Graph Neural Network Based Column Generation for Energy Management in Networked Microgrid

Yuchong Huo, *Member*, *IEEE*, Zaiyu Chen, *Member*, *IEEE*, Qun Li, *Member*, *IEEE*, Qiang Li, *Member*, *IEEE*, and Minghui Yin, *Member*, *IEEE*

***Abstract*—In this paper, we apply a model predictive control based scheme to the energy management of networked microgrid, which is reformulated based on column generation. Although column generation is effective in alleviating the computational intractability of large-scale optimization problems, it still suffers from slow convergence issues, which hinders the direct real-time online implementation. To this end, we propose a graph neural network based framework to accelerate the convergence of the column generation model. The acceleration is achieved by selecting promising columns according to certain stabilization method of the dual variables that can be customized according to the characteristics of the microgrid. Moreover, a rigorous energy management method based on the graph neural network accelerated column generation model is developed, which is able to guarantee the optimality and feasibility of the dispatch results. The computational efficiency of the method is also very high, which is promising for real-time implementation. We conduct case studies to demonstrate the effectiveness of the proposed energy management method.**

***Index Terms*—Column generation, energy management, graph neural network, machine learning, microgrid.**

I. Introduction

MICROGRID (MG) is an important part of the modern smart grid, where significant level of renewable energy sources can be accommodated. To deal with the intermittent nature of renewable generation, model predictive control (MPC) based methods have been extensively studied in recent years for energy management in networked MG (denoted as MG energy management in this paper). For example, [1] addressed the intra-hour economic dispatch problem of MG with a closed-loop distributed MPC, which was able to reduce the variations of the scheduled generation caused by fluctuating renewable power. Reference [2] developed an MPC based framework for energy scheduling in an MG by considering the temporal and spatial correlations between multi-site renewable generations. A dual-mode distributed economic MPC for management of distributed energy resources (DERs) in an MG was proposed in [3], where different DERs worked iteratively and cooperatively to solve the economic objective functions. Reference [4] developed a hybrid economic MPC method using weather forecasts for an isolated MG which allowed the automatic grid-connection to provide ancillary services to the main grid.

The MPC scheme for energy management is able to ensure the optimality of the dispatch schedules without violating the operating constraints of the networked MG [5], [6], thanks to its foresight and self-correcting capabilities. However, MPC requires the online solution to a mixed-integer programming problem at each dispatch step, making it computationally expensive and hindering its real-time implementation in networked MGs, especially considering the complexity of the energy management problem.

In recent years, a large research effort has been focused on alleviating the computational intractability of MPC scheme for energy management with decomposition methods, and the column generation (CG) method has attracted particular attention. Compared with other decomposition methods, CG has two advantageous features. First, it is able to fully leverage the primal block-angular structure of the MG energy management and reduce computational cost [7]. Second, it provides a convenient framework for recovering integer variables. Reference [7] developed a CG based framework that allowed the MG operator to integrate any type of resources whose operation could be formulated within a mixed-integer linear programming (MILP). A Dantzig-Wolfe decomposition and parallel asynchronous CG method was proposed in [8] to solve a multi-stage stochastic planning of an integrated power and natural gas system in an MG. Reference [9] designed a CG based distributed computing framework for multi-stage stochastic planning of MGs. An iterative CG method was proposed in [10] for MG energy management, which successfully reduced the size of the problem.

While the CG method systematically reduces the computational complexity of MPC scheme for energy management in MGs by utilizing structural features, it is known to suffer

Manuscript received: June 6, 2023; revised: October 24, 2023; accepted: February 8, 2024. Date of CrossCheck: February 8, 2024. Date of online publication: March 20, 2024.

This work was supported in part by the National Science Foundation of China (No. 51977111), the Jiangsu Qinglan Project, and the State Grid Corporation Science and Technology Project "Key technologies of active frequency support for mid and long distance offshore wind farm with multiple grid-forming converter connected via VSC-HVDC" (No. 5108-202218280A-2-241-XG).

Y. Huo, Z. Chen (corresponding author), and M. Yin are with the School of Automation, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: yuchong. huo@mail. mcgill. ca; Zaiyu. Chen@njust. edu. cn; ymhui@vip.163.com).

Q. Li and Q. Li are with the State Grid Jiangsu Electric Power Co., Ltd. Research Institute, Nanjing 211103, China (e-mail: qun_li@sina.com; 35830342@qq.com).

DOI: 10.35833/MPCE.2023.000385

from slow convergence difficulties [11], [12], which creates major obstacles for its real-time implementation.

Recently, machine learning (ML) technique has emerged as a promising tool for accelerating the convergence and reducing the computational complexity of traditional optimization algorithms, which is usually referred to as learning-to-optimize [13], [14]. In MG energy management, the same optimization problem is always solved repeatedly, differing only in the input parameters. The fundamental concept of learning-to-optimize involves training an ML model to quickly approximate certain resource-intensive computations within MG energy management, aiming to enhance the overall computational efficiency. The ML model is also known as a "proxy model" in literature. For instance, in [15] and [16], model-free deep reinforcement learning (RL) was employed to create surrogate models for energy management in networked MGs. Similarly, the deep neural network (DNN) was used in online energy management to provide computational acceleration in a single building or house in [17] and [18].

Although ML techniques are excellent at reducing the computational complexity and improving the real-time performance of traditional optimization algorithms, they cannot assure the feasibility of direct outputs considering the constraints of the original optimization problem [13], [14]. The fundamental reason for this phenomenon is that the prevailing ML models (especially DNN) are data-driven and "black-box" in nature, where the well-established physical models of MG energy management are completely absent. Thus, none of the ML models presented in the research works mentioned above have provided a rigorous solution to MG energy management.

In fact, it can be found that the properties of CG and ML are clearly complementary. Specifically, CG is a model-driven method, which utilizes the structural feature of the MG energy management problem and guarantees the feasibility and optimality of its solution. Besides, ML is good at identifying the optimal mapping of an optimization problem, which achieves a significant reduction in computational complexity. Therefore, many researchers are looking into ML based acceleration methods for the traditional CG method. Moreover, the learning task dedicated to CG has the following unique requirements.

1) In CG, the number of selected columns will be different from one iteration to another, and the ML model should be able to generalize to problems with different sizes.

2) The ML model should have the same predictions regardless of the permutation of the columns (permutation invariance).

In existing literature, graph neural network (GNN) [19], [20] is always adopted when designing acceleration frameworks for CG because it is able to handle the above requirements perfectly. For example, [12] and [21] presented a GNN based method to accelerate CG, where the GNN model was applied to select the most promising columns generated to reduce the computing time spent at each iteration of CG. Reference [22] developed a GNN based neural CG architecture that iteratively selected columns to be added to the problem. This architecture was inspired by stabilization techniques of CG where dual information was utilized. A GNN and RL based acceleration method for CG was proposed in [23], where CG was treated as a sequential decision-making problem. However, it is worth mentioning that the GNN based acceleration methods above are all dedicated to certain classes of optimization problems, e. g., cutting stock problem, crew scheduling problem, vehicle routing problem, which is difficult to guarantee that such methods can work in different networked MGs under different operating conditions.

Inspired by previous work, in this paper, the powers of MPC, CG, and GNN are combined to design a computationally efficient and rigorous framework for energy management of networked MG. The contributions of this paper are as follows.

1) We formulate the energy management of net-worked MG based on the MPC scheme, which is in the form of an MILP problem. We create the CG model for the MPC based formulation, comprising a restricted master problem (RMP) and multiple subproblems (SPs).

2) We propose a GNN based framework to accelerate convergence and improve the computational efficiency of the CG model. The state of the RMP is encoded using GNN which operates on a bipartite graph representation of the column-constraint interaction in the RMP. The message passing mechanism is designed based on graph convolutional framework and neighborhood attention. The GNN model accelerates the convergence of the CG by selecting promising columns provided by different SPs that can smooth and stabilize the dual variables associated with the RMP.

3) An MG energy management method based on the GNN accelerated CG method is proposed. In this method, the GNN model is fused with the CG framework and the physical model of the MG. And the energy management method is rigorous where both the feasibility and the optimality of the output dispatch solutions can be guaranteed. Also, the energy management method is computationally efficient, which is promising for real-time implementation.

4) Compared with the existing GNN based acceleration methods for CG such as [12], [22], the proposed method enables the MG operator to tailor the smoothing and stabilizing methods for CG based on the specific characteristics of the target networked MG, which guarantees that our proposed method is applicable to a wide range of networked MGs with various operating conditions.

II. Modeling of MG Energy Management

A. MPC Scheme for MG Energy Management

In this paper, the MG energy management is formulated based on an MPC scheme. At the beginning of each time step, an optimization problem is solved.

1) Objective function: (1) is to minimize the total operating cost at current time step t_0, i.e., $\tau=0$, while considering the entire forward-looking horizon $\tau \in \Xi_H$, where τ is the index of look ahead time step; and Ξ_H is the the set of receding ho-

rizon time steps.

$$\min\left(\sum_{r_g\in\Xi_g}(\alpha_{r_g}p_g^{r_g}(\tau)+\beta_{r_g}u_g^{r_g}(\tau))+\sum_{r_d\in\Xi_{ir}}\varepsilon^d(\tilde{p}_d^{r_d}(\tau)-p_d^{r_d}(\tau))+\sum_{r_\delta\in\Xi_\delta}\varepsilon^\delta(\tilde{p}_\delta^{r_\delta}(\tau)-p_\delta^{r_\delta}(\tau))\right) \tag{1}$$

where Ξ_g, Ξ_δ, and Ξ_{ir} are the sets of diesel generators, renewable generation, and interruptible loads, respectively; r_g, r_d, and r_δ are the indices of diesel generators, interruptible loads, and renewable generation, respectively; $p_d^{r_d}$ is the active power consumption of r_d; $p_g^{r_g}$ is the active power output of r_g; $u_g^{r_g}$ is the commitment status of r_g; α_{r_g} and β_{r_g} are the consumption rate and constant of r_g, respectively; ε^d and ε^δ are the penalties associated with curtailing loads and renewable generation, respectively; and $p_\delta^{r_\delta}$ is the active power output of r_δ. We make the assumption that there are no operating costs related to energy storage. The solution is the optimal scheduling plan for the DERs to be applied at $\tau\in\Xi_H$ in the forward-looking horizon. However, the scheduling decisions are only implemented for the current time step $\tau=0$. As the next time step begins, this procedure is reiterated over a shifted horizon, considering the updated forecasts and operating conditions of DERs.

