optimization for task scheduling in cloud computing," *Cluster Computing*, vol. 22, pp. 1087–1098, 2019.

[14] B. Tan, H. Ma, and Y. Mei, "Novel genetic algorithm with dual chromosome representation for resource allocation in container-based clouds," in *2019 IEEE 12th International Conference on Cloud Computing (CLOUD)*. IEEE, 2019, pp. 452–456.

[15] C. Zhang, Y. Wang, H. Wu, and H. Guo, "An energy-aware host resource management framework for two-tier virtualized cloud data centers," *IEEE Access*, vol. 9, pp. 3526–3544, 2020.

[16] S. Hamdan, M. Ayyash, and S. Almajali, "Edge-computing architectures for internet of things applications: A survey," *Sensors*, vol. 20, no. 22, p. 6441, 2020.

[17] B. Tan, H. Ma, Y. Mei, and M. Zhang, "A cooperative coevolution genetic programming hyper-heuristics approach for on-line resource allocation in container-based clouds," *IEEE Transactions on Cloud Computing*, vol. 10, no. 3, pp. 1500–1514, 2020.

[18] T. Shi, H. Ma, and G. Chen, "Energy-aware container consolidation based on pso in cloud data centers," in *2018 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2018, pp. 1–8.

[19] S. Nabi and M. Ahmed, "Pso-rdal: Particle swarm optimization-based resource-and deadline-aware dynamic load balancer for deadline constrained cloud tasks," *The Journal of Supercomputing*, vol. 78, no. 4, pp. 4624–4654, 2022.

[20] M. Li, X. Wang, Y. Gong, Y. Liu, and C. Jiang, "Binary glowworm swarm optimization for unit commitment," *Journal of Modern Power Systems and Clean Energy*, vol. 2, no. 4, pp. 357–365, 2014.

[21] M. Bansal and S. K. Malik, "A multi-faceted optimization scheduling framework based on the particle swarm optimization algorithm in cloud computing," *Sustainable Computing: Informatics and Systems*, vol. 28, p. 100429, 2020.

[22] A. M. Mohamed and H. M. Abdelsalam, "A multicriteria optimization model for cloud service provider selection in multicloud environments," *Software: Practice and Experience*, vol. 50, no. 6, pp. 925–947, 2020.

[23] H. Liu, S. Duan, and H. Luo, "A hybrid engineering algorithm of the seeker algorithm and particle swarm optimization," *Materials Testing*, vol. 64, no. 7, pp. 1051–1089, 2022.

[24] L. Zuo, S. Dong, L. Shu, C. Zhu, and G. Han, "A multiqueue interlacing peak scheduling method based on tasks' classification in cloud computing," *IEEE Systems Journal*, vol. 12, no. 2, pp. 1518–1530, 2016.

[25] S. Byatarayanapura Venkataswamy, I. Mandal, and S. Keshavarao, "Chicwhale optimization algorithm for the vm migration in cloud computing platform," *Evolutionary Intelligence*, vol. 13, no. 4, pp. 725–739, 2020.

[26] A. F. S. Devaraj, M. Elhoseny, S. Dhanasekaran, E. L. Lydia, and K. Shankar, "Hybridization of firefly and improved multi-objective particle swarm optimization algorithm for energy efficient load balancing in cloud computing environments," *Journal of Parallel and Distributed Computing*, vol. 142, pp. 36–45, 2020.

[27] Y. Gao and Q. Li, "A new framework for the complex system's simulation and analysis," *Cluster Computing*, vol. 22, pp. 9097–9104, 2019.

[28] G. S. Manku, A. Jain, and A. Das Sarma, "Detecting near-duplicates for web crawling," in *Proceedings of the 16th International Conference on World Wide Web*, 2007, pp. 141–150.

[29] H. Wei, J. X. Yu, and C. Lu, "String similarity search: A hash-based approach," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 1, pp. 170–184, 2017.

