Energy Reports 9 (2023) 3544–3552

Contents lists available at ScienceDirect

Energy Reports

journal homepage: www.elsevier.com/locate/egyr

Research paper

Graph convolutional network-based security-constrained unit commitment leveraging power grid topology in learning

Xian Tang, Xiaoqing Bai *, Zonglong Weng, Rui Wang

Guangxi Key Laboratory of Power System Optimization and Energy Saving Technology, School of Electrical Engineering, Guangxi University, Nanning 530004, China

ARTICLE INFO

Article history:
Received 25 October 2022
Received in revised form 12 February 2023
Accepted 18 February 2023
Available online xxxx

Keywords:
Security-constrained unit commitment
Graph convolutional network
GCN-SCUC

ABSTRACT

Security-constrained unit commitment (SCUC) is a complex optimization problem in power system operation, which is computationally intensive. To bring significant time-savings, this paper presents a graph convolutional network (GCN)-based SCUC approach (GCN-SCUC) using the information of power grid topology. Instead of tackling the mixed integer linear programming (MILP)-based SCUC (MILP-SCUC), the GCN learner predicts the unit decisions first, and then the MILP-SCUC problem is transformed into a continuous convex one. Numerical experiments are performed on the modified IEEE-30 and IEEE-118 systems to verify the feasibility of our approach both in terms of accuracy and computation time. Moreover, compared with the state-of-the-art MILP-SCUC, the proposed approach achieves speedups of between 13x and 17x on different testing examples with high accuracy.

1. Introduction

As an extension problem of the traditional unit commitment model, the SCUC problem is an essential decision-making tool for power system operation (Yang et al., 2022a). The best possible scheduling of unit commitment needs to be obtained by considering various unit and system constraints (Muralikrishnan et al., 2020). Although utilities use the DC-based SCUC problem in practice for computational tractability, the SCUC problem is naturally AC-based with voltage and reactive power constraints. Mathematically, the SCUC problem is a large-scale mixed integer programming (MIP) problem that is proven to be NP-hard (Pineda and Morales, 2022; Fu et al., 2013). The nonlinear AC power flow equations and vast integer variables lead to a high computation burden. Therefore, it is crucial to study effective methods for trimming down its computational burden with almost no loss of solution quality. There are several research achievements, either of the physical-model-driven method or the data-driven method, related to the SCUC problem. See (Yang et al., 2022a) for a recent survey.

The physical-model-driven approach is based on rigorous logic deductions, which are supported by mathematical theories either in model construction or model compacting. Safdarian et al. (2020) decomposed the scheduling horizon into multiple subhorizons to speed up the efficiency of solving the SCUC problem. A set of auxiliary logical expressions and counting variables was formulated to handle intertemporal ramp constraints and minimum on/off times between consecutive subhorizons. In Chen et al. (2020), by decomposing the system into several zones, a distributed optimization framework was proposed to reduce the size of the SCUC problem, which can be solved in parallel. Since compacting the model is an effective way to enhance the computational efficiency of SCUC, the details about the reduction of the variable strategy and redundant constraints removing the strategy-based physical model can be found in Li et al. (2020), Zhai et al. (2010). In comparison, the authors of Du et al. (2019) proposed a linearized model with minor approximation errors. Generally, a more refined model can result in more accurate results. However, model refinements usually aggravate the computational complexity. Therefore, the computational efficiency has been dramatically improved by tightening or linearizing the SCUC model.

The data-driven method is based on historical data, which has been applied to various fields of power systems, such as load forecasting (Sakkas and Abang, 2022), unit on-off state prediction, and so on. Long short-term memory (LSTM) was introduced in Yang et al. (2019), which was the first time to apply the data-driven method for the SCUC problem. The abilities of self-learning and self-evolution were obtained by keeping revising the model. On the basis of Yang et al. (2019, 2022b) further developed the LSTM into the sequence-to-sequence (E-Seq2Seq) method. The results indicated that, compared with the physical-model-driven methods, the data-driven method possessed strong generality, high solution accuracy, and efficiency. Information from historical

* Corresponding author.
E-mail addresses: tangx@st.gxu.edu.cn (X. Tang), baixq@gxu.edu.cn (X. Bai), wzl98520@163.com (Z. Weng), 13059048171@163.com (R. Wang).

https://doi.org/10.1016/j.egyr.2023.02.042

Nomenclature

Indices and Sets

t	Period index
T	Periods set
N_B	Buses set
N_G	Units set
N_L	Branch lines set

Constants

c_i	Fixed coefficient of cost in unit i
b_i	Linear coefficient of cost in unit i
a_i	Quadratic coefficient of cost in unit i
FP_i	Fuel price of the unit i
$\underline{SR}_t$	Spinning reserve in period t
$\overline{P_l}$	Upper bound of transmission line power
$CU_{i,t}$	Startup fuel consumption of unit i in the period t
$P_{Di,t}$	Active load in the bus i at the period t
$Q_{Di,t}$	Reactive load in the bus i at the period t
$\Delta p_{i,t}$	Difference between the upper and lower bound of active load power in the bus i at the period t
$\Delta q_{i,t}$	Difference between the upper and lower bound of reactive load power in the bus i at the period t
$\underline{V}_i/\overline{V}_i$	Lower/upper voltage in the bus i
G_{ij}/B_{ij}	Real/imaginary part of the admittance matrix
R_i^D/R_i^U	Ramp down/up limit of the unit i
$U_{i,T}/D_{i,T}$	Minimum down/uptime of the unit i
$\overline{P}_{Gi}/\underline{P}_{Gi}$	Upper/lower limit of active power at the unit i
$\overline{Q}_{Gi}/\underline{Q}_{Gi}$	Upper/lower limit of reactive power at the unit i
$T_{i,t}^{on}/T_{i,t}^{off}$	On/off time of unit i at the period t
$p_{Di,t}^L/p_{Di,t}^U$	Lower/upper bound of active load power in the bus i at the period t
$q_{Di,t}^L/q_{Di,t}^U$	Lower/upper bound of reactive load power in the bus i at the period t

Variables

$u_{i,t}$	Binary equal to 1 if the unit is online at the period t
$P_{Gi,t}$	Per unit active power capacity at the period t
$Q_{Gi,t}$	Per unit reactive power capacity at the period t
$V_{i,t}$	The voltage in the bus i at the period t
$P_{li,t}$	Transmission line power in the branch i at the period t
$\Delta P_{i,t}$	Unbalanced active power in the bus i at the period t
$\Delta Q_{i,t}$	Unbalanced reactive power in the bus i at the period t
$\eta_{i,t}$	Abides by a uniform distribution between 0 and 1.

data is extracted by machine learning, which can improve the performance of similar SCUC instances in the future.

