An Improved Spatial Branch-and-Bound Algorithm for Non-Convex Optimal Electricity-Gas Flow

Pengxiang Liu, *Student Member, IEEE*, Zhi Wu, *Member, IEEE*, Wei Gu, *Senior Member, IEEE*, Yuping Lu, *Senior Member, IEEE*

***Abstract*—Addressing non-convexity plays a fundamental role in solving the optimal electricity-gas flow models. In this paper, an improved spatial branch-and-bound algorithm is proposed to solve the non-convex problem, which is formulated as a mixed-integer bilinear programming, for its exact solution. The core of the algorithm is to divide the non-convex model into convex and small sub-models by branching on specific continuous variables, so that the non-convex problem can be equivalent to a rooted tree for exploration. The exactness of the algorithm is guaranteed by the same criterion as the classical branch-and-bound algorithm. To alleviate the computational burden, a novel two-stage spatial branching strategy is developed to improve the effectiveness and efficiency of the branching operations. The performance of the proposed algorithm is verified on two integrated electricity-gas systems with different sizes. Numerical results demonstrate that our method achieves a balance among feasibility, optimality, and efficiency. The comparison with another 6 convexification-based methods, 3 state-of-the-art non-convex optimization solvers, and 2 spatial branch-and-bound algorithms with classical branching rules further shows the superiority of our algorithm.**

***Index Terms*—Optimal electricity-gas flow, branching strategy, spatial branch-and-bound, mixed-integer bilinear programming, non-convex optimization.**

NOMENCLATURE

A. Abbreviations

AMP	Adaptive multivariate partitioning
BT	Bound tightening
CCP	Convex-concave procedure
CPG	Conventional power plant
IEGS	Integrated electricity-gas system
GFU	Gas-fired unit
MIBLP	Mixed-integer bilinear programming
OEGF	Optimal electricity-gas flow
OPF	Optimal power flow
PWL	Piecewise linearization
RLT	Reformulation-linearization technique
sB&B	Spatial branch-and-bound
SCP	Sequential cutting-plane
SOCP	Second order conic programming

B. Indices and Sets

i, j	Indices of buses in power system
m, n	Indices of nodes in gas system
t	Index of time intervals
Ref	Index of reference (slack) buses
Ω_b, Ω_l	Sets of buses and lines
Ω_c, Ω_u	Sets of CPG and GFU buses
Ω_k, Ω_s	Sets of gas compressors and suppliers
Ω_n, Ω_p	Sets of nodes and pipelines
Ω_t	Set of time intervals
α_i^-, α_i^+	Sets of lines whose from/to bus is bus i
β_m^-, β_m^+	Sets of gas compressors with node m as its inlet/outlet node
γ_m^-, γ_m^+	Sets of pipelines with node m as the head-end/tail-end node

C. Parameters

C_i^a, C_i^b, C_i^c	Cost coefficients of power generation
C_m^p	Cost coefficient of gas production
C_i^s	Cost coefficient of load curtailment
$D_{i,t}^p, D_{i,t}^q$	Active/reactive load of bus i
$D_{m,t}^g$	Gas demand of node m
$F_{e,mn}^{max}$	Maximum gas flow of compressor mn
F_{mn}^{max}	Maximum gas flow of pipeline mn
$F_{u,m}^{max}$	Maximum gas consumption of GFU m
$F_{s,m}^{min}, F_{s,m}^{max}$	Output ranges of gas supplier m
$F_{s,m}^{ramp}$	Ramping limit of gas supplier m
G_{ij}, B_{ij}	Conductance/susceptance of line ij
$H_{mn}^{min}, H_{mn}^{max}$	Compression ratio ranges of compressor mn
K_{mn}	Line-pack parameter of pipeline mn
L^{min}	Minimum line-pack for gas system
M	A sufficient large number
$P_{c,i}^{ramp}, P_{u,i}^{ramp}$	Ramping limits of CPG and GFU i
$P_{ij}^{min}, P_{ij}^{max}$	Active power flow ranges of line ij
$P_{c,i}^{min}, P_{c,i}^{max}$	Active power output ranges of CPG i
$P_{u,i}^{min}, P_{u,i}^{max}$	Active power output ranges of GFU i
$Q_{ij}^{min}, Q_{ij}^{max}$	Reactive power flow ranges of line ij
$Q_{c,i}^{min}, Q_{c,i}^{max}$	Reactive power output ranges of CPG i
$Q_{u,i}^{min}, Q_{u,i}^{max}$	Reactive power output ranges of GFU i
T_m	Energy conversion coefficient of GFU m
V_i^{min}, V_i^{max}	Lower/upper limit of voltage of bus i
W_{mn}	Weymouth constant of pipeline mn
Π_m^{min}, Π_m^{max}	Lower/upper limit of pressure of node m
η_{mn}	Conversion efficiency of compressor mn