2) Model of diesel generator: the operating constraints of $\tau\in\Xi_g$ include the active and reactive power limits, and the minimum up and down time constraints.

$$u_g^{r_g}(\tau)\underline{p}_g^{r_g}\le p_g^{r_g}(\tau)\le u_g^{r_g}(\tau)\overline{p}_g^{r_g} \tag{2}$$

$$u_g^{r_g}(\tau)\underline{q}_g^{r_g}\le q_g^{r_g}(\tau)\le u_g^{r_g}(\tau)\overline{q}_g^{r_g} \tag{3}$$

$$u_g^{r_g}(\tau)-u_g^{r_g}(\tau-1)-v_{r_g}^+(\tau)+v_{r_g}^-(\tau)=0 \tag{4}$$

$$v_{r_g}^+(\tau)-v_{r_g}^-(\tau)\le\zeta(\tau) \tag{5}$$

$$\sum_{t=\max\{1,\tau-MUT_{r_g}+1\}}^{\tau}v_{r_g}^+(t)\le u_g^{r_g}(\tau) \tag{6}$$

$$\sum_{t=\max\{1,\tau-MDT_{r_g}+1\}}^{\tau}v_{r_g}^-(t)\le 1-u_g^{r_g}(\tau) \tag{7}$$

where $q_g^{r_g}$ is the reactive power output of r_g; t is the index of time; $\overline{(\cdot)}$ and $\underline{(\cdot)}$ denote the upper and lower bounds of the variable, respectively; MUT_{r_g} and MDT_{r_g} are the minimum up and down time of r_g, respectively; and $v_{r_g}^+(\tau)$, $v_{r_g}^-(\tau)$, and $\zeta(\tau)$ are the binary auxiliary variables, and $\zeta(\tau)=0$ if the time that the generator remains online or offline is less than its minimum up or down time, while $\zeta(\tau)=1$ is the opposite.

3) Model of energy storage: the constraints of energy storage $r_e\in\Xi_e$ include its technical limits of power capacity and energy holding, where r_e is the index of energy storage; and Ξ_e is the set of energy storage.

$$\underline{p}_e^{r_e}\le p_e^{r_e}(\tau)\le\overline{p}_e^{r_e} \tag{8}$$

$$\underline{q}_e^{r_e}\le q_e^{r_e}(\tau)\le\overline{q}_e^{r_e} \tag{9}$$

$$\underline{s}_e^{r_e}\le s_e^{r_e}(t_0-1)-T_\Delta\sum_{t=0}^{\tau}p_e^{r_e}(t)\eta^{r_e}\le\overline{s}_e^{r_e} \tag{10}$$

where η^{r_e} is the efficiency of r_e, $\eta^{r_e}=\eta_c^{r_e}$ inducates charging and $\eta^{r_e}=1/\eta_d^{r_e}$ indicates discharging, and $\eta_c^{r_e}$ and $\eta_d^{r_e}$ are the charging and discharging efficiencies, respectively; $p_e^{r_e}$ and $q_e^{r_e}$ are the active and reactive power outputs of energy storage, respectively; T_Δ is the length of each time step; and $s_e^{r_e}$ is the energy level of r_e.

4) Model of renewable generation and loads: the power output of renewable generation and the power consumption of loads in the forward-looking horizon need to be predicted. To predict the renewable generation and load demand, the modified vector autoregressive (VAR) model is applied [24]. We consider two types of loads in the MG: critical load and interruptible load. According to [25] and [26], critical loads must be supplied under all normal operating conditions of the MG. If needed, interruptible loads may be curtailed, but every effort should be made to fulfill them.

$$\begin{cases}0\le p_\delta^{r_\delta}(\tau)\le\tilde{p}_\delta^{r_\delta}(\tau)\\0\le q_\delta^{r_\delta}(\tau)\le\tilde{q}_\delta^{r_\delta}(\tau)\end{cases}\quad r_\delta\in\Xi_\delta \tag{11}$$

$$\begin{cases}p_d^{r_d}(\tau)=\tilde{p}_d^{r_d}(\tau)\\q_d^{r_d}(\tau)=\tilde{q}_d^{r_d}(\tau)\end{cases}\quad r_d\in\Xi_{cr} \tag{12}$$

$$\begin{cases}0\le p_d^{r_d}(\tau)\le\tilde{p}_d^{r_d}(\tau)\\0\le q_d^{r_d}(\tau)\le\tilde{q}_d^{r_d}(\tau)\end{cases}\quad r_d\in\Xi_{ir} \tag{13}$$

where Ξ_{cr} is the set of critical loads; $q_\delta^{r_\delta}$ is the reactive power output of r_δ; and $q_d^{r_d}$ is the reactive power consumption of r_d.

5) Power balance: the interaction between the MG and the distribution network is beyond the scope of this paper. As a result, the active and reactive power exchanges at the point of interconnection (POI) $p_{poi}(\tau)$ and $q_{poi}(\tau)$ are taken as parameters. When the MG is in grid-connected mode, $p_{poi}(\tau)$ and $q_{poi}(\tau)$ are set to the corresponding values in the power exchange schedule. However, when the MG is in islanded mode, $p_{poi}(\tau)$ and $q_{poi}(\tau)$ are both set to be 0.

$$\sum_{r_g\in\Xi_g}p_g^{r_g}(\tau)+\sum_{r_e\in\Xi_e}p_e^{r_e}(\tau)+\sum_{r_\delta\in\Xi_\delta}p_\delta^{r_\delta}(\tau)+p_{poi}(\tau)=\sum_{r_d\in\Xi_{cr}\cup\Xi_{ir}}p_d^{r_d}(\tau) \tag{14}$$

$$\sum_{r_g\in\Xi_g}q_g^{r_g}(\tau)+\sum_{r_e\in\Xi_e}q_e^{r_e}(\tau)+\sum_{r_\delta\in\Xi_\delta}q_\delta^{r_\delta}(\tau)+q_{poi}(\tau)=\sum_{r_d\in\Xi_{cr}\cup\Xi_{ir}}q_d^{r_d}(\tau) \tag{15}$$

6) Model of MG network: an MG typically operates as radial. The power flows in the networked MG are modeled using linearized DistFlow equations [27], [28].

$$P_{l+1}(\tau)=P_l(\tau)-\sum_{r_g,r_e,r_\delta,r_d\in\mathcal{N}(n)}(p_g^{r_g}(\tau)+p_e^{r_e}(\tau)+p_\delta^{r_\delta}(\tau)-p_d^{r_d}(\tau)) \tag{16}$$

$$Q_{l+1}(\tau)=Q_l(\tau)-\sum_{r_g,r_e,r_\delta,r_d\in\mathcal{N}(n)}(q_g^{r_g}(\tau)+q_e^{r_e}(\tau)+q_\delta^{r_\delta}(\tau)-q_d^{r_d}(\tau)) \tag{17}$$

$$V_{n+1}(\tau)=V_n(\tau)-2(\rho_lP_l(\tau)+\xi_lQ_l(\tau)) \tag{18}$$

where P_l and Q_l are the active and reactive power flows on branch l, respectively; ρ_l and ξ_l are the resistance and reac-

tance of branch l, respectively; V_n is the voltage magnitude at bus n; and $\mathcal{N}(\cdot)$ is employed to associate the position of DER or load with the relevant node of the network.

It is worth noting that the physical network constraints of an MG or distribution system with radial structure are mainly concerned with node voltage constraints [29], which are presented as:

$$\underline{V}_n \le V_n(\tau) \le \bar{V}_n \tag{19}$$

B. CG Framework

For the sake of simplicity, we reformulate the complete model (1)-(19) into its compact form.

$$\min_{\boldsymbol{x}} \sum_{r \in \Xi_g \cup \Xi_{ir} \cup \Xi_\delta} \sum_{\tau \in \Xi_H} \boldsymbol{c}_r^{\mathrm{T}} \boldsymbol{x}_{r,\tau} \tag{20}$$

s.t.

$$\sum_{r \in \Xi_g \cup \Xi_{ir} \cup \Xi_\delta} \boldsymbol{A}_r \boldsymbol{x}_{r,\tau} \le \boldsymbol{b}_\tau \quad \forall \tau \in \Xi_H \tag{21}$$

$$\boldsymbol{x}_{r,\tau} \in \mathcal{X}_{r,\tau} \quad \forall r \in \Xi_g \cup \Xi_{ir} \cup \Xi_\delta, \tau \in \Xi_H \tag{22}$$

where $\boldsymbol{x}_{r,\tau}$ is the vector of decision variables associated with the resource r at $\tau \in \Xi_H$; $\boldsymbol{c}_r$ is the vector of cost parameters; $\boldsymbol{A}_r$ is the matrix representing the coefficients of $\boldsymbol{x}_{r,\tau}$; $\boldsymbol{b}_\tau$ is the vector representing the right-hand side values of the constraint inequalities; and $\mathcal{X}_{r,\tau}$ is the feasible region of $\boldsymbol{x}_{r,\tau}$ ($\tau \in \Xi_H$), defined by the operating constraints of the specific DER or load. Formula (21) represents the power balance (14) and (15) and the power flow constraints (16)-(19) of the networked MG, i.e., the coupling constraints. CG transforms the energy management problem (1)-(19) into an RMP along with multiple SPs. The RMP ensures that the coupling constraints (21) are satisfied.

$$\min_{\lambda} \sum_{\omega \in \hat{\Omega}_r} \sum_{r \in \Xi_g \cup \Xi_{ir} \cup \Xi_\delta} \sum_{\tau \in \Xi_H} \lambda_{\omega,r} \boldsymbol{c}_r^{\mathrm{T}} \hat{\boldsymbol{x}}_{\omega,r,\tau} \tag{23}$$

s.t.

$$\sum_{\omega \in \hat{\Omega}_r} \sum_{r \in \Xi_g \cup \Xi_{ir} \cup \Xi_\delta} \lambda_{\omega,r} \boldsymbol{A}_r \hat{\boldsymbol{x}}_{\omega,r,\tau} \le \boldsymbol{b}_\tau \quad \forall \tau \in \Xi_H \tag{24}$$

$$\sum_{\omega \in \hat{\Omega}_r} \lambda_{\omega,r} = 1 \quad \forall r \in \Xi_g \cup \Xi_{ir} \cup \Xi_\delta \tag{25}$$

$$\lambda_{\omega,r} \ge 0 \quad \forall r \in \Xi_g \cup \Xi_{ir} \cup \Xi_\delta, \omega \in \hat{\Omega}_r \tag{26}$$

where Ω_r is the index set of extreme vertices (columns) of $conv(\bigcup_{\tau \in \Xi_H} \mathcal{X}_{r,\tau})$, $\hat{\Omega}_r \subset \Omega_r$ is the subset of columns that are currently considered, and $conv(\cdot)$ is the convex hull; $\hat{\boldsymbol{x}}_{\omega,r,\tau}$ is the ω^{th} column of the DER or load r; and $\lambda_{\omega,r}$ is the associated primal variable. The vectors of dual variables linked with (24) and (25) are denoted as $\boldsymbol{\pi}_\tau$ and $\boldsymbol{\sigma}_r$, respectively. An SP corresponds to the individual DER or load r ($\forall r \in \Xi_g \cup \Xi_{ir} \cup \Xi_\delta$) is:

$$\hat{\boldsymbol{x}}_{\omega^*,r,\tau} = \arg\min_{\boldsymbol{x}_{r,\tau} \in \mathcal{X}_{r,\tau}} \hat{\boldsymbol{c}}_r = \sum_{\tau \in \Xi_H} (\boldsymbol{c}_r^{\mathrm{T}} - \boldsymbol{\pi}_\tau^{\mathrm{T}} \boldsymbol{A}_r) \boldsymbol{x}_{r,\tau} - \boldsymbol{\sigma}_r \tag{27}$$

where $\hat{\boldsymbol{c}}_r$ is the reduced cost for adding a new column to the RMP.