In recent years, researchers have focused on studying how machine-learning approaches may refine power system optimization. Deep integration of optimization models and machine learning approaches is expected to be the most promising technology trend (Ruan et al., 2021). The model-data-driven method usually combines artificial intelligence technologies with the model-based method to solve the SCUC problem. Generally, the data-driven method was utilized as preprocess steps (Yang et al., 2020; Pineda et al., 2020). A data-driven classification approach to identify whether the SCUC instance is easy or hard, integrating a variable-aggregation method, is presented in Yang et al. (2020). A support vector machine was introduced to predict redundant constraints in the formulation, reasonable initial feasible solutions, and affine subspaces, which can speed up the computational efficiency of the MILP solver (Xavier et al., 2020). Moreover, the authors in Wu et al. (2022a) proposed a convolutional neural network (CNN) to relieve the computation burden of the SCUC problem.

Both data-driven and model-data-driven approaches can relieve the computational burden of the SCUC problem. The power grid is a complex network consisting of buses (generation points, load points, etc.) connected by transmission lines. Therefore, they can be conveniently represented as an undirected graph $G = (V, E)$, where $E \subseteq V \times V$ and V denote the sets of edges and nodes (Fig. 1). The nodes include generator and load buses, respectively. The data of SCUC with the network are defined as graph-structured data with nodes and edges (Liao et al., 2022). However, the existing deep learning methods in Euclidean domains, such as recurrent neural network (RNN) and CNN, ignore the topological information and need various samples for training.

The Graph neural networks (GNN), neural network architectures targeted to learning representations of graphs, such as graph convolutional networks (GCN), have shown good performance on graph-structured data. The learning model for prediction tasks is successful in practice on nodes, graphs, and configurations of points. Based on graph neural networks, a graph neural solver was developed to solve AC power flow for balancing computational speed and accuracy (Donon et al., 2020). In Liao et al. (2020), the GCN proved that the accuracy and robustness were better than the convolutional neural network in settling the reactive power optimization. In Liao et al. (2021), a spectral-based GCN is presented to prompt the accuracy of transformer fault diagnosis. While the authors of Khodayar and Wang (2019), Wu et al. (2022b) used the spatial–temporal graph convolutional Networks for wind speed and short-term load forecasting, respectively. Furthermore, Nair et al. (2020) encodes MIP to the GCN as a bipartite graph and computes the initial feasible solution. Qin et al. (2022) applied GCN to achieve distribution network situation awareness.

Motivated by the model-data-driven methods and the GCN method for solving problems in power systems, in this paper, a novel method based on GCN is proposed to predict the state of units of SCUC problem first. Then, the outputs of units and the voltage values can be obtained by tackling a convex problem. The key contributions of this paper are threefold:

- GCN is introduced to assist in solving the SCUC problem, which is a typical scenario of machine learning applications in the electrical field. The on-off variable of the thermal power unit is the label of GCN learning. In contrast to existing machine learning work, GCN can learn information regarding network topology in the training process, achieving high prediction accuracy with fewer samples.
- A combination of the data-driven and model-driven method, that is, a GCN-based learner and a convex optimization problem, solves the SCUC problem. If the zero–one variable is not feasible, it can be used as the initial value to solve the SCUC problem. The proposed GCN-based SCUC algorithm (GCN-SCUC) allows us to find a primal feasible solution.

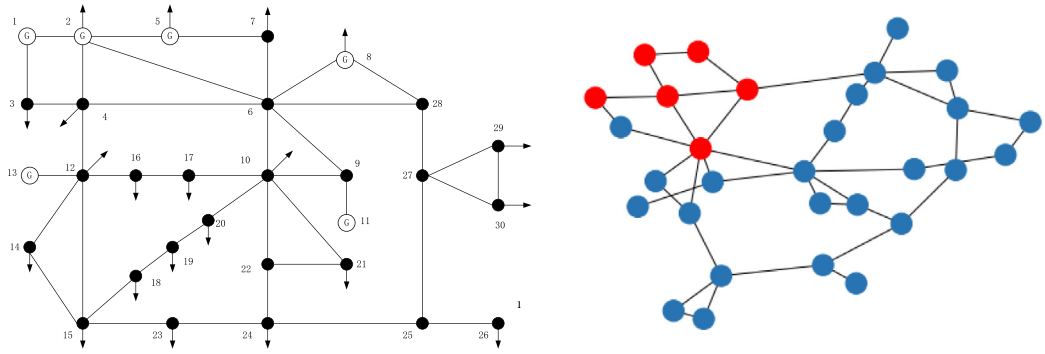

Fig. 1. One-line diagram (left) and corresponding graphical representation (right) for synthetic grid 30-IEEE, blue and red circles denote load buses and buses with generator, respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

- The proposed GCN-SCUC method reduces the solution time and outperforms the MILP-SCUC method in terms of computational efficiency with a lower degree of suboptimality or infeasibility. The computational efficiency of the proposed method achieves speedups of an average of about 15x for testing power systems.

The remainder of this paper is laid out as follows: in Section 2, the methodology is introduced, including the mathematical formulation of the SCUC problem and the detail about the GCN-SCUC method. Section 3 provides simulation results of different testing systems, followed by a conclusion and future work in Section 4.

2. Methodology

In this section, the original mathematical model of the SCUC problem is introduced, referring to the equations in Bai and Wei (2009). Then, the detail for the process of the GCN-based SCUC algorithm (GCN-SCUC) applied to tackle the SCUC problem is described. Fig. 2 shows the framework of the proposed method.