D. Variables

$f_{m,t}^u$	Gas consumption of GFU m
$f_{m,t}^s$	Gas injection of supplier m
$f_{mn,t}$	Average gas flow of pipeline mn

This work is supported by the National Key Technologies R & D Program of China, Grant No. 2020YFE0200400 (Corresponding author: Zhi Wu).

P. Liu, Z. Wu, W. Gu and Y. Lu are with the School of Electrical Engineering, Southeast University, Nanjing, China. (e-mail: 230198646@seu.edu.cn, zwu@seu.edu.cn, wgu@seu.edu.cn, luyuping@seu.edu.cn)

$f^-_{mn,t}, f^+_{mn,t}$	Gas flow at the head-end (node m) / tail-end (node n) of pipeline mn
$f^e_{mn,t}$	Gas flow of compressor mn
$l_{mn,t}$	Line-pack of pipeline mn
$p^c_{i,t}, q^c_{i,t}$	Active/reactive power output of CPG i
$p^u_{i,t}, q^u_{i,t}$	Active/reactive power output of GFU i
$p^s_{ij,t}, q^s_{ij,t}$	Active/reactive power losses of line ij
$p_{ij,t}, q_{ij,t}$	Active/reactive power flow of line ij
$v_{i,t}$	Squared voltage magnitude of bus i
$w^p_{i,t}, w^q_{i,t}$	Active/reactive load curtailment of bus i
$\theta_{i,t}$	Voltage angle of bus i
$\theta_{ij,t}$	Angle difference of line ij
$\tau_{mn,t}$	Gas consumption of compressor mn
$\pi_{m,t}$	Gas pressure of node m
$\sigma^s_{ij,t}$	Auxillary variable for line losses
$\sigma^w_{mn,t}, \sigma^a_{mn,t}$	Auxillary variables for gas flow equations
$\sigma^-_{mn,t}, \sigma^+_{mn,t}$	Auxillary variables for nodal pressures
λ_{mn}	Binary variable representing the direction of gas flow in pipeline mn

I. Introduction

With the popularization of gas-fired units (GFUs), the interaction between power system and gas system has been enhanced [1]. As one of the mathematical foundations, optimal electricity-gas flow (OEGF) is of great significance to the planning [2] and operation [3] of such integrated energy systems. Since both steady-state optimal power flow and gas flow models are nonlinear and non-convex, which makes the problem non-deterministic polynomial-time hard, tremendous effort has been devoted to addressing the issue for the secure and economic operations of integrated electricity-gas systems (IEGSs) over the past few years.

Linear approximation and convex relaxation are two basic methods to ensure the computational efficiency and robustness of solving OEGF problems. For the linearization approaches, DC optimal power flow (OPF) model [4] is widely adopted to simplify the problem by ignoring reactive power and voltage magnitude. To further characterize the nonlinearity, piecewise linearization (PWL) [5] and Taylor series expansion [6] are two most commonly-used methods to approximate nonlinear gas flow equations. However, linear approximation approach makes a certain level of compromise on the accuracy of solutions. By comparison, convex relaxation is another approach to reformulate the quadratic equality constraints. For example, reference [7] discusses the technology of second order conic programming (SOCP) in both radial and meshed grids. In [8], a mixed-integer SOCP model is proposed to cope with the non-convex Weymouth equations in gas systems. In order to further improve the tightness of such conic relaxation in both power and gas systems, convex envelops are investigated in [9], [10] to eliminate as many infeasible operational points as possible. Although the SOCP-based models are much easier to solve, if certain conditions cannot be held [7], as is usually the case, the tightness is still hard to guarantee.