[30] Q. Z. Guo, Z. Zeng, S. W. Zhang, X. Feng, and H. Guan, "Simhash for large scale image retrieval," in *Applied Mechanics and Materials*, vol. 651. Trans Tech Publ, 2014, pp. 2197–2200.
[31] A. Shrivastava and P. Li, "In defense of minhash over simhash," in *Artificial Intelligence and Statistics*. PMLR, 2014, pp. 886–894.
[32] G. E. Hinton and R. R. Salakhutdinov, "Replicated softmax: an undirected topic model," *Advances in Neural Information Processing Systems*, vol. 22, pp. 1607–1614, 2009.
[33] R. Zunino and P. Gastaldo, "Analog implementation of the softmax function," in *2002 IEEE International Symposium on Circuits and Systems. Proceedings (Cat. No. 02CH37353)*, vol. 2. IEEE, 2002, pp. II–II.
[34] G. Jēkabsons, "Evaluation of fingerprint selection algorithms for local text reuse detection," *Applied Computer Systems*, vol. 25, no. 1, pp. 11–18, 2020.
[35] R. Sutoyo, I. Ramadhani, A. D. Ardiatma, S. C. Bavana, H. L. H. S. Warnars, A. Trisetyarso, B. S. Abbas, and W. Suparta, "Detecting documents plagiarism using winnowing algorithm and k-gram method," in *2017 IEEE International Conference on Cybernetics and Computational Intelligence (CyberneticsCom)*. IEEE, 2017, pp. 67–72.
[36] M. E. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, no. 2, pp. 167–256, 2003.
[37] J. Duch and A. Arenas, "Community detection in complex networks using extremal optimization," *Physical Review E*, vol. 72, no. 2, p. 027104, 2005.
[38] Z. Wang, M. A. Andrews, Z.-X. Wu, L. Wang, and C. T. Bauch, "Coupled disease–behavior dynamics on complex networks: A review," *Physics of Life Reviews*, vol. 15, pp. 1–29, 2015.
[39] T. Cao, X. Wu, S. Wang, and X. Hu, "Maximizing influence spread in modular social networks by optimal resource allocation," *Expert Systems with Applications*, vol. 38, no. 10, pp. 13 128–13 135, 2011.
[40] S. M. H. Bamakan, I. Nurgaliev, and Q. Qu, "Opinion leader detection: A methodological review," *Expert Systems with Applications*, vol. 115, pp. 200–222, 2019.
[41] M. Gupta and R. Mishra, "Spreading the information in complex networks: Identifying a set of top-n influential nodes using network structure," *Decision Support Systems*, vol. 149, p. 113608, 2021.
[42] D. Kempe, J. Kleinberg, and É. Tardos, "Maximizing the spread of influence through a social network," in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2003, pp. 137–146.
[43] A. Zareie, A. Sheikhahmadi, and K. Khamforoosh, "Influence maximization in social networks based on topsis," *Expert Systems with Applications*, vol. 108, pp. 96–107, 2018.
[44] P. Bonacich, "Factoring and weighting approaches to status scores and clique identification," *Journal of Mathematical Sociology*, vol. 2, no. 1, pp. 113–120, 1972.
[45] M. E. Newman, "Spread of epidemic disease on networks," *Physical Review E*, vol. 66, no. 1, p. 016128, 2002.
[46] G. Sabidussi, "The centrality index of a graph," *Psychometrika*, vol. 31, no. 4, pp. 581–603, 1966.
[47] H.-L. Liu, C. Ma, B.-B. Xiang, M. Tang, and H.-F. Zhang, "Identifying multiple influential spreaders based on generalized closeness centrality," *Physica A: Statistical Mechanics and its Applications*, vol. 492, pp. 2237–2248, 2018.
[48] B. Pittel, J. Spencer, and N. C. Wormald, "Sudden emergence of a giantk-core in a random graph," *Journal of Combinatorial Theory, Series B*, vol. 67, no. 1, pp. 111–151, 1996.
[49] X. Ye and S. Wei, "A primary way of solving sampling bias problem in complex internet topology," *International Journal of Future Generation Communication and Networking*, vol. 9, no. 1, pp. 319–328, 2016.
[50] D.-B. Chen, H. Gao, L. Lü, and T. Zhou, "Identifying influential nodes in large-scale directed networks: the role of clustering," *PloS One*, vol. 8, no. 10, p. e77455, 2013.
[51] K. Rahimkhani, A. Aleahmad, M. Rahgozar, and A. Moeini, "A fast algorithm for finding most influential people based on the linear threshold model," *Expert Systems with Applications*, vol. 42, no. 3, pp. 1353–1361, 2015.
[52] L. Lü, D. Chen, X.-L. Ren, Q.-M. Zhang, Y.-C. Zhang, and T. Zhou, "Vital nodes identification in complex networks," *Physics Reports*, vol. 650, pp. 1–63, 2016.
[53] A. Zareie, A. Sheikhahmadi, and M. Jalili, "Influential node ranking in social networks based on neighborhood diversity," *Future Generation Computer Systems*, vol. 94, pp. 120–129, 2019.
[54] L.-l. Ma, C. Ma, H.-F. Zhang, and B.-H. Wang, "Identifying influential spreaders in complex networks based on gravity formula," *Physica A: Statistical Mechanics and its Applications*, vol. 451, pp. 205–212, 2016.