The CG method iteratively addresses the energy management problem. In each iteration, the RMP is initially solved to optimality, leading to the computation of the dual variables $\boldsymbol{\pi}_\tau$ and $\boldsymbol{\sigma}_r$. Next, we consider adding new columns to the RMP, which are obtained by solving each SP. For the DER or load r, the reduced cost for adding a new column to the RMP is $\hat{\boldsymbol{c}}_r$. Therefore, the SP (27) gives the column $\hat{\boldsymbol{x}}_{\omega^*,r,\tau}$ with the minimal reduced cost. The new columns of all the SPs are added to the RMP and in the next iteration, and the process will be repeated until the reduced costs of all SPs turn nonnegative. As can be observed, the SPs are MILP problems, while the RMP (23)-(26) is a relaxation of the original energy management (20)-(22), which is a linear programming (LP) problem. It is worth noting that the optimal value of $\lambda_{\omega,r}$ is always fractional. To obtain integer feasible solutions, the branch-and-price method is used, where CG is embedded at each node of a branch-and-bound tree. Further details may be found in [7].

III. Methodology

To tackle the convergence challenge of the conventional CG method, this section introduces a column selection algorithm based on GNN. At a high level, the workflow of the proposed algorithm is illustrated in Fig. 1. At each iteration, the proposed algorithm selects promising columns provided by different SPs and adds them to the RMP. In this section, we first introduce the key components and salient features of the proposed algorithm. Subsequently, the paper delves into the specifics of the proposed algorithm and its application in the MG energy management method.

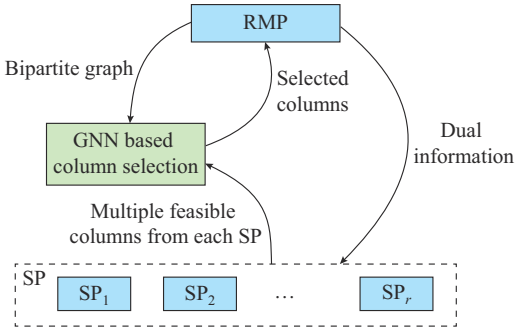

Fig. 1. Schematic illustration of proposed algorithm.

A. GNN Model

1) Bipartite Graph Representation for RMP

The key idea of the proposed algorithm is to use a bipartite graph to encode an instance of the RMP (23)-(26), as shown in Fig. 2(a). It can be observed from Fig. 2 that there are three types of vertices in the bipartite graph, namely column vertex, constraint vertex, and objective function vertex. A column vertex corresponds to a column $\hat{\boldsymbol{x}}_{\omega,r,\tau}$ in the RMP (23)-(26). A constraint vertex corresponds to a constraint in (24). The objective function vertex corresponds to the objective function (23). Please note that the number of column vertices in the bipartite graph equals the number of columns in the RMP in current iteration. And Fig. 2(a) only shows four of them.

At the same time, there exists an edge between the objective function vertex and a column vertex only when the coef-

ficient of the column in the objective function (23) is non-zero. Similarly, there exists an edge between a constraint vertex and a column vertex only when the coefficient of the column in that constraint is non-zero. The features and labels for the vertices and edges of the bipartite graph considered in this paper are described in Appendix A.

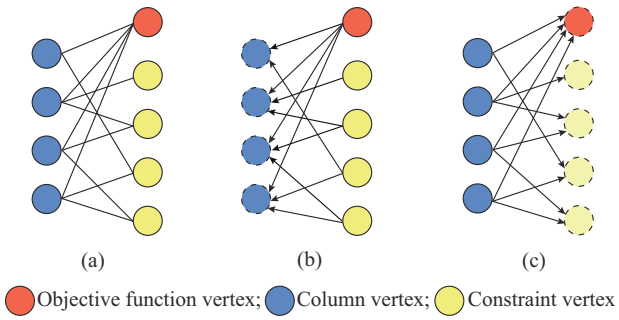

Fig. 2. Bipartite graph representation for RMP. (a) Bipartite graph with three types of vertices. (b) Update of variable vertices. (c) Update of constraint vertices and objective function vertex.

2) Message Passing Mechanism

For the bipartite graph, we design the corresponding message passing mechanism, which is the cornerstone of the proposed algorithm. Detailed steps of the message passing mechanism are described in Algorithm 1.

Algorithm 1: message passing mechanism for bipartite graph based on graph convolutional framework and neighborhood attention

1: $\boldsymbol{h}_i^{(0)}=EMBED(\boldsymbol{\theta}_i)$, $\forall i \in \Xi_V \cup \Xi_C \cup \Xi_O$

2: **for** $k = 1$ to K **do**

3: $\boldsymbol{a}_i^{(k)}=\phi_V^{(k)}\left(\boldsymbol{h}_i^{(k-1)}\middle\| \sum_{i' \in \mathcal{F}(i)} \mu_{ii'}^{(k)} \boldsymbol{h}_{i'}^{(k-1)}\right)$, $\boldsymbol{h}_i^{(k)}=\psi_V^{(k)}(\boldsymbol{h}_i^{(k-1)}\|\boldsymbol{a}_i^{(k)})$ $\quad \forall i \in \Xi_V$

4: $\boldsymbol{a}_i^{(k)}=\phi_C^{(k)}\left(\boldsymbol{h}_i^{(k-1)}\middle\| \sum_{i' \in \mathcal{F}(i)} \mu_{ii'}^{(k)} \boldsymbol{h}_{i'}^{(k-1)}\right)$, $\boldsymbol{h}_i^{(k)}=\psi_C^{(k)}\left(\boldsymbol{h}_i^{(k-1)}\| \boldsymbol{a}_i^{(k)}\right)$ $\quad \forall i \in \Xi_C$

5: $\boldsymbol{a}_i^{(k)}=\phi_O^{(k)}\left(\boldsymbol{h}_i^{(k-1)}\middle\| \sum_{i' \in \mathcal{F}(i)} \mu_{ii'}^{(k)} \boldsymbol{h}_{i'}^{(k-1)}\right)$, $\boldsymbol{h}_i^{(k)}=\psi_O^{(k)}\left(\boldsymbol{h}_i^{(k-1)}\| \boldsymbol{a}_i^{(k)}\right)$ $\quad \forall i \in \Xi_O$

6: **end for**

7: $\tilde{\pi}_i=\chi_C(\boldsymbol{h}_i^{(K)})$ $\quad \forall i \in \Xi_C$

8: $\tilde{o}_i=\chi_O(\boldsymbol{h}_i^{(K)})$ $\quad \forall i \in \Xi_O$

In Algorithm 1, the embeddings of all the vertices are first obtained, where $\boldsymbol{\theta}_i$ is the feature vector of vertex i; $EMBED(\cdot)$ is the embedding function; $\boldsymbol{h}_i$ is the corresponding embedding of vertex i; Ξ_V is the set of column vertices; Ξ_C is the set of constraint vertices; Ξ_O is the set of objective function vertices; $\mu_{ii'}^{(k)}$ is the attention coefficient; $\boldsymbol{a}_i^{(k)}$ represents the aggregated messages for vertex i after the k^{th} iteration of message passing; and the superscripts 0, k, and K indicate the initial embedding, the embedding after the k^{th} iteration of message passing, and the final embedding after K iterations, respectively.

Next, we apply the graph convolutional layers on the bipar-tite graph to update the embeddings of different vertices (*Steps 5-7*), where $\|$ is the concatenation operator; $\psi_V^{(k)}, \phi_V^{(k)}$, $\psi_C^{(k)}, \phi_C^{(k)}, \psi_O^{(k)}$, and $\phi_O^{(k)}$ are the fully-connected feed-forward neural networks with rectified linear unit (ReLU) activation functions; and $\mathcal{F}(\cdot)$ is the neighborhood function.

In *Steps 5-7*, the update process of different vertices consists of two phases. In the first phase, the representations of all the constraint vertices and the objective function vertex are aggregated to update the representations of all the column vertices, as shown in Fig. 2(b). The second phase uses the column representations to update the constraint vertices and the objective function vertex, as shown in Fig. 2(c). In each iteration of the update process, every vertex in the bipartite graph gathers more information from its neighboring vertices, e.g., a column vertex is updated using the information of its neighboring constraint vertices or the objective function vertex and vice versa. However, it is worth noting that although there are no edges between the objective function vertex and constraint vertices, they can still contain the information from each other through the information transmitted from column vertices.

Furthermore, we introduce neighborhood attention into the graph convolutional framework. For the edges between the objective function vertex and column vertices, $\mu_{ii'}^{(k)}$ can be computed as:

$$\mu_{ii'}^{(k)}=\frac{\exp(\varphi_{VO}(\boldsymbol{h}_i^{(k-1)}\|\boldsymbol{h}_{ii'}^{(k-1)}\|\boldsymbol{h}_{i'}^{(k-1)}))}{\sum_{i' \in \mathcal{F}(i)} \exp(\varphi_{VO}(\boldsymbol{h}_i^{(k-1)}\|\boldsymbol{h}_{ii'}^{(k-1)}\|\boldsymbol{h}_{i'}^{(k-1)}))} \tag{28}$$

At the same time, for the edges between constraint vertices and column vertices, the corresponding attention coefficient is:

$$\mu_{ii'}^{(k)}=\frac{\exp(\varphi_{VC}(\boldsymbol{h}_i^{(k-1)}\|\boldsymbol{h}_{ii'}^{(k-1)}\|\boldsymbol{h}_{i'}^{(k-1)}))}{\sum_{i' \in \mathcal{F}(i)} \exp(\varphi_{VC}(\boldsymbol{h}_i^{(k-1)}\|\boldsymbol{h}_{ii'}^{(k-1)}\|\boldsymbol{h}_{i'}^{(k-1)}))} \tag{29}$$

where $\boldsymbol{h}_{ii'}$ is the embedding of the edge between vertex i and vertex i'; and φ_{VO} and φ_{VC} are the fully-connected feed-forward neural networks with LeakyReLU activation functions. LeakyReLU can be represented as $f(x)=\max(-\kappa x, x)$, and κ is typically set to be 0.2. In addition, each vertex in the bipartite graph will accumulate more information from other vertices by performing additional iterations. After K iterations, two fully-connected feed-forward neural networks with sigmoid activation functions, namely χ_C and χ_O, are used to predict the labels associated with each constraint vertex, i.e., the dual variable associated with constraint vertex i (the predicted value is denoted as $\tilde{\pi}_i$), and the objective function vertex, i.e., the objective function value associated with the objective function vertex (the predicted value is denoted as $\tilde{o}_i$), respectively.

B. GNN Based Column Selection

The GNN model proposed in Algorithm 1 is utilized to build a column selection method for accelerating the traditional CG method. In each iteration of the CG, we record the vector of dual variables associated with each constraint in (24).

$$\boldsymbol{\pi}=\mathop{\Big\|}_{\tau \in \Xi_H} \boldsymbol{\pi}_\tau \tag{30}$$

We let $\boldsymbol{\pi}^{(j)}$ represent the value of $\boldsymbol{\pi}$ in the j^{th} iteration of the CG process and $o^{(j)}$ represent the corresponding objective function value associated with (23) in the j^{th} iteration.

Compared with the traditional CG method, the proposed algorithm requires that each SP generates multiple feasible

columns in each iteration of the CG process. For example, if the SP is formulated as an MILP problem and solved using a commercial solver, e.g., Gurobi or CPLEX, then, multiple feasible columns can be obtained directly.