2.1. Problem formulation

The SCUC problem can be formulated in (1). The objective of SCUC is to minimize the startup cost of units and the system operating cost, just as follows:

$$\min \sum_{t=1}^{T}\sum_{i=1}^{N_G} FP_i(u_{i,t}f_i(P_{Gi,t}) + u_{i,t}(1-u_{i,t-1})CU_{i,t}) \tag{1a}$$

$$s.t. \quad (2)\text{–}(12) \tag{1b}$$

where $f_i(P_{Gi,t}) = a_iP_{Gi,t}^2 + b_iP_{Gi,t} + c_i$ represents the fuel consumption function of the unit, a_i is the quadratic coefficient of cost in the unit i, b_i is the linear coefficient of cost in the unit i, c_i is the fixed coefficient of cost in the unit i. t is the period index, T is period set, N_G is units set. FP_i denotes fuel price of the unit, $u_{i,t}$ binary equal to 1 if the unit is online at the period t. $P_{Gi,t}$ represents per unit active power capacity at the period t, $CU_{i,t}$ is the startup fuel consumption of the unit i in the period t. In this model, the objective function should be minimized, subjecting the constraints as follows:

(1) AC power flow constraints

$$\begin{cases} \Delta P_{i,t} = \sum_{j\in N_B} V_{i,t}V_{j,t}G_{ij}cos\theta_{ij,t} + \sum_{j\in N_B} V_{i,t}V_{j,t}B_{ij}sin\theta_{ij,t} \\ \Delta P_{i,t} = P_{Gi,t} - P_{Di,t}, \ i \in N_B, t \in T \end{cases} \tag{2}$$

$$\begin{cases} \Delta Q_{i,t} = \sum_{j\in N_B} V_{i,t}V_{j,t}G_{ij}sin\theta_{ij,t} - \sum_{j\in N_B} V_{i,t}V_{j,t}B_{ij}cos\theta_{ij,t} \\ \Delta Q_{i,t} = Q_{Gi,t} - Q_{Di,t}, \ i \in N_B, t \in T \end{cases} \tag{3}$$

where $\Delta P_{i,t}/\Delta Q_{i,t}$ refers to unbalanced active/reactive power in the bus i at the period t, N_B is buses set, $V_{i,t}$ is the voltage in the bus i at the period t, G_{ij}/B_{ij} is the real/imaginary part of the admittance matrix. $Q_{Gi,t}$ is per unit reactive power capacity at the period t, if the bus has no generators, $P_{Gi,t}/Q_{Gi,t}$ is equal to 0. $P_{Di,t}/Q_{Di,t}$ is active/reactive load in the bus i at the period t. $\theta_{ij,t}$ is phase angle difference at the period t. These constraints ensure power balance.

(2) Branch flow limitations

$$-\overline{P_l} \le P_{li,t} \le \overline{P_l}, \ i \in N_L, t \in T \tag{4}$$

$$P_{li,t} = V_{i,t}^2G_{ij} - V_{i,t}V_{j,t}(G_{ij}cos\theta_{ij,t} + B_{ij}sin\theta_{ij,t}), \quad (i,j) \in N_L, t \in T \tag{5}$$

where $\overline{P_l}$ is the upper bound of transmission line power, $P_{li,t}$ is transmission line power in the branch i at the period t, N_L is branch lines set. The above constraints ensure that branch power does not cross the boundary.

(3) Ramping up/down limitations

$$P_{Gi,t} - P_{Gi,t-1} \le R_i^U, \ i \in N_G, t \in T \tag{6}$$

$$P_{Gi,t-1} - P_{Gi,t} \le R_i^D, \ i \in N_G, t \in T \tag{7}$$

where R_i^D/R_i^U is the ramp down/up limit of the unit i. The ramping constraints limit the fluctuation of the power output of the unit.

(4) Minimum up/downtime limitations

$$\begin{cases} (u_{i,t-1} - u_{i,t})(T_{i,t-1}^{on} - U_{i,T}) \ge 0 \\ (u_{i,t} - u_{i,t-1})(T_{i,t-1}^{off} - D_{i,T}) \ge 0 \end{cases}, \ i \in N_G, t \in T \tag{8}$$

where $U_{i,T}/D_{i,T}$ is the minimum down/uptime of the unit i. $T_{i,t-1}^{on}/T_{i,t-1}^{off}$ is on/off time of unit i at the period $t-1$. The above constraints ensure the requirements that the unit i lasts $U_{i,T}$ periods when turns on and $D_{i,T}$ when turns off.

(5) Active and reactive power limitations

$$\underline{P}_{Gi}u_{i,t} \le P_{Gi,t} \le \overline{P}_{Gi}u_{i,t}, \ i \in N_G, t \in T \tag{9}$$

$$\underline{Q}_{Gi}u_{i,t} \le Q_{Gi,t} \le \overline{Q}_{Gi}u_{i,t}, \ i \in N_G, t \in T \tag{10}$$

where $\overline{P}_{Gi}/\underline{P}_{Gi}$ is the upper/lower limit of active power at the unit i, $\overline{Q}_{Gi}/\underline{Q}_{Gi}$ is the upper/lower limit of reactive power at the unit i. The unit's output power needs to be limited between the upper and lower bound when the unit turns on.

(6) Voltage limitations

$$\underline{V}_i \le V_{i,t} \le \overline{V}_i, \ i \in N_B, t \in T \tag{11}$$

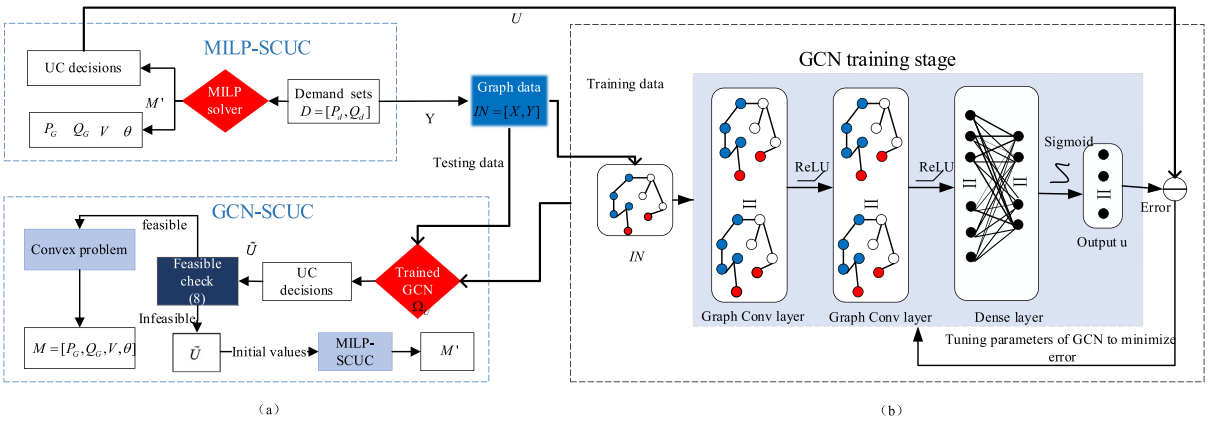

Fig. 2. The framework of the GCN-SCUC method.

where $\underline{V}_i/\overline{V}_i$ is lower/upper voltage in the bus i. The voltage in the bus i needs to be limited between the upper bound and lower bound.