Ensuring tightness (feasibility) is a critical requirement for an OEGF-oriented algorithm due to the concerns for security and reliability, especially when the obtained OEGF solutions are applied to the implementation of an intra-day or real-time dispatch scheme. Therefore, tightening approaches have been given a lot of attention in recent years. For example, the first-order Taylor series expansion was used in [11] to approximate the nonlinear line loss terms in an iterative manner. A sparse semidefinite programming based algorithm was developed in [12] to form a tight and traceable relaxation of the Weymouth equality constraints. In [13], a bound tightening (BT) method was proposed to tighten the bounds of convex hulls. Besides, iterative correction methods such as sequential cutting-planes (SCP) [14] and convex-concave procedure (CCP) [15] are two heuristics approaches whose core idea is to redirect a feasible solution based on the obtained non-tight one. However, these tightening approaches heavily depend on the quality of initial solutions, and none of them is capable of ensuring the global optimality of the solutions. Certainly, using a general-purpose nonlinear optimization solver (e.g., Ipopt [16]) might be one of the simplest ways to handle the OEGF problem [17]–[19]. However, the inability to deal with binary variables limits its application to continuous optimization.

Motivated by the pressing need for solving the non-convex OEGF models and their mixed-integer extensions, the spatial branch-and-bound (sB&B) algorithm, which could be the best available method to solve mixed-integer non-convex programs to ϵ-global optimality [20], is explored in this paper. It works by iteratively partitioning the domain of continuous variables so that tighter relaxations can be achieved to prune infeasible solutions. Currently, the sB&B algorithm has been embedded into many well-known solvers such as Gurobi [21] and SCIP [22] and has been adopted to solve benchmark mixed-integer nonlinear instances (including pooling and blending problems [23]). In the field of global optimization, the improvement in sB&B algorithm is still being explored. For example, a multi-way branching technique is developed in [24] to improve the performance of the conventional guided 2-way branching. In [25], a shifted branching strategy is presented to quantify the rationality of branching operations. The core of [24] and [25] is to enhance the McCormick relaxations of bilinear terms. A similar idea is adopted in [26] where an adaptive multivariate partitioning (AMP) algorithm is proposed based on piecewise McCormick constraints. However, since these works focus on developing a general-purpose algorithm where all non-convex constraints are converted to bilinear terms, the criterion for the selection of branching position is limited to the imbalance of bilinear constraints only. In other words, the resulting mixed-integer bilinear programming (MIBLP) reformulation discards the imbalance of other types of non-convex constraints. Thus, this limitation greatly impairs the efficiency in solving large-scale optimization problems. Moreover, it has been observed that the performance of sB&B algorithm, as well as other global optimization methods, is highly dependent on the type of non-convexities [27]. Therefore, by exploiting the special mathematical structure of the model to be solved, it is promising to devise improvements that lead to a better performance of the sB&B algorithm.

To sum up, these studies [5]–[15], [24]–[26] are trapped in a dilemma among feasibility, optimality, and efficiency while solving non-convex models. To be specific, the approximation

and relaxation based strategies [5]–[10] cannot guarantee the tightness of non-convex constraints (i.e., the obtained solution is also infeasible) in spite of the high efficiency. By contrast, although the tightening methods in [11]–[15] are able to find a solution which satisfies all non-convex constraints, none of them can guarantee its ϵ-global optimality. As to the existing sB&B algorithms [24], [25], as well as their variant [26], the computing efficiency is still far from real application.

To address the problems, this paper develops an improved sB&B algorithm to achieve a balance among feasibility, optimality and efficiency. The proposed algorithm is oriented to a specific type of optimization in which non-convex constraints are troubled by non-tight relaxations. Although the programs of this type (including the OEGF problem in this paper) are quite common in the power industry, few studies have devised specific strategies to improve the sB&B algorithm.

The major contributions are summarized as follows.

1) A high-efficiency sB&B algorithm is developed to solve the OEGF models, where the feasibility and optimality of the obtained solutions can be ensured within a flexible threshold, so that a balance among feasibility, optimality and efficiency can be achieved.

2) A novel two-stage spatial branching strategy is designed to determine a preferable branching position. For one thing, the strategy can balance the complexity of each branching node. For another, the errors of conic relaxation are incorporated