At the beginning of the j^{th} iteration of the CG process, each SP (27) first generates W ($W>1$) feasible columns $\{\hat{\boldsymbol{x}}_{r,\tau,\omega}^{(j)}\}_{\omega=1}^{W}$ ($r\in\Xi_g\cup\Xi_{ir}\cup\Xi_\delta$, $\tau\in\Xi_H$) using the dual variables in vector $\boldsymbol{\pi}^{(j-1)}$. The details of the proposed algorithm are described in Algorithm 2, where o^* is the minimum objective function value obtained from all previous iterations of the CG process, serving as a benchmark for evaluating and selecting the most promising columns in the current iteration. The main idea of Algorithm 2 is also illustrated in Fig. 3.

Algorithm 2: GNN based column selection algorithm

1: Generate the bipartite graph $\mathcal{G}$ according to the columns and coefficients of RMP at the beginning of the j^{th} iteration of the CG process
2: $o^*\leftarrow\min\{o^{(1)},o^{(2)},\ldots,o^{(j-1)}\}$
3: Define the dual center $\boldsymbol{\pi}_C^{(j)}$
4: Generate W aggregated columns $\{\hat{\boldsymbol{x}}_{\omega}^{(j)}\}_{\omega=1}^{W}$ using the columns generated by each SP
5: **for** $m=1$ to M **do**
6: Randomly select w different aggregated columns and add them to bipartite graph $\mathcal{G}$, and obtain the extended bipartite graph $\mathcal{G}'$
7: Predict the dual variable $\tilde{\pi}_i^{(m)}$ associated with each constraint vertex $i\in\Xi_C$ using the extended bipartite graph $\mathcal{G}'$ and Algorithm 1
8: $\tilde{\boldsymbol{\pi}}^{(m)}\leftarrow\Big\|_{i\in\Xi_C}\tilde{\pi}_i^{(m)}$
9: Predict the objective value $\tilde{o}^{(m)}$ associated with the objective function vertex using the extended bipartite graph $\mathcal{G}'$ and Algorithm 1
10: $\Gamma^{(m)}\leftarrow\text{dist}(\boldsymbol{\pi}_C^{(j)},\tilde{\boldsymbol{\pi}}^{(m)})-\gamma(o^*-\tilde{o}^{(m)})$
11: **end for**
12: $m^*=\arg\min_{1\le m\le M}\Gamma^{(m)}$
13: Add w aggregated columns selected in *Step 6* when $m=m^*$ to RMP

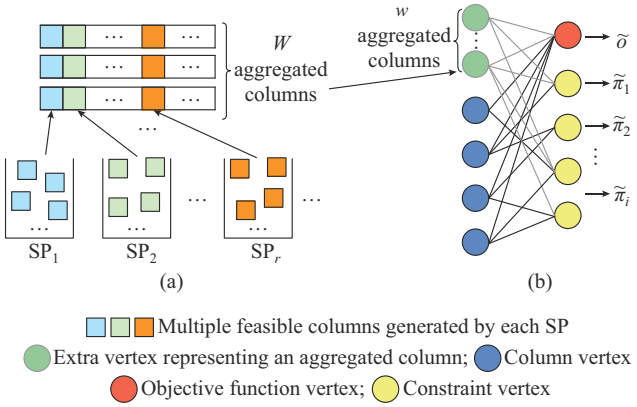

Fig. 3. Illustration of main idea of Algorithm 2. (a) Aggregated columns generated by concatenating columns generated by different SPs. (b) Extended bipartite graph $\mathcal{G}'$.

Please note that the aggregated column $\hat{\boldsymbol{x}}_{\omega}^{(j)}$ is generated by concatenating the column, and $\hat{\boldsymbol{x}}_{r,\tau,\omega}^{(j)}$ is generated by each SP ($r\in\Xi_g\cup\Xi_{ir}\cup\Xi_\delta$, $\tau\in\Xi_H$), as shown in Fig. 3(a). Moreover, the extended bipartite graph $\mathcal{G}'$ is obtained by adding selected aggregated columns to the bipartite graph $\mathcal{G}$, as shown in Fig. 3(b). The labels associated with the objective function vertex and constraint vertices are predicted, which will be used in selecting the most promising aggregated columns. At the end of the j^{th} iteration of the CG process, w ($1\le w\le W$) aggregated columns will be selected and added to the RMP. In Fig. 3(b), we add w aggregated columns to the bipartite graph $\mathcal{G}$ to generate the extended bipartite graph $\mathcal{G}'$. And in the next iteration (*Step 5* of Algorithm 2), the previous aggregated columns in $\mathcal{G}'$ will be replaced with new aggregated columns.

C. Discussion

In this subsection, we begin by providing a theoretical analysis of the feasibility of the proposed MPC scheme. We then proceed to discuss the convergence of the MPC as well as the attention mechanism.

1) Theoretical Analysis of Feasibility of MPC

According to [30], MPC is called feasible if all the constraints, i.e., (21) and (22), are satisfied.

In the proposed MPC scheme, we adopt soft-constraint method, which is commonly used in existing literature, e.g., [31]-[33], to ensure the feasibility of MPC. Specifically, we allow that the renewable generation (11) and interruptible loads (13) may be curtailed if necessary. We have $p_\delta^r(\tau)<\tilde{p}_\delta^r(\tau)$ or $p_d^r(\tau)<\tilde{p}_d^r(\tau)$ when the active power of renewable generation or interruptible load is curtailed; otherwise, the equality holds. And the same goes for reactive power. Furthermore, all the curtailment is penalized in the objective function of the MPC as:

$$\min\sum_{\tau\in\Xi_H}\left(\sum_{r_g\in\Xi_g}(\alpha_{r_g}p_g^{r_g}(\tau)+\beta_{r_g}u_g^{r_g}(\tau)\right)+\underbrace{\sum_{r_d\in\Xi_{ir}}\varepsilon^d\overbrace{(\tilde{p}_d^{r_d}(\tau)-p_d^{r_d}(\tau))}^{\text{Softening}}+\sum_{r_\delta\in\Xi_\delta}\varepsilon^\delta\overbrace{(\tilde{p}_\delta^{r_\delta}(\tau)-p_\delta^{r_\delta}(\tau))}^{\text{Softening}}}_{\text{Penalties}}\tag{31}$$

In fact, the curtailments are slack variables which soften the coupling constraint (21) of MG. The penalty factors ε^d and ε^δ are typically set to be significantly high values, surpassing the deployment cost of DERs such as generators and storage. This ensures that curtailing load or renewable generation is considered as a last resort, employed only when necessary to maintain the feasibility of the MPC.

Moreover, it is worth noting that (21) and (22) are formulated with a predicted trajectory of renewable generation and loads in the forward-looking horizon. However, in practice, renewable generation and loads cannot be predicted with perfect accuracy. And it is possible for unanticipated large forecast error to cause the output schedules of MPC to become infeasible. In this case, the MG energy management system will execute an emergency dispatch order [25] in practical engineering. The emergency dispatch order includes the changes in generation schedules and load shedding to restore feasibility.

Overall, the soft-constraint method adopted in this paper and the emergency dispatch order in practical engineering ensure the feasibility of the proposed MPC scheme.

2) Theoretical Analysis of Convergence of MPC

In the proposed MPC scheme, the optimization problems (20)-(22) are solved using Algorithm 2. Therefore, analyzing the convergence of MPC is tantamount to analyzing the convergence of the GNN based CG algorithm, which has the following property.

Property 1: the feasible domain associated with (20)-(22) is non-empty and bounded, and the GNN based CG algorithm will converge to the optimal solution in finite number of steps.

Proof: the soft-constraint method ensures that the feasible domain associated with (20)-(22) is non-empty, which has been explained earlier. Moreover, this domain is also bounded due to the limitations of power outputs of DERs and power consumption of loads. As a result, the number of feasible columns associated with each DER or load is finite. Since each column is generated and added to the RMP at most once, the CG algorithm will converge in a finite number of steps.

Property 2: let Z_{MP}^* and Z_{RMP}^* denote the optimal objective value of the mathematical programming (20)-(22) and the optimal objective value of RMP (23)-(26), respectively, and $\hat{c}_r^*$ denotes the minimal reduced cost in (27). Then, we can obtain:

$$Z_{RMP}^* + \sum_{r \in \Xi_g \cup \Xi_{ir} \cup \Xi_\delta} \hat{c}_r^* \le Z_{MP}^* \le Z_{RMP}^* \tag{32}$$

Proof: the solution to every RMP determines a feasible solution to the original optimization problem (20)-(22) in the CG process, which also provides an upper bound on the optimal objective function value of (20). Therefore, $Z_{MP}^* \le Z_{RMP}^*$. Meanwhile, we formulate the Lagrangian dual function $L(\boldsymbol{\pi})$, ($\boldsymbol{\pi} \ge \mathbf{0}$) associated with (20)-(22).

$$L(\boldsymbol{\pi}) = \min_{\boldsymbol{x}} \sum_{r \in \Xi_g \cup \Xi_{ir} \cup \Xi_\delta} \sum_{\tau \in \Xi_H} [\boldsymbol{c}_r^{\mathrm{T}} \boldsymbol{x}_{r,\tau} + \boldsymbol{\pi}_\tau^{\mathrm{T}} (\boldsymbol{b}_\tau - \boldsymbol{A}_r \boldsymbol{x}_{r,\tau})] \tag{33}$$

According to the duality theory, $L(\boldsymbol{\pi})$ gives a lower bound on the optimum in (20), i.e., Z_{MP}^*, for any $\boldsymbol{\pi} \ge \mathbf{0}$. By duality, we have the following property. Please note that $Z_{RMP}^* = \max\left(\sum \boldsymbol{\pi}_\tau^{\mathrm{T}} \boldsymbol{b}_\tau + \sum \sigma_r\right)$ according to the duality theory of linear programming.

$$\begin{aligned}
\mathcal{L} = & \max_{\boldsymbol{\pi} \ge \mathbf{0}} L(\boldsymbol{\pi}) = \\
& \max_{\boldsymbol{\pi} \ge \mathbf{0}} \min_{\boldsymbol{x}} \sum_{r \in \Xi_g \cup \Xi_{ir} \cup \Xi_\delta} \sum_{\tau \in \Xi_H} [\boldsymbol{c}_r^{\mathrm{T}} \boldsymbol{x}_{r,\tau} + \boldsymbol{\pi}_\tau^{\mathrm{T}} (\boldsymbol{b}_\tau - \boldsymbol{A}_r \boldsymbol{x}_{r,\tau}) + \boldsymbol{\sigma}_r - \boldsymbol{\sigma}_r] = \\
& \max_{\boldsymbol{\pi} \ge \mathbf{0}} \min_{\boldsymbol{x}} \left(\sum_{\tau \in \Xi_H} \boldsymbol{\pi}_\tau^{\mathrm{T}} \boldsymbol{b}_\tau + \sum_{r \in \Xi_g \cup \Xi_{ir} \cup \Xi_\delta} \boldsymbol{\sigma}_r \right) + \\
& \sum_{\tau \in \Xi_H} \sum_{r \in \Xi_g \cup \Xi_{ir} \cup \Xi_\delta} (\boldsymbol{c}_r^{\mathrm{T}} \boldsymbol{x}_{r,\tau} - \boldsymbol{\pi}_\tau^{\mathrm{T}} \boldsymbol{A}_r \boldsymbol{x}_{r,\tau} - \boldsymbol{\sigma}_r) = \\
& Z_{RMP}^* + \sum_{r \in \Xi_g \cup \Xi_{ir} \cup \Xi_\delta} \hat{c}_r^* \le Z_{MP}^*
\end{aligned} \tag{34}$$

Despite the theoretical guarantee of the convergence, traditional CG processes are known to have a slow convergence issue. The root cause of the issue lies in the fact that it selects columns by minimizing the reduced cost myopically based on the information available in the current iteration. In other words, only the primal objective function value of RMP, i.e., the upper bound Z_{RMP}^*, is considered when selecting columns in traditional CG, and the dual information that constitutes the lower bound $\left(Z_{RMP}^* + \sum \hat{c}_r^*\right)$ is neglected [11]. Thus, it cannot be guaranteed that the lower bound will continue to improve with increasing iterations. The convergence issue of traditional CG is mainly reflected in the following three aspects [11].