(7) Spinning reserves limitations

$$\sum_{i\in N_G}(u_{i,t}\overline{P}_{Gi} - u_{i,t}P_{Gi,t}) \geq SR_t,\ \ i \in N_G, t \in T \tag{12}$$

where SR_t is the spinning reserve in period t. The above constraints ensure that the unit meets the spinning reserves of the system.

In order to build the MILP model, which can make the SCUC problem be solved by the MILP solver. Especially, the AC power flow constraints and minimum up/downtime limitations have to be linearized, and the linearized process can refer to Yang et al. (2017), Zhu et al. (2019). The MILP model can be formulated like this:

$$\begin{aligned} &\min c^T x \\ &s.t.\quad Ax \leq b \\ &\qquad l \leq x \leq u \\ &\qquad x_i \in \{0, 1\}, i \in \mathrm{I} \end{aligned} \tag{13}$$

where $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$. l and u are the lower or upper bound of variables, respectively. $\mathrm{I} \subseteq \{1, \ldots, n\}$ refers to the index of integer variables.

2.2. GCN-based approximation to SCUC

The GCN-based SCUC method (GCN-SCUC) framework to tackle the SCUC problem is displayed in Fig. 2, containing the GCN training stage and the convex optimization problem calculating part. The detailed information is introduced as follows.

2.2.1. GCN training process

Before solving SCUC, demand information and typology are available. The entire learning process can be regarded as node classification in graph learning, while each node is assigned a feature vector x_n. x_n contains the load and unit parameters. The admittance matrix Y is a weighted adjacency matrix. We propose the following vector IN expressing the input for the trained GCN.

$$IN = [X, Y] \tag{14}$$

$$X = \begin{bmatrix} x_1^{\mathrm{T}} \\ \vdots \\ x_{N_B}^{\mathrm{T}} \end{bmatrix} \tag{15}$$

$$x_n = [P_{Dn}, Q_{Dn}, \underline{P}_{Gi}, \overline{P}_{Gi}, \underline{Q}_{Gi}, \overline{Q}_{Gi}]^{\mathrm{T}} \in R^6, i \in N_G, n \in N_B \tag{16}$$

In order to cover possible loading situations that may occur during the operation of the system in the training stage concerning the equation in Hasan et al. (2021), a set of demand scenarios are generated as follows:

$$\begin{cases} \Delta p_{i,t} = p^U_{Di,t} - p^L_{Di,t} \\ p_{i,t} = p^L_{Di,t} + \Delta p_{i,t} \times \eta_{i,t} \\ P_D = \left[p_{1,1} \ldots p_{i,1} \ldots p_{i,t}\right],\ i \in N_B, t \in T \end{cases} \tag{17}$$

$$\begin{cases} \Delta q_{i,t} = q^U_{Di,t} - q^L_{Di,t} \\ q_{i,t} = q^L_{Di,t} + \Delta q_{i,t} \times \eta_{i,t} \\ Q_D = \left[q_{1,1} \ldots q_{i,1} \ldots q_{i,t}\right], i \in N_B, t \in T \end{cases} \tag{18}$$

where $\eta_{i,t}$ abides by a uniform distribution between 0 and 1. $\Delta p_{i,t}$, $\Delta q_{i,t}$ denote the difference between the maximum and minimum bound of active and reactive load power, respectively. Based on the demand provided with the standard IEEE system, the possible minimum $(p^L_{Di,t}/q^L_{Di,t})$ and maximum $(p^U_{Di,t}/q^U_{Di,t})$ nodal demand values can be obtained. For each demand scenario, MILP-SCUC is solved, and the optimal UC decisions are stored in the set U. Demand scenarios making the SCUC problem infeasible are deleted.

By training the GCN to predict unit decisions, the trained GCN Ω_U may capture the mapping relation of the node load, topology structure, and units' state. Which can be expressed as

$$IN \mapsto u_i, \forall i = 1, \ldots, N_G \tag{19}$$

the input vector to Ω_U is IN, and their target is U. The pseudocode for training GCN is represented in Algorithm 1.

2.2.2. Optimization - GCN-SCUC

Once the GCN is trained for a given demand and typology, the output Ω_U is unit decisions $\widetilde{U}$. $\widetilde{U}$ is used to construct the convex optimization problem.

$$\min \sum_{t=1}^{T}\sum_{i=1}^{N_G} FP_i(f_i(P_{Gi,t}) + CU_{i,t}) \tag{20a}$$

$$s.t.\ \ (2)\text{–}(7), (11) \tag{20b}$$

$$\underline{P}_{Gi} \leq P_{Gi,t} \leq \overline{P}_{Gi},\ \ i \in N_G, t \in T \tag{20c}$$

$$\underline{Q}_{Gi} \leq Q_{Gi,t} \leq \overline{Q}_{Gi},\ \ i \in N_G, t \in T \tag{20d}$$

$$\sum_{i\in N_G}(\overline{P}_{Gi} - P_{Gi,t}) \geq SR_t,\ \ i \in N_G, t \in T \tag{20e}$$

Constraints (2)–(5) refer to the linearized ones. Usually, neural network predictions may endanger infeasibility. Therefore, it is

Algorithm 1: GCN Training Process

Step1. Dataset: Generate demand scenarios set by (17)-(18), and form the demand set

$D=[P_d,Q_d]$

Step 2. Solve the MILP-SCUC for each scenario in D and discard infeasible cases

Step 3. Form input data $IN=[X,Y]$

Step 4. Form labeled data $U=[u_1,...,u_{N_G}]$

Step 5. Train Ω_U using IN (input) and U (target) in the dataset

necessary to demonstrate whether the result is a feasible solution or not. According to Fig. 2(a), if the prediction result is feasible, by solving a convex optimization problem, the outputs of units and the voltage values can be obtained. Where M represents the solution set of continuous variables, P_G is the active power solution set, Q_G is the reactive power solution set, V is the voltage amplitude solution set, θ is the voltage phase angle solution set. However, in rare cases, when the prediction result is infeasible, the unit decisions $\widetilde{U}$ predicted by Ω_U are not the feasible solution for the SCUC problem. We solve problem (13) to obtain feasible results. The pseudocode for GCN-SCUC is as follows.