[55] A. Zareie and A. Sheikhahmadi, "A hierarchical approach for influential node ranking in complex social networks," *Expert Systems with Applications*, vol. 93, pp. 200–211, 2018.
[56] F. Morone and H. A. Makse, "Influence maximization in complex networks through optimal percolation," *Nature*, vol. 524, no. 7563, pp. 65–68, 2015.
[57] R. Narayanam and Y. Narahari, "A shapley value-based approach to discover influential nodes in social networks," *IEEE Transactions on Automation Science and Engineering*, vol. 8, no. 1, pp. 130–147, 2010.
[58] A. ŞİMŞEK and K. Resul, "Using swarm intelligence algorithms to detect influential individuals for influence maximization in social networks," *Expert Systems with Applications*, vol. 114, pp. 224–236, 2018.
[59] J. Zhan, S. Gurung, and S. P. K. Parsa, "Identification of top-k nodes in large networks using katz centrality," *Journal of Big Data*, vol. 4, no. 1, pp. 1–19, 2017.
[60] J. Wang, X. Chen, F. Zhang, F. Chen, and Y. Xin, "Building load forecasting using deep neural network with efficient feature fusion," *Journal of Modern Power Systems and Clean Energy*, vol. 9, no. 1, pp. 160–169, 2021.
[61] J. Wang, F. Zhang, H. Liu, J. Ding, and C. Gao, "Interruptible load scheduling model based on an improved chicken swarm optimization algorithm," *CSEE Journal of Power and Energy Systems*, vol. 7, no. 2, pp. 232–240, 2020.
[62] N. Jia, J. Wang, and N. Li, "Application of data mining in intelligent power consumption," in *International Conference on Automatic Control and Artificial Intelligence (ACAI 2012)*. IET, 2012, pp. 538–541.
[63] X. Zheng, H. Jia, and J. Wang, "Energy internet development based on blockchain technology," in *ICCREM 2019: Innovative Construction Project Management and Construction Industrialization*. American Society of Civil Engineers Reston, VA, 2019, pp. 167–178.
[64] S. Zhang, J. Wang, H. Liu, J. Tong, and Z. Sun, "Prediction of energy photovoltaic power generation based on artificial intelligence algorithm," *Neural Computing and Applications*, vol. 33, pp. 821–835, 2021.
[65] C.-M. Chen, Y. Hao, and T.-Y. Wu, "Discussion of "ultra super fast authentication protocol for electric vehicle charging using extended chaotic maps"," *IEEE Transactions on Industry Applications*, vol. 59, no. 2, pp. 2091–2092, 2023.
[66] T.-Y. Wu, A. Shao, and J.-S. Pan, "Ctoa: Toward a chaotic-based tumbleweed optimization algorithm," *Mathematics*, vol. 11, no. 10, p. 2339, 2023.
[67] T.-Y. Wu, H. Li, and S.-C. Chu, "Cppe: An improved phasmatodea population evolution algorithm with chaotic maps," *Mathematics*, vol. 11, no. 9, p. 1977, 2023.
[68] C.-M. Chen, L. Chen, Y. Huang, S. Kumar, and J. M.-T. Wu, "Lightweight authentication protocol in edge-based smart grid environment," *EURASIP Journal on Wireless Communications and Networking*, vol. 2021, pp. 1–18, 2021.
[69] T.-Y. Wu, Y.-Q. Lee, C.-M. Chen, Y. Tian, and N. A. Al-Nabhan, "An enhanced pairing-based authentication scheme for smart grid communications," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1–13, 2021.
[70] H. Xiong, Y. Wang, W. Li, and C.-M. Chen, "Flexible, efficient, and secure access delegation in cloud computing," *ACM Transactions on Management Information Systems (TMIS)*, vol. 10, no. 1, pp. 1–20, 2019.