1) Dual oscillation: the optimal dual vector of each iteration may jump erratically from one extreme value to another.

2) Primal degeneracy: the optimization problems (20)-(22) are usually degenerate and thus the optimal dual vector is non-unique, which also contributes to dual oscillations.

3) Tailing-off effect: toward the end of the CG process, the convergence of the traditional CG becomes very slow.

At the same time, many smoothing and stabilizing mechanisms for improving the lower bound by considering the dual aspect have been proposed in [34], [35], which can be roughly classified into the following categories.

1) Setting bounds for dual variables to avoid large deviations.

2) Smoothing dual variables by taking convex combination of the dual variables in current and previous iterations.

3) Penalizing the deviation of the dual variables generated in current iteration from a dual center.

The three categories for smoothing and stabilizing purposes have a unified mathematical formulation, which consists of a dual center and a distance function that measures the distance between the dual variables generated in current iteration and the dual center.

In this paper, the proposed algorithm inherits the concepts of dual centers and distance function. It expedites the convergence of traditional CG by penalizing promising columns based on their distance from a dual center, which keeps improving the lower bound and results in a smoother, more stable, and faster convergence. More specifically, in Algorithm 2, the dual center $\boldsymbol{\pi}_C^{(j)}$ is defined at the beginning of the j^{th} iteration of the CG process. And the dual variables generated in current iteration $\tilde{\boldsymbol{\pi}}^{(m)}$ will be penalized according to its distance from the $\boldsymbol{\pi}_C^{(j)}$. In *Step 10* of Algorithm 2, the distance is denoted with $\text{dist}(\boldsymbol{\pi}_C^{(j)}, \tilde{\boldsymbol{\pi}}^{(m)})$, and the best aggregated columns are selected based on comprehensively considering the penalty of dual vectors and the improvement of the primal objective value. $\gamma > 0$ is the parameter which evaluates the relative importance of primal and dual information.

The definitions of $\boldsymbol{\pi}_C^{(j)}$ and dist(·) can take many forms. For example, [36] simply uses the optimal dual vector generated in last iteration as the stability center $\boldsymbol{\pi}_C^{(j)} = \boldsymbol{\pi}^{(j-1)}$, while the weighted sum of the dual vectors in previous iterations $\boldsymbol{\pi}_C^{(j)} = \sum_{\kappa=0}^{j-1} \varsigma^{j-\kappa} \boldsymbol{\pi}^{(\kappa)}$ is used in [11], where $\varsigma \in [0,1]$ is the discount factor. At the same time, dist(·) can be quadratic, piecewise linear, or boxtype [34]. There are also many other forms of the stability center and the distance function. However, we can observe that all the three categories of smoothing and stabilizing mechanisms for traditional CG can be implemented in Algorithm 2 through selecting appropriate $\boldsymbol{\pi}_C^{(j)}$ (*Step 3*), and dist(·) (*Step 10*), which reflects the flexibility of Algorithm 2. At the same time, it is worth mentioning that existing ML based acceleration methods for CG [12], [22] are usually tied to fixed dual centers and distance functions, which is difficult to guarantee that such methods can work in different MGs under different operating conditions. In fact, the smoothing and stabilizing scheme associated with Algorithm

2 can be customized according to the characteristics of the MG, which is a significant highlight of the proposed algorithm.

3) Analyzing Role of Attention Mechanism

The attention mechanism is a crucial component in improving the performance and effectiveness of GNN models. In this paper, we incorporate coefficient information from the RMP into the GNN model [37], utilizing an attention mechanism to reflect the importance of edge connections within the graph. This improves the efficiency and accuracy of the information processing capabilities of the GNN model. Additionally, attention mechanisms can highlight areas of focus during information processing, enhancing the interpretability of the GNN model.

Compared with the first graph attention model proposed in [38], the attention coefficient in (28) and (29) has the following improvements.

1) We adopt fully-connected feed-forward neural networks φ_{VO} and φ_{VC} to compute attention coefficients, instead of using trainable matrices as in [38]. The utilization of feed-forward neural networks provides significant approximation capabilities, which significantly enhances the efficiency of the proposed GNN model.

2) By incorporating the edge embedding $\boldsymbol{h}_{ii'}$ in our attention coefficient, we can leverage the coefficient information of the RMP, which is fully contained within the edge embedding, to reflect the importance of different neighboring vertices on a specific vertex in the GNN model. This improvement allows for a more accurate representation of the graph structure.

D. GNN Based MG Energy Management

The GNN based MG energy management is outlined in Algorithm 3, which can be integrated into a practical industrial controller, facilitating efficient online implementation.

Algorithm 3: GNN based MG energy management at time step t

1: $j \leftarrow 0$
2: Initialize the RMP with a few artificial columns and the initial dual vector $\boldsymbol{\pi}^{(0)}$ is obtained
3: $j \leftarrow j+1$
4: Each SP generates W feasible columns using $\boldsymbol{\pi}^{(j-1)}$
5: Calculate the reduced cost for each SP
6: **if** reduced cost of each SP becomes nonnegative **then**
7: Go to *Step 13*
8: **else**
9: Call Algorithm 2 to generate w aggregated columns that will be added to RMP
10: Solve RMP and obtain the optimal dual vector $\boldsymbol{\pi}^{(j)}$
11: Go to *Step 3*
12: **end if**
13: Recover the integer feasible solution

At the onset of each time step t, the RMP initiates with a set of artificial columns, and the initial dual vector $\boldsymbol{\pi}^{(0)}$ is computed. Next, Algorithm 2 is called to generate w aggregated columns that will be added to the RMP. The reduced cost for each SP is computed, and if the reduced costs for all SPs turn nonnegative, the final integer feasible solutions for all DERs and loads are restored. Otherwise, the RMP is solved again with the augmented columns, and the iterative process continues.

The branch-and-bound tree can be utilized to retrieve the integer feasible solutions by incorporating the GNN based CG at both the root node and subsequent branching nodes. For more comprehensive information, please refer to [7]. Due to space constraints, the details are not elaborated here.

Moreover, it is worth noting that the coupling constraint (24) and the local constraints of DERs or loads (27) of the MG are handled by the RMP (*Step 10* of Algorithm 3) and SPs (*Step 4* of Algorithm 3), respectively. Thus, the feasibility of the energy management provided by the GNN based MG energy management is guaranteed.

IV. Case Study

A. Test System Description

The GNN based MG energy management is tested on a small community MG (MG_S), a medium-sized MG (MG_M), and a large-scale MG (MG_L) to validate its efficacy. All three MGs incorporate critical loads, interruptible loads, dispatchable generators, renewable generation, and energy storage. The length of each time step is set as $T_\Delta = 5$ min.

The forecast horizon spans 1 hour, and each simulation extends over a period of 1 year. The system topology, along with data regarding dispatchable generators and energy storage, is extracted from [2] for MG_S, from [39] for MG_M, and from [40] for MG_L. Additionally, all renewable data are obtained from [41] and all load data are obtained from [42].

1) MG Energy Management Strategies (MGEMSs)

The following MGEMSs are simulated for comparison.

1) GNNCG-MGEMS: the proposed GNN based MGEMS in Algorithm 3.

2) TradCG-MGEMS: the traditional CG based MGEMS. Further details can be found in [7].

3) GNNCG$^{\text{cmp}}$-MGEMS: the GNN based CG used for MGEMS [12].

2) Performance Metrics

The effectiveness of MGEMSs is assessed by comparing their total generation cost (TGC) [43] in MG dispatching.

$$TGC = \sum_{t=1}^{T} \sum_{r_g \in \Xi_g} (\alpha_{r_g} p_g^{r_g}(t) + \beta_{r_g} u_g^{r_g}(t)) \tag{35}$$

At the same time, the computational performance of the MG energy management is evaluated by the following metrics. For convenience, the following metrics are only assessed at the root node of the branch-and-bound tree.

1) Avg CPU time: this is the average CPU time spent for the algorithm to converge per dispatch step in the simulation.

2) Avg iteration number: this is the average total iteration number taken for the algorithm to converge per dispatch step in the simulation.

3) Analysis Tools

The CG method including (23) - (27) is solved using the commercial LP and MILP solver CPLEX. The proposed GNNCG-MGEMS is validated using the Python based ML software Jupyter Notebook. The computer used is equipped with an Intel Core i5 2.70 GHz processor and 8 GB RAM.

B. Comparison with TradCG-MGEMS

First, the proposed GNNCG-MGEMS is compared against TradCG-MGEMS over the three MG test benches. Moreover, two different smoothing and stabilizing mechanisms are implemented in the proposed GNNCG-MGEMS, where the corresponding dual center and distance functions are defined as follows.

1) GNNCG[1]-MGEMS: the dual center is defined according to [36], where the optimal dual vector generated in the last iteration is used as the stability center $\boldsymbol{\pi}_C^{(j)}=\boldsymbol{\pi}^{(j-1)}$, and the distance function is quadratic $\text{dist}(\boldsymbol{\pi}_C^{(j)},\tilde{\boldsymbol{\pi}}^{(m)})=\left\|\boldsymbol{\pi}_C^{(j)}-\tilde{\boldsymbol{\pi}}^{(m)}\right\|_2$.

2) GNNCG[2]-MGEMS: the dual center is also defined according to [36], and the box-type distance function defined in [36] is used here.

In addition, in Algorithm 2, W and w are set to be 6 and 3, respectively, and γ is set to be 1. These values of hyper-parameters will be used throughout the entire case study.

We initially compare the TGC of GNNCG[1]-MGEMS and GNNCG[2]-MGEMS with TradCG-MGEMS in the three MGs, as illustrated in Fig. 4.

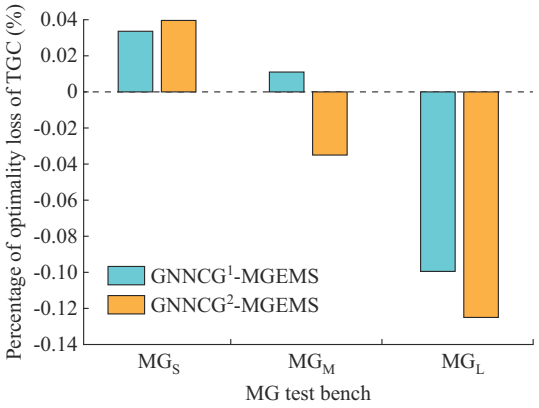

Fig. 4. Comparison of TGC of GNNCG[1]-MGEMS and GNNCG[2]-MGEMS with TGC of TradCG-MGEMS in three MG test benches.