Algorithm 2: GCN-SCUC

Step 1. For a given vector $IN=[X,Y]$, run Ω_U to determine $\tilde{U}$

Step 2. Feasible checking

if feasible

Formulated convex problem using $\tilde{U}$

Minimize (20) subject to (20b)- (20e)

return $M=[P_G,Q_G,V,\theta]$

else

Taking $\tilde{U}$ as the initial value (warming start)

Solve problem (13)

return $M'=[U,P_G,Q_G,V,\theta]$

3. Numerical experiments

The proposed method GCN-SCUC is validated on the modified IEEE 30-bus system (case 30) with 6 generators and 41 branches (Zimmerman et al., 2011; Zimmerman and Murillo-Sanchez, 2020). Furthermore, the modified IEEE 118-bus system (case 118) with 54 generators and 186 branches is utilized to show the effectiveness and scalability of GCN-SCUC. Two systems covering a 24-hour period are considered. Notably, all the analysis is under the condition of the power grids with a given topology where all grid connections are the same among the samples. The MILP optimization problem is programmed in MATLAB with Gurobi (Gurobi Optimization, 2021) running on CPU (Intel(R) Core (TM) i7-8700, 3.2 GHz), and the MIP gap is set to be 10e−3. We utilize GPU (NVIDIA RTX A6000) for the GCN predictions.

3.1. Parameters description

Data by simulation are applied for training and testing the GCN model. According to Algorithm 1, labeled 500 and 300 samples are generated by solving the MILP-SCUC with Gurobi for case 30 and case 118, respectively. Then, transformed the data to graph data like (14), where each timestamp denotes a complete graph state. Similarly, collecting each timestamp data created a dataset to load into the model.

The GCN used in this paper is the stack of 4 graph convolution layers with a mean aggregator to propagate the calculated features to the hidden layers and 2 linear layers, activated by the ReLU function. The sigmoid function transformed results from linear layers to get the final results. Adam optimizer by minimizing the cross-entropy loss ρ_{GCN} between the targeted U and the predicted output of the GCN.

$$\rho_{GCN}=-\frac{1}{k}\sum_j(u_j^*\log p_j^*+(1-u_j^*)\log(1-p_j^*)) \tag{21}$$

where $j\in N_G$. p_j^* is the predicted probability that the unit j is on or off, while u_j^* is the ground-truth state of units. 90% of the samples are used for the training and 10% for the testing sets. The number of training epochs is 200, and the learning rate is set to 0.005. Parameter optimization is performed using an exhaustive search over set limits for each parameter, and the best results are selected for each group.

The accuracy (Acc) and precision (Pre) are leveraged to assess the performance of the trained GCN, while the average cost gap is utilized to measure how close the MILP-SCUC and GCN-SCUC solutions are. The formulas are referred to (Hasan et al., 2021).

$$\text{Acc}=\frac{T_0+T_1}{T_0+T_1+F_0+F_1}\times 100\% \tag{22}$$

$$\text{Pre}=\frac{T_1}{T_1+F_1}\times 100\% \tag{23}$$

$$\text{Cost gap}=\left|\frac{C_{GCN-SCUC}-C_{MILP-SCUC}}{C_{MILP-SCUC}}\right|\times 100\% \tag{24}$$

where T_0/F_0 represent the number of units off is predicted true or false, T_1/F_1 denote the number of units on is predicted true or false. $C_{GCN-SCUC}$ is the cost of the GCN-SCUC method, while $C_{MILP-SCUC}$ is the cost of the MILP-SCUC method.

3.2. Case 30

3.2.1. Comparison of predicted performance

GCN is used for predicting the state of units, taking CNN and DNN as the benchmark. Three types of trained models with different samples have been constructed to demonstrate the GCN model can obtain higher accuracy with fewer samples. (1) Model

Table 1
The performance of trained GCN, trained CNN and trained DNN.

Model	Acc(%)			Pre(%)		
Method	Model 1	Model 2	Model 3	Model 1	Model 2	Model 3
GCN	80.2	86	86.7	78.0	88.2	85.4
CNN	67.7	68.9	69.7	73.5	74.5	71.3
DNN	53.5	66.39	72.2	11.3	40.9	52.3

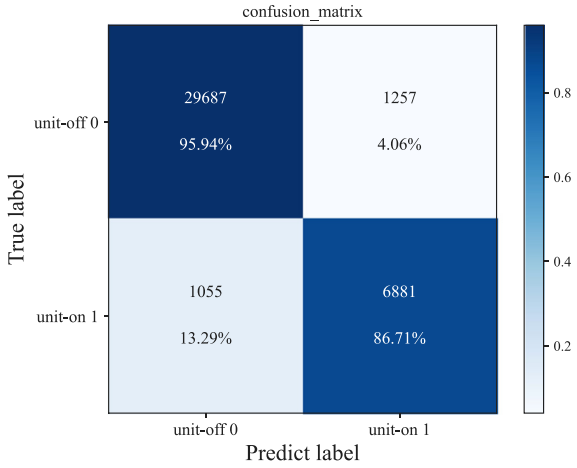

Fig. 3. The confusion matrix of case 30 (Model 2). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

1: 31 samples, including 2 samples to test the trained model; (2) Model 2: 100 samples, including 10 samples to test the trained model; (3) Model 3: 500 samples, including 30 samples to test the trained model.

Table 1 displays the prediction results of trained GCN, trained CNN, and trained DNN among different samples. It can be observed that the trained GCN achieves the best performance among others. Compared to the trained CNN, the Acc difference is nearly 20%, while the Pre difference is more than 10%. Compared to the trained DNN, in the three trained models, the Acc difference is more than 20%, while the Pre difference is as high as 40%. Moreover, the trained GCN model can obtain higher accuracy with fewer samples. That is to say, considering power grid typology in the learning process can extract more information with fewer samples.