The figure presents the percentages of optimality loss of TGC for GNNCG[1]-MGEMS and GNNCG[2]-MGEMS relative to for TradCG-MGEMS. The definition of the optimality loss is:

$$O_{pt,loss}=\frac{TGC_{\text{GNNCG}}-TGC_{\text{TradCG}}}{TGC_{\text{TradCG}}} \tag{36}$$

where TGC_{GNNCG} and TGC_{TradCG} are the TGCs of GNNCG[1]-MGEMS (or GNNCG[2]-MGEMS) and TradCG-MGEMS, respectively.

It can be found that the differences between the TGCs of GNNCG[1]-MGEMS (or GNNCG[2]-MGEMS) and TradCG-MGEMS are quite subtle in the three MGs. Moreover, the GNNCG[1]-MGEMS and GNNCG[2]-MGEMS outperform the TradCG-MGEMS in terms of TGC (the optimality losses become negative) in the MG_L. We can also observe that the advantage of the proposed GNNCG-MGEMS in TGC over the TradCG-MGEMS increases with the size of the MG.

Subsequently, we delve into the computational performance of MGEMSs, as summarized in Table I. The table reveals that the proposed GNNCG[1]-MGEMS and GNNCG[2]-MGEMS exhibit substantially lower average CPU time for each dispatch step compared with TradCG-MGEMS. Thus, GNNCG[1]-MGEMS (or GNNCG[2]-MGEMS) achieves comparable optimality in terms of operating costs than TradCG-MGEMS, significantly reducing the online computational burden. In addition, it is worth mentioning that in MG_L, the average CPU time required by TradCG-MGEMS to run each dispatch step is 314 s. It is even longer than the length of each time step ($T_\Delta=5$ min), which implies that the online optimization algorithm of TradCG-MGEMS cannot converge within the specified time period. Table I additionally indicates that the average iteration number per dispatch step for the proposed GNNCG-MGEMS is significantly less than that of TradCG-MGEMS. Therefore, by incorporating smoothing and stabilizing mechanisms, the proposed GNNCG-MGEMS achieves a much better computational effectiveness compared with TradCG-MGEMS.

TABLE I
COMPUTATIONAL PERFORMANCE OF THREE MGEMSS IN THREE MG TEST BENCHES

MGEMS	MG_S		MG_M		MG_L	
	Average CPU time (s)	Average iteration number	Average CPU time (s)	Average iteration number	Average CPU time (s)	Average iteration number
GNNCG[1]-MGEMS	19	8	42	12	118	28
GNNCG[2]-MGEMS	9	4	34	10	189	45
TradCG-MGEMS	58	26	129	41	314	78

Figure 5 further compares the convergence rates of GNNCG[1]-MGEMS and TradCG-MGEMS by observing the decrease of the objective value of the RMP (23). The corresponding dispatch step was executed in MG_L at 00:00 on January 1st. We can see that the proposed GNNCG[1]-MGEMS only needs 26 iterations to converge, while the TradCG-MGEMS needs approximately 120 iterations. Also, it can be observed that the TradCG-MGEMS has a long convergence tail (tailing-off effect) toward the end of the CG process. On the contrary, the GNNCG[1]-MGEMS converges fast and the objective value of RMP almost decreases with the same rate, which further validates the effectiveness of GNNCG-MGEMS.

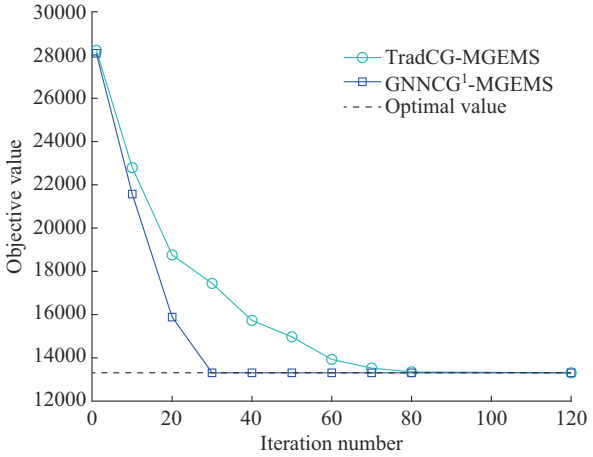

Fig. 5. Comparison between convergence rates of GNNCG[1]-MGEMS and TradCG-MGEMS.

We conduct simulations to quantitatively analyze the convergence speed of the proposed GNNCG[1]-MGEMS, GNNCG[2]-MGEMS, and the TradCG-MGEMS using box plots and convergence plots. Figure 6 compares the CPU time and iteration number for convergence associated with different MGEMSs in MG_L. We can observe that GNNCG[1]-MGEMS and GNNCG[2]-MGEMS exhibit statistically significant improvements in terms of the speed of convergence compared with TradCG-MGEMS. In Fig. 7, we present a convergence plot that shows the CG solving trajectories associated with GNNCG[1]-MGEMS and TradCG-MGEMS. For the sake of illustration, Fig. 7 only displays the trajectories of the first 300 dispatch steps in the simulation. We record the objective values of the RMP at each CG iteration, which are normalized to be in the range of [0,1] for ease of comparison. It is evident that TradCG-MGEMS always converges slower than the proposed GNNCG[1]-MGEMS.

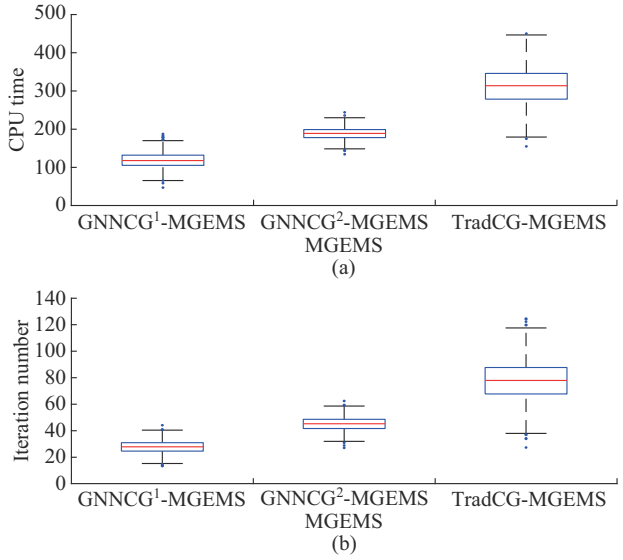

Fig. 6. Comparison of CPU time and iteration number for convergence associated with different MGEMSs in MG_L. (a) CPU time. (b) Iteration number.

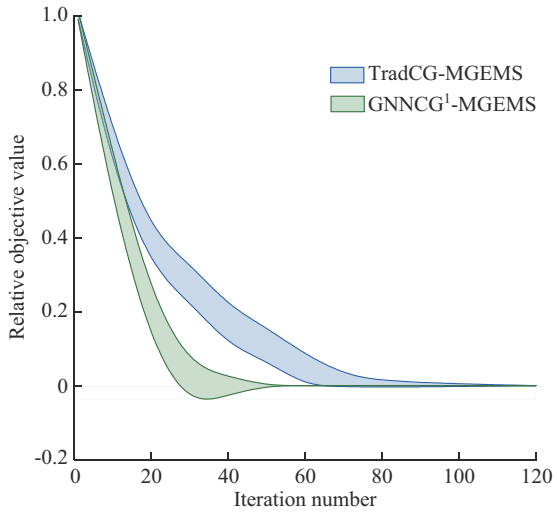

Fig. 7. Convergence plots of different MGEMSs.

C. Comparison with Existing GNNCG^{cmp}-MGEMS

Next, the proposed GNNCG[1]-MGEMS and GNNCG[2]-MGEMS are compared with the GNNCG^{cmp}-MGEMS. The performance of the three MGEMSs is tested under all the three MG test benches. The results are presented in Fig. 8. Figure 8 shows the percentages of optimality loss of the TGC associated with GNNCG[1]-MGEMS, GNNCG[2]-MGEMS, and GNNCG^{cmp}-MGEMS (taken with respect to the TGC of TradCG-MGEMS) in the three MG test benches. We can observe that GNNCG^{cmp}-MGEMS has similar performance in terms of TGC in the three MG test benches, which is surpassed by the proposed GNNCG[1]-MGEMS and GNNCG[2]-MGEMS in MG_M and MG_L. The computational performance of GNNCG^{cmp}-MGEMS is presented in Table II, where the metrics of GNNCG[1]-MGEMS and GNNCG[2]-MGEMS are also presented for comparison purposes. It can be observed from Table II that the average CPU time and the average iteration number per dispatch step of GNNCG^{cmp}-MGEMS are significantly higher than those associated with the proposed GNNCG-MGEMS in MG_M and MG_L, while in MG_S, the three MGEMSs have similar computational performance.

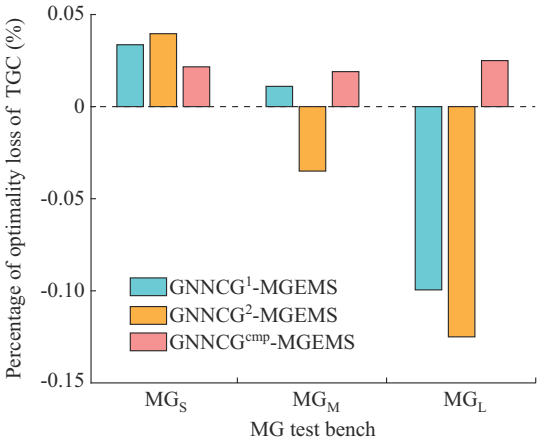

Fig. 8. Comparison of percentage of optimality loss of TGC associated with GNNCG[1]-MGEMS and GNNCG[2]-MGEMS with that of GNNCG^{cmp}-MGEMS in three MG test benches.

TABLE II
COMPUTATIONAL PERFORMANCE OF THREE MGEMSS IN THREE MG TEST BENCHES

MGEMS	MG_S		MG_M		MG_L	
	Average CPU time (s)	Average iteration number	Average CPU time (s)	Average iteration number	Average CPU time (s)	Average iteration number
GNNCG[1]-MGEMS	19	8	42	12	118	28
GNNCG[2]-MGEMS	9	4	34	10	189	45
GNNCG^{cmp}-MGEMS	16	7	78	23	278	66

In fact, the GNNCG^{cmp}-MGEMS proposed in [12] is tied to a fixed acceleration mechanism for CG, where only the objective values of the RMP and the number of selected col-

umns are considered. Such mechanism may work in an MG with a small size, but cannot guarantee its performance in MGs with medium or large sizes. On the contrary, the proposed GNNCG[1]-MGEMS and GNNCG[2]-MGEMS consider simultaneously the objective values of the RMP and the dual information (see *Step 10* of Algorithm 2), which smooths and stabilizes the dual variables and thus offers much better acceleration.

D. Comparison with GNN Models

We conduct simulations to compare the proposed GNNCG-MGEMS with the latest proposed GNN models in [19], [20], and [44]. And the following MGEMSs are simulated for comparison.

1) mwGNN-MGEMS: this MGEMS is produced by replacing the GNN model in GNNCG[1]-MGEMS with the GNN model proposed in [19]. Please note that the phenomenon of imbalanced data distribution discussed in [19] is not observed in our application, and all nodes are treated as "majority nodes" when training the GNN (in [19], a "majority node" means that the vertex in the GNN has sufficient sampling data).

2) dfiGNN-MGEMS: this MGEMS is produced by replacing the GNN model in GNNCG[1]-MGEMS with the GNN model proposed in [20].

3) pran-MGEMS: this MGEMS is produced by replacing the attention coefficient in GNNCG[1]-MGEMS with the attention mechanism proposed in [44].

All MGEMSs are implemented in MG_L. To help reflect the value of our proposed GNNCG-MGEMS, the time of the simulation is set to be 3 years, i.e., 3153605-min time intervals, which will form a large test data set.

Figure 9 compares the percentage of optimality loss of TGC associated with GNNCG[1]-MGEMS, GNNCG[2]-MGEMS, mwGNN-MGEMS, dfiGNN-MGEMS, and pran-MGEMS with that of TradCG-MGEMS in MG_L. The results show that GNNCG[1]-MGEMS and GNNCG[2]-MGEMS exhibit better optimization performance in terms of TGC, with smaller optimality loss compared with mwGNN-MGEMS, dfiGNN-MGEMS, and pran-MGEMS.

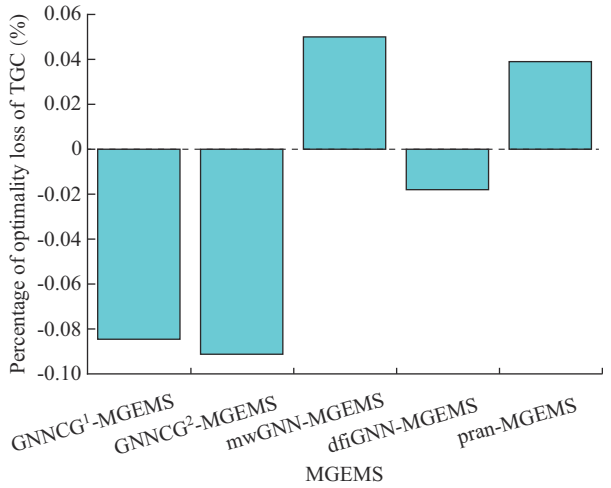

Fig. 9. Comparison of percentage of optimality loss of TGC associated with GNNCG[1]-MGEMS, GNNCG[2]-MGEMS, mwGNN-MGEMS, dfiGNN-MGEMS, and pran-MGEMS with that of TradCG-MGEMS in MG_L.

The computational performance of different MGEMSs in MG_L is analyzed in Table III. The results demonstrate that the proposed GNNCG[1]-MGEMS and GNNCG[2]-MGEMS require significantly lower CPU time on average to execute each dispatch step compared with the other four MGEMSs. Additionally, Table III shows that the average iteration numbers per dispatch step for mwGNN-MGEMS, dfiGNN-MGEMS, and pran-MGEMS are significantly higher than those of the proposed GNNCG[1]-MGEMS and GNNCG[2]-MGEMS in MG_L.

TABLE III
COMPUTATIONAL PERFORMANCE OF DIFFERENT MGEMSS IN MG_L

MGEMS	Average CPU time (s)	Average iteration number
GNNCG[1]-MGEMS	126	31
GNNCG[2]-MGEMS	171	42
mwGNN-MGEMS	262	63
dfiGNN-MGEMS	217	51
pran-MGEMS	228	57
TradCG-MGEMS	371	94

To further analyze the convergence speed of different MGEMSs, we use box plot shown in Fig. 10 and convergence plot shown in Fig. 11 to provide quantitative insights.

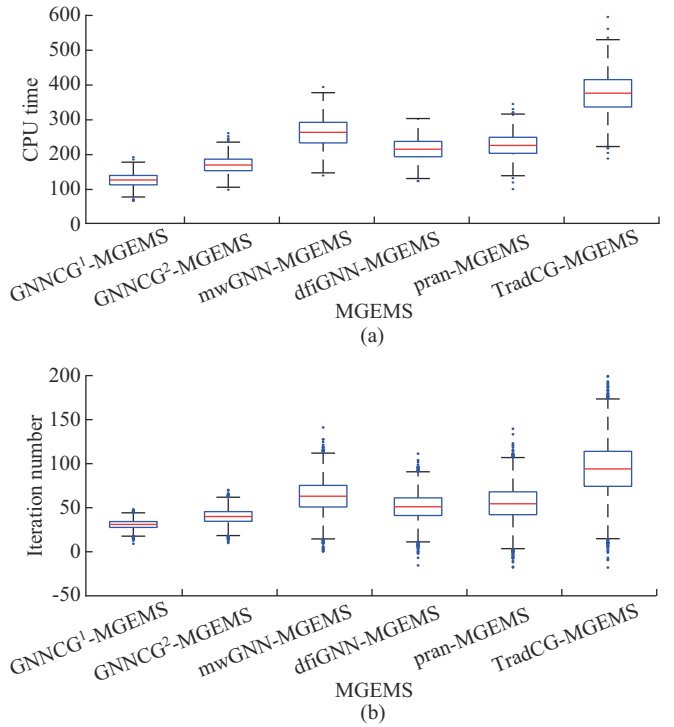

Fig. 10. Comparison of CPU time and iteration number for convergence associated with different MGEMSs. (a) CPU time. (b) Iteration number.

Figure 10 compares the CPU time and total iteration number taken for convergence per dispatch step associated with all MGEMSs presented in Table III using box plots.

By comparing the minimum, maximum, median, and quartiles of the distribution in Fig. 10 [45], it can be observed that the proposed GNNCG[1]-MGEMS and GNNCG[2]-MGEMS achieve statistically significant improvements in

the convergence speed of the CG method compared with the other four MGEMSs.

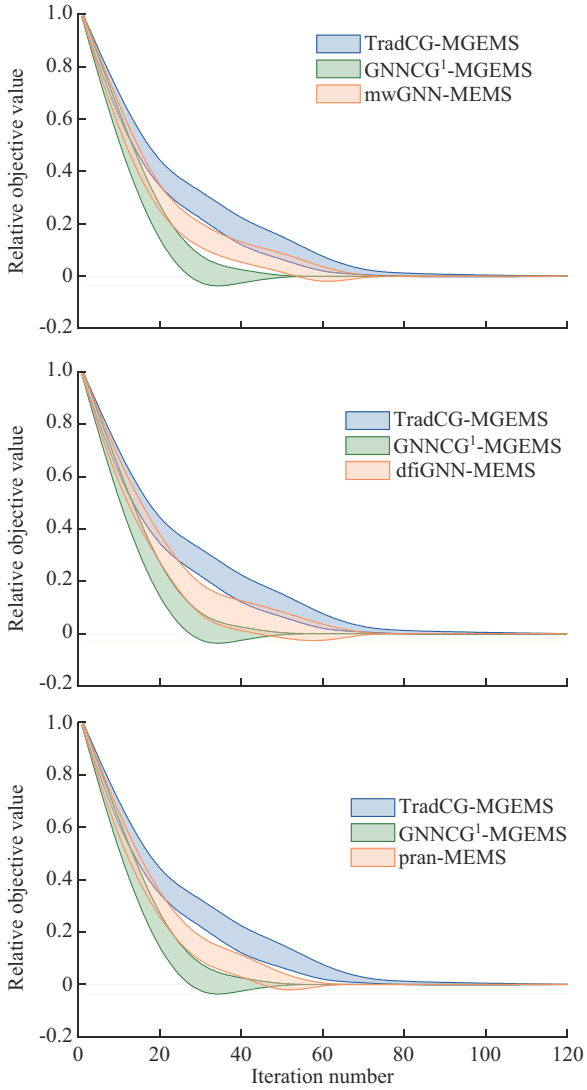

Fig. 11. Convergence plots of different MGEMS.

In the convergence plot shown in Fig. 11, we visualize the CG solving trajectories associated with different MGEMSs. Please note that Fig. 11 only shows the trajectories of the first 300 dispatch steps in the simulation. We record the objective values of the RMP at each CG iteration and normalize the objective values to be in the range of [0,1] for convenience. In Fig. 11, we compare the convergence curves of GNNCG[1]-MGEMS and TradCG-MGEMS with one of the curves in mwGNN-MGEMS, dfiGNN-MGEMS, and pran-MGEMS. It is clear that TradCG-MGEMS is always the slowest. Moreover, the proposed GNNCG[1]-MGEMS not only terminates in fewer iterations but also dominates all other four GNN models throughout the CG iterations.

V. Conclusion

In this paper, the MG energy management problem is formulated with the MPC scheme, which is reformulated based on the CG method afterwards. A GNN based framework has been developed to accelerate the convergence of the CG method by selecting promising columns that stabilize the dual variables associated with the RMP. The smoothing and stabilizing scheme associated with the GNN based framework can be customized according to the characteristic of the MG. A rigorous GNNCG-MGEMS is proposed for networked MG.

The verified results have demonstrated that the proposed GNNCG-MGEMS achieves the same optimality in terms of operating costs compared with traditional CG, while significantly reducing the computational cost including CPU time and iteration number needed for convergence. Compared with another GNN based accelerating method for traditional CG, the proposed algorithm allows the MG operator to customize the smoothing and stabilizing scheme for CG, which guarantees that the proposed algorithm can achieve much better computational performance in different networked MGs with different operating conditions. Finally, the proposed algorithm can leverage the physical model of the MG and satisfy the physical constraints of the networked MG.

In this paper, we do not take into account market mechanisms and environmental factors such as carbon emissions, which are also important for the operation of practical MGs. Ongoing research in this area involves incorporating market mechanisms into energy management to enable MG operators to participate in electricity markets and generate additional revenue streams. Additionally, incorporating environmental factors such as carbon emissions into the energy management can promote sustainable development.

Appendix A

In this paper, we consider the following features for different types of vertices and edges of the bipartite graph.

1) Column vertices: the total operating cost associated with the scheduling decisions in that column; the reduced cost of the column; and the total number of other vertices that a column vertex connects to.

2) Constraint vertices: the righthand-side value of the constraint in the RMP that corresponds to a constraint vertex in the current iteration; and the total number of other vertices that a constraint vertex connects to.

3) Objective function vertex: the total number of other vertices that the objective function vertex connects to; the edge between a column vertex and the objective function vertex; the coefficient vector of the column in the objective function associated with the RMP in the current iteration; the edge between a column vertex and a constraint vertex; and the coefficient vector of the column in the constraint associated with the RMP in the current iteration.

The labels are the optimal dual variables associated with (24) (i.e., all the constraint vertices), as well as the optimal objective function value of the RMP (23) (i.e., the objective function vertex) in the current iteration.

References

[1] M. A. Velasquez, J. Barreiro-Gomez, N. Quijano *et al.*, "Intra-hour microgrid economic dispatch based on model predictive control," *IEEE Transactions on Smart Grid*, vol. 11, no. 3, pp. 1968-1979, May 2020.