Synthesize the performance of the Acc and Pre indices, Model 2 has been selected and constructed a confusion matrix shown in Fig. 3 for a detail describe the formulation of Acc and Pre indices. Since each test sample contains 144 binary variables, for 10 test samples, the actual and predicted units' status of 10×144 binary variables are observed to calculate the indices, as shown in Fig. 3. The dark blue block in the second row of Fig. 3. shows that 88.24% of units' states are correctly predicted to be on. In the first row, the dark blue block depicts that 83.01% of units' states are correctly predicted to be off.

3.2.2. Cost and computation time-saving comparison between MILP-SCUC and GCN-SCUC

Based on the prediction result of GCN in Model 2, the total cost of GCN-SCUC and MILP-SCUC are compared and are shown in Fig. 4. The performance of GCN-SCUC is investigated based on the gap from the optimal solution obtained by the MILP-SCUC method in Fig. 4. The smaller the cost gap is, the more accurate the resolution of the GCN-SCUC will be. According to Fig. 4, the cost of testing samples by GCN-SCUC is generally higher than those of MILP-SCUC. Compared with MILP-SCUC, the percentage optimality gap is relatively small. The maximization gap is 8.91%, while the minimization gap is 0.91%. The dashed line with the red arrow shows the average gap is only 3.42% for the MILP-SCUC and GCN-SCUC methods. That is to say, the solution of the proposed GCN-SCUC method is very close to the global optimal solution. Thereby, the result of GCN can be interpreted as one of the multi-solution of the SCUC problem.

Table 2
The computation time of GCN-SCUC in case 30.

Item		Avg(s)	
GCN-SCUC	GCN	0.03	0.92
	Convex problem	0.89	
MILP-SCUC		15.78	
Time-saving		94.17%	

Fig. 5 shows the computation time of MILP-SCUC and GCN-SCUC for each testing sample. Moreover, the percentage of time-saving is also displayed in Fig. 5, and the dashed line with the red arrow shows the average time-saving rate. Note that the computation time of GCN-SCUC refers to the sum between GCN prediction time and the convex optimization problem calculating time. Table 2 shows the average time of 10 testing samples. The average time of MILP-SCUC is 15.78 s, and the average time of GCN-SCUC is 0.92 s. Compared with MILP-SCUC, the computational efficiency of the GCN-SCUC method achieves speedups of about 17x. Therefore, the time-saving by up to 94.17%. This is because, reducing the equivalent number of binary variables benefits SCUC more since this directly decreases the higher number of constraints compared with MILP-SCUC.

3.2.3. Unit states and voltage values displayed

Table 3 shows the unit states results of GCN-SCUC and MILP-SCUC on one day, while the voltage magnitude of the GCN-SCUC method is displayed in Fig. 6. The numbers in Table 3 represent the state of the unit at that time is turned on. We found that the unit decisions predicted by GCN are one of the multi-solution of the SCUC problem, and the solution is very close to the global optimal solution.

3.3. Case 118

To validate the proposed algorithm, a relatively large security-constrained unit commitment problem of 54 units is tested. 270 samples are used to train the model, 30 samples are selected as testing samples. The system contains 1296 binary variables predicted by GCN. With respect to case 118, the Acc performs better since the increasing number of units shows more regularity. The results for computation time and the operation cost gap are also displayed in Table 3. They achieve speedups of between 13.5x and 14x, an average optimality error of 1.25% for the case 118 system. This indicates that the proposed algorithm is scalable for a large-scale power system of the SCUC problem (see Table 4).

4. Conclusion and future work

In this work, the SCUC model with AC power flow constraints is formulated as a MILP problem. The proposed learning-based

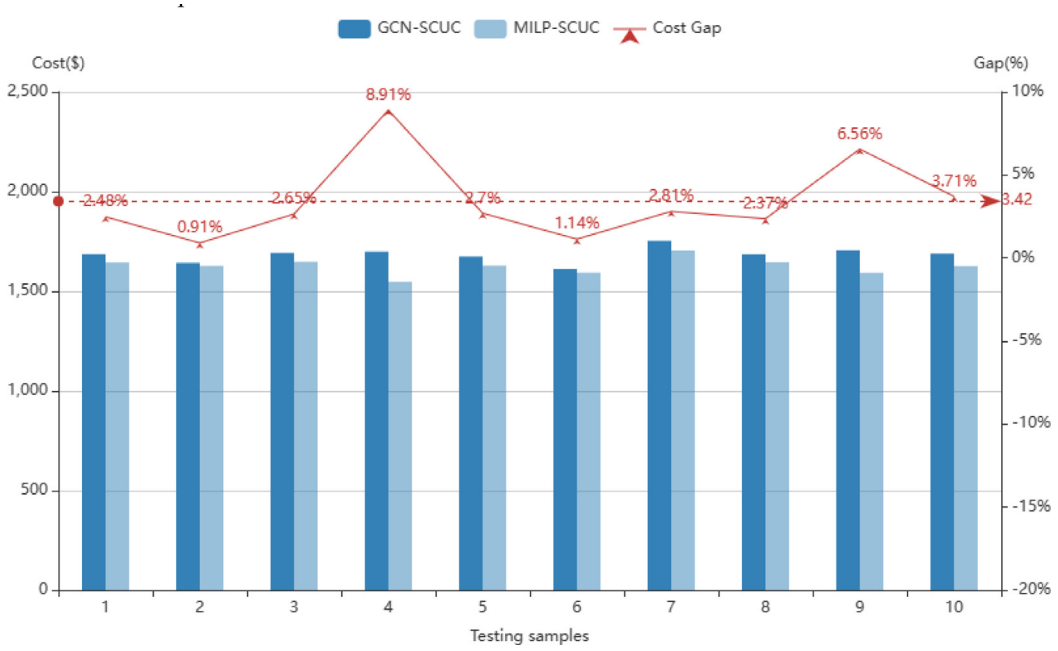

Fig. 4. The total cost and cost gap between GCN-SCUC and MILP-SCUC.