[2] Y. Huo, F. Bouffard, and G. Joós, "Spatio-temporal flexibility management in low-carbon power systems," *IEEE Transactions on Sustainable Energy*, vol. 11, no. 4, pp. 2593-2605, Oct. 2020.
[3] Y. Jia, Z. Y. Dong, C. Sun *et al*., "Distributed economic model predictive control for a wind-photovoltaic-battery microgrid power system," *IEEE Transactions on Sustainable Energy*, vol. 11, no. 2, pp. 1089-1099, Apr. 2020.
[4] D. P. Silva, J. L. F. Salles, J. F. Fardin *et al*., "Management of an island and grid-connected microgrid using hybrid economic model predictive control with weather data," *Applied Energy*, vol. 278, p. 115581, Nov. 2020.
[5] Y. Zhu, Y. Zhang, and Z. Wei, "Hierarchical cluster coordination control strategy for large-scale wind power based on model predictive control and improved multi-time-scale active power dispatching," *Journal of Modern Power Systems and Clean Energy*, vol. 11, no. 3, pp. 827-839, May 2023.
[6] J. Xu, H. Fu, S. Liao *et al*., "Demand-side management based on model predictive control in distribution network for smoothing distributed photovoltaic power fluctuations," *Journal of Modern Power Systems and Clean Energy*, vol. 10, no. 5, pp. 1326-1336, Sept. 2022.
[7] M. F. Anjos, A. Lodi, and M. Tanneau, "A decentralized framework for the optimal coordination of distributed energy resources," *IEEE Transactions on Power Systems*, vol. 34, no. 1, pp. 349-359, Jan. 2019.
[8] C. Saldarriaga-Cortés, H. Salazar, R. Moreno *et al*., "Stochastic planning of electricity and gas networks: an asynchronous column generation approach," *Applied Energy*, vol. 233-234, pp. 1065-1077, Jan. 2019.
[9] A. Flores-Quiroz and K. Strunz, "A distributed computing framework for multi-stage stochastic planning of renewable power systems with energy storage as flexibility option," *Applied Energy*, vol. 291, p. 116736, Jun. 2021.
[10] T. Schütz, X. Hu, M. Fuchs *et al*., "Optimal design of decentralized energy conversion systems for smart microgrids using decomposition methods," *Energy*, vol. 156, pp. 250-263, Aug. 2018.
[11] A. Pessoa, R. Sadykov, E. Uchoa *et al*., "Automation and combination of linear-programming based stabilization techniques in column generation," *INFORMS Journal on Computing*, vol. 30, no. 2, pp. 339-360, May 2018.
[12] M. Morabit, G. Desaulniers, and A. Lodi, "Machine learning-based column selection for column generation," *Transportation Science*, vol. 55, no. 4, pp. 815-831, Jun. 2021.
[13] Y. Bengio, A. Lodi, and A. Prouvost, "Machine learning for combinatorial optimization: a methodological tour d'horizon," *European Journal of Operational Research*, vol. 290, no. 2, pp. 405-421, Apr. 2021.
[14] G. Ruan, H. Zhong, G. Zhang *et al*., "Review of learning-assisted power system optimization," *CSEE Journal of Power and Energy Systems*, vol. 7, no. 2, pp. 221-231, Mar. 2021.
[15] Y. Du and F. Li, "Intelligent multi-microgrid energy management based on deep neural network and model-free reinforcement learning," *IEEE Transactions on Smart Grid*, vol. 11, no. 2, pp. 1066-1076, Mar. 2020.
[16] Q. Zhang, K. Dehghanpour, Z. Wang *et al*., "A learning-based power management method for networked microgrids under incomplete information," *IEEE Transactions on Smart Grid*, vol. 11, no. 2, pp. 1193-1204, Mar. 2020.
[17] R. Lu, S. Hong, and M. Yu, "Demand response for home energy management using reinforcement learning and artificial neural network," *IEEE Transactions on Smart Grid*, vol. 10, no. 6, pp. 6629-6639, Nov. 2019.
[18] E. Mocanu, D. C. Mocanu, P. H. Nguyen *et al.*, "On-line building energy optimization using deep reinforcement learning," *IEEE Transactions on Smart Grid*, vol. 10, no. 4, pp. 3698-3708, Jul. 2019.
[19] K. Wang, J. An, M. Zhou *et al*., "Minority-weighted graph neural network for imbalanced node classification in social networks of internet of people," *IEEE Internet of Things Journal*, vol. 10, no. 1, pp. 330-340, Jan. 2023.
[20] Z. Zhao, Z. Yang, C. Li *et al*., "Dual feature interaction-based graph convolutional network," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 9, pp. 9019-9030, Sept. 2023.
[21] H. Yuan, P. Jiang, and S. Song, "The neural-prediction based acceleration algorithm of column generation for graph-based set covering problems," in *Proceedings of 2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, Niagara Falls, Canada, Oct. 2022, pp. 1115-1120.
[22] B. Babaki, S. D. Jena, and L. Charlin, "Neural column generation for capacitated vehicle routing," in *Proceedings of AAAI-22 Workshop on Machine Learning for Operations Research (ML4OR)*, Vancouver, Canada, Dec. 2022, pp. 1-8.
[23] C. Chi, A. Aboussalah, E. Khalil *et al*., "A deep reinforcement learning framework for column generation," *Advances in Neural Information Processing Systems*, vol. 35, pp. 9633-9644, Oct. 2022.
[24] J. Ekström, M. Koivisto, I. Mellin *et al*., "A statistical model for hourly large-scale wind and photovoltaic generation in new locations," *IEEE Transactions on Sustainable Energy*, vol. 8, no. 4, pp. 1383-1393, Oct. 2017.
[25] *IEEE Standard for the Specification of Microgrid Controllers*, IEEE Standard 2030.7-2017.
[26] Y. Huo, Z. Chen, J. Bu *et al*., "Learning assisted column generation for model predictive control based energy management in microgrids," *Energy Reports*, vol. 9, pp. 88-97, Sept. 2023.
[27] Z. Wang, B. Chen, J. Wang *et al*., "Coordinated energy management of networked microgrids in distribution systems," *IEEE Transactions on Smart Grid*, vol. 6, no. 1, pp. 45-53, Jan. 2015.
[28] H. Lin, F. Ul Nazir, B. C. Pal *et al*., "A linearized branch flow model considering line shunts for radial distribution systems and its application in volt/var control," *Journal of Modern Power Systems and Clean Energy*, vol. 11, no. 4, pp. 1191-1200, Jul. 2023.
[29] F. Katiraei, R. Iravani, N. Hatziargyriou *et al*., "Microgrids management," *IEEE Power and Energy Magazine*, vol. 6, no. 3, pp. 54-65, May 2008.
[30] F. Borrelli, A. Bemporad, and M. Morari, *Predictive Control for Linear and Hybrid Systems.* Cambridge: Cambridge University Press, 2017.
[31] X. Ping, J. Hu, T. Lin *et al*., "A survey of output feedback robust mpc for linear parameter varying systems," *IEEE/CAA Journal of Automatica Sinica*, vol. 9, no. 10, pp. 1717-1751, Oct. 2022.
[32] M. Schwenzer, M. Ay, T. Bergs *et al*., "Review on model predictive control: An engineering perspective," *The International Journal of Advanced Manufacturing Technology*, vol. 117, no. 5, pp. 1327-1349, Aug. 2021.
[33] P. O. Scokaert and J. B. Rawlings, "Feasibility issues in linear model predictive control," *AIChE Journal*, vol. 45, no. 8, pp. 1649-1659, Apr. 1999.
[34] J. Desrosiers and M. E. Lbbecke, *A Primer in Column Generation*. Boston: Springer US, 2005, pp. 1-32.
[35] M. Morabit, G. Desaulniers, and A. Lodi, "Machine-learning-based arc selection for constrained shortest path problems in column generation," *INFORMS Journal on Optimization*, vol. 5, no. 2, pp. 191-210, Apr. 2023.
[36] L. M. Rousseau, M. Gendreau, and D. Feillet, "Interior point stabilization for column generation," *Operations Research Letters*, vol. 35, no. 5, pp. 660-668, Sept. 2007.
[37] W. Liao, B. Bak-Jensen, J. R. Pillai *et al*., "A review of graph neural networks and their applications in power systems," *Journal of Modern Power Systems and Clean Energy*, vol. 10, no. 2, pp. 345-360, Mar. 2022.
[38] P. Veličković, G. Cucurull, A. Casanova *et al*., "Graph attention networks," in *Proceedings of International Conference on Learning Representations (ICLR)*, Vancouver, Canada, Oct. 2018 , pp. 1-12.
[39] V. Mohan, R. Suresh, J. G. Singh *et al*., "Microgrid energy management combining sensitivities, interval and probabilistic uncertainties of renewable generation and loads," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 7, no. 2, pp. 262-270, Jun. 2017.
[40] C. Zhang, Y. Xu, Z. Y. Dong *et al*., "Robust operation of microgrids via two-stage coordinated energy storage and direct load control," *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 2858-2868, Jul. 2017.
[41] National Renewable Energy Laboratory. (2023, Sept.). Renewable resource data. [Online]. Available: https://www.nrel.gov/grid/data-tools.html
[42] National Renewable Energy Laboratory. (2023, Sept.). End-use load profiles. [Online]. Available: https://www. nrel. gov/ buildings/end-use-load- profiles.html
[43] Y. Huo, F. Bouffard, and G. Jos, "Integrating learning and explicit model predictive control for unit commitment in microgrids," *Applied Energy*, vol. 306, p. 118026, Jan. 2022.
[44] T. Sun, C. Wang, H. Dong *et al*., "A novel parameter-optimized recurrent attention network for pipeline leakage detection," *IEEE/CAA Journal of Automatica Sinica*, vol. 10, no. 4, pp. 1064-1076, Apr. 2023.
[45] R. R. Hossain and R. Kumar, "Machine learning accelerated real-time model predictive control for power systems," *IEEE/CAA Journal of Automatica Sinica*, vol. 10, no. 4, pp. 916-930, Apr. 2023.

Yuchong Huo received the B.Eng. and M.S. degrees in electrical power engineering from Southeast University, Nanjing, China, in 2012 and 2015, respectively, and the Ph.D. degree in electrical power engineering from McGill University, Montreal, Canada, in 2021. He is currently a Lecturer with the School of Automation, Nanjing University of Science and Technology, Nanjing, China. His research interests include power system flexibility, economics, and control.

Zaiyu Chen received the B.S. degree in information and computing science and the Ph.D. degree in control science and engineering from the Nanjing University of Science and Technology, Nanjing, China, in 2012 and 2019, respectively. From 2020 to 2021, he was a Research Fellow with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. He is currently a Lecturer with the School of Automation, Nanjing University of Science and Technology. His research interests include active power control for wind turbines and its application in power system frequency regulation.

Qun Li received the B.Eng. degree in electronic equipment structure design from Southeast University, Nanjing, China, in 1989, the M.Eng. degree in electronic equipment structure design from Xidian University, Xi'an, China, in 1995, and the Ph.D. degree in power system and its automation from Southeast University, in 1999. He is currently a Professor Engineer with State Grid Jiangsu Electric Power Co., Ltd. Research Institute, Nanjing, China. His research interests include optimization of power system operation and renewable power integration.

Qiang Li received the B.Eng., M.Eng., and Ph.D. degrees in power systems and automation from Hohai University, Nanjing, China, in 2003, 2006, and 2011, respectively. He is currently a Professorate Senior Engineer with the State Grid Jiangsu Electric Power Co., Ltd. Research Institute, Nanjing, China. His research interests include flexible transmission of mid-far offshore wind power.

Minghui Yin received the B.Eng. and M.Eng. degrees in electrical power engineering, and the Ph.D. degree in control science and engineering from the Nanjing University of Science and Technology, Nanjing, China, in 1999, 2002, and 2009, respectively. From 2007 to 2008, he was a Research Assistant with the Department of Electrical Engineering, The Hong Kong Polytechnic University, Hong Kong, China. From 2016 to 2017, he was a Visiting Scholar with the School of Electrical and Information Engineering, The University of Sydney, Sydney, Australia. He is currently a Professor with the School of Automation, Nanjing University of Science and Technology. His research interests include wind power conversion system and power system transient stability.