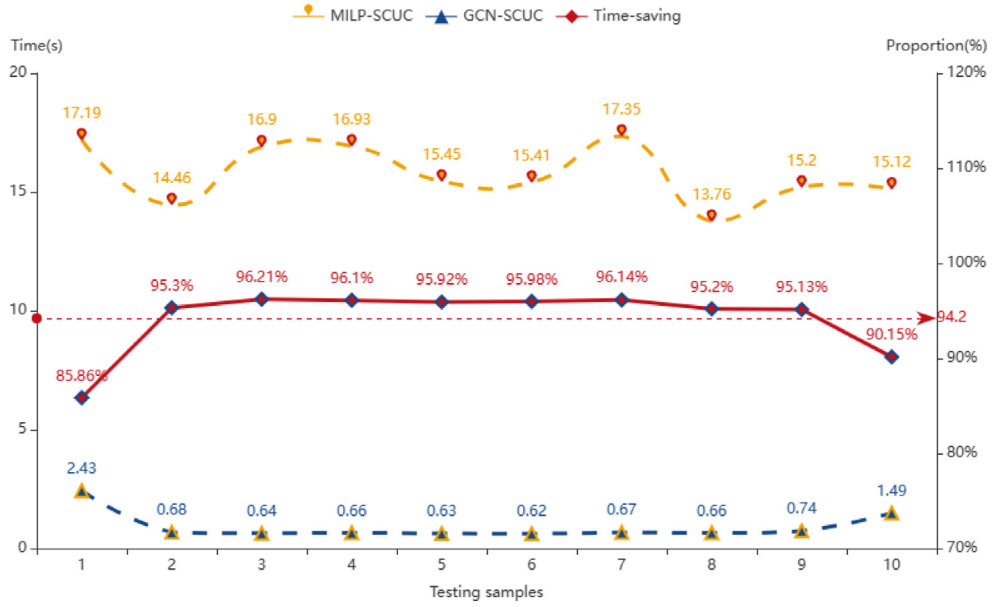

Fig. 5. The computation time of MILP-SCUC and GCN-SCUC.

Table 3
The unit states of GCN-SCUC and MILP-SCUC in case 30 (one day).

Method Unit state	GCN-SCUC Unit-on	MILP-SCUC Unit-on
Unit 1	/	2, 10
Unit 2	2–3, 5, 8, 11–12, 14, 17–18, 20–21, 23	4–5, 7–10, 14, 16, 19–20, 23
Unit 3	1–24	1–24
Unit 4	1–24	1–24
Unit 5	1–24	1–3, 5–6, 8–9, 11–15, 17–18, 20–24
Unit 6	/	/

GCN-SCUC method is utilized to solve the SCUC problem. First, we present the application of GCN to obtain the unit on/off status. With the generator on/off status results, the continuous variables corresponding to the voltage magnitudes and generator power outputs can be solved by a convex optimization problem. The results validate that GCN, which takes system topology into account, can achieve high accuracy with fewer samples compared with other learning methods. Additionally, the proposed GCN-SCUC method achieves near-optimal SCUC solutions with a mean deviation from the optimal solution of 3.42%. Moreover, for case

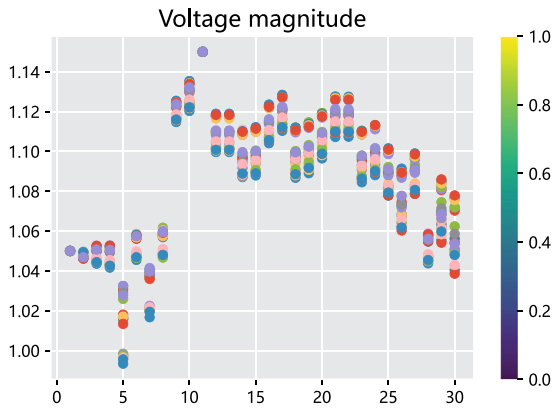

Fig. 6. Voltage magnitude of GCN-SCUC method in case 30 (one day).

Table 4
Performance of GCN-SCUC in case 118.

Method	MILP-SCUC	GCN-SCUC
Avg(s)	39.39	2.82
Time-saving	92.84%	
Cost gap	1.25%	

30 with 6 units, the computation time is significantly reduced from 15.78 s to 0.92 s. While in case 118 with 54 units, the computation time can be decreased from 39.39 s to 2.82 s.

However, some limits to this first investigation are left for future research. The topology of power systems is invariant, which means the fixed adjacency matrix. In practice, the features of nodes and adjacency matrix change over time. Moreover, there is still room for improving the accuracy of prediction. Further investigation regarding architecture improvements of GCN or alternative variants of GNN approaches may be considered; meanwhile, the new method can adapt to network changes and uncertainty on the generation side.

CRediT authorship contribution statement

Xian Tang: Conceptualization, Methodology, Software, Investigation, Formal analysis, Writing – original draft. **Xiaoqing Bai:** Conceptualization, Funding acquisition, Resources, Supervision, Writing – review & editing. **Zonglong Weng:** Software, Visualization, Writing – review & editing. **Rui Wang:** Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the National Natural Science Foundation of China [grant numbers 51967001].

References

Bai, X., Wei, H., 2009. Semi-definite programming-based method for security-constrained unit commitment with operational and optimal power flow constraints. IET Gener. Transm. Distrib. 3 (2), 182–197.

Chen, Y., Wang, F., Ma, Y., Yao, Y., 2020. A distributed framework for solving and benchmarking security constrained unit commitment with warm start. IEEE Trans. Power Syst. 35 (1), 711–720. http://dx.doi.org/10.1109/TPWRS.2019.2930706.

Donon, Balthazar, et al., 2020. Neural networks for power flow: Graph neural solver. Electr. Power Syst. Res. 189, 106547. http://dx.doi.org/10.1016/j.epsr.2020.106547.

Du, E., Zhang, N., Kang, C., Xia, Q., 2019. A high-efficiency network-constrained clustered unit commitment model for power system planning studies. IEEE Trans. Power Syst. 34 (4), 2498–2508. http://dx.doi.org/10.1109/TPWRS.2018.2881512.

Fu, Y., Li, Z., Wu, L., 2013. Modeling and solution of the large-scale security-constrained unit commitment. IEEE Trans. Power Syst. 28 (4), 3524–3533. http://dx.doi.org/10.1109/TPWRS.2013.2272518.

Gurobi Optimization, 2021. LLC. Gurobi Optimizer Reference Manual.

Hasan, F., Kargarian, A., Mohammadi, J., 2021. Hybrid learning aided inactive constraints filtering algorithm to enhance AC OPF solution time. IEEE Trans. Ind. A 57 (2), 1325–1334.

Khodayar, M., Wang, J., 2019. Spatio-temporal graph deep neural network for short-term wind speed forecasting. IEEE Trans. Sustain. Energy 10 (2), 670–681. http://dx.doi.org/10.1109/TSTE.2018.2844102.

Li, X., Zhai, Q., Zhou, J., Guan, X., 2020. A variable reduction method for large-scale unit commitment. IEEE Trans. Power Syst. 35 (1), 261–272. http://dx.doi.org/10.1109/TPWRS.2019.2930571.

Liao, W., Bak-Jensen, B., Pillai, J.R., et al., 2022. A review of graph neural networks and their applications in power systems. J. Mod. Power Syst. Clean Energy 10 (2), 345–360. http://dx.doi.org/10.35833/MPCE.2021.000058.

Liao, W., Yang, D., Wang, Y., Ren, X., 2021. Fault diagnosis of power transformers using graph convolutional network. CSEE J. Power Energy Syst. 7 (2), 241–249. http://dx.doi.org/10.17775/CSEEJPES.2020.04120.

Liao, W., Yun, Y., Wang, Y., et al., 2020. Reactive power optimization of distribution network based on graph convolutional network. Power Syst. Technol. (99), 1–12. http://dx.doi.org/10.13335/j.1000-3673.pst.2020.1593.

Muralikrishnan, N., Jebaraj, L., Rajan, C.C.A., 2020. A comprehensive review on evolutionary optimization techniques applied for unit commitment problem. IEEE Access 8, 132980-133014. http://dx.doi.org/10.1109/ACCESS.2020.3010275.

Nair, Vinod, et al., 2020. Solving mixed integer programs using neural networks. arXiv, abs/2012.13349.

Pineda, S., Morales, J.M., 2022. Is learning for the unit commitment problem a lowhanging fruit? Electr. Power Syst. Res. 207, 107851. http://dx.doi.org/10.1016/j.epsr.2022.107851.

Pineda, S., Morales, J.M., Jiménez-Cordero, A., 2020. Data-driven screening of network constraints for unit commitment. IEEE Trans. Power Syst. 35 (5), 3695–3705. http://dx.doi.org/10.1109/TPWRS.2020.2980212.

Qin, Liwen, Yu, Xiaoyong, Gui, Haitao, et al., 2022. Super resolution distribution network measurement considering distribution network topology reconstruction. Energy Rep. 8 (4), 313–320. http://dx.doi.org/10.1016/j.egyr.2022.01.129.

Ruan, G., Zhong, H., Zhang, G., et al., 2021. Review of learning-assisted power system optimization. CSEE J. Power Energy Syst. 7 (2), 221–231. http://dx.doi.org/10.17775/CSEEJPES.2020.03070.

Safdarian, F., Mohammadi, A., Kargarian, A., 2020. Temporal decomposition for security-constrained unit commitment. IEEE Trans. Power Syst. 35 (3), 1834–1845. http://dx.doi.org/10.1109/TPWRS.2019.2947410.

Sakkas, Nikolaos P., Abang, Roger, 2022. Thermal load prediction of communal district heating systems by applying data-driven machine learning methods. Energy Rep. 8, 1883–1895. http://dx.doi.org/10.1016/j.egyr.2021.12.082.

Wu, T., Angela Zhang, Y.-J., Wang, S., 2022a. Deep learning to optimize: Security-constrained unit commitment with uncertain wind power generation and BESSs. IEEE Trans. Sustain. Energy 13 (1), 231–240. http://dx.doi.org/10.1109/TSTE.2021.3107848.

Wu, Zeqing, Mu, Yunfei, Deng, Shuai, Li, Yang, 2022b. Spatial–temporal short-term load forecasting framework via K-shape time series clustering method and graph convolutional networks. Energy Rep. 8, 8752–8766. http://dx.doi.org/10.1016/j.egyr.2022.06.122.

Xavier, Álinson S., et al., 2020. Learning to solve large-scale security-constrained unit commitment problems. INFORMS J. Comput. 33 (2), 739–756. http://dx.doi.org/10.1287/ijoc.2020.0976.

Yang, J., Zhang, N., Kang, C., Xia, Q., 2017. A state-independent linear power flow model with accurate estimation of voltage magnitude. IEEE Trans. Power Syst. 32 (5), 3607–3617. http://dx.doi.org/10.1109/TPWRS.2016.2638923.

Yang, Nan, et al., 2019. Research on data-driven intelligent security-constrained unit commitment dispatching method with selflearning ability. Proc. Chinese Soc. Electr. Eng. 39 (20), 2934–2946. http://dx.doi.org/10.13334/j.0258-8013.pcsee.180504.

Yang, Yafei, et al., 2020. Integrated datadriven framework for fast SCUC calculation. IET Gener. Transm. Distrib. 14 (24), 5728–5738. http://dx.doi.org/10.1049/ietgtd.2020.0823.

Yang, N., et al., 2022a. A comprehensive review of security-constrained unit commitment. J. Mod. Power Syst. Clean Energy 10 (3), 562–576. http://dx.doi.org/10.35833/MPCE.2021.000255.

Yang, N., et al., 2022b. Intelligent data-driven decision-making method for dynamic multisequence: An E-Seq2Seq-based SCUC expert system. IEEE Trans. Industr. Inform. 18 (5), 3126–3137. http://dx.doi.org/10.1109/TII.2021.3107406.

Zhai, Q., Guan, X., Cheng, J., Wu, H., 2010. Fast identification of inactive security constraints in SCUC problems. IEEE Trans. Power Syst. 25 (4), 1946–1954. http://dx.doi.org/10.1109/TPWRS.2010.2045161.

Zhu, R., Wei, H., Bai, X., 2019. Wasserstein metric based distributionally robust approximate framework for unit commitment. IEEE Trans. Power Syst. 34 (4), 2991–3001. http://dx.doi.org/10.1109/TPWRS.2019.2893296.

Zimmerman, R.D., Murillo-Sanchez, C.E., 2020. MATPOWER (Version 7.1). http://dx.doi.org/10.5281/zenodo.4074135, [Software]. Available: https://matpower.org.

Zimmerman, R.D., Murillo-Sánchez, C.E., Thomas, R.J., 2011. MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education. IEEE Trans. Power Syst. 26 (1), 12–19. http://dx.doi.org/10.1109/TPWRS.2010.2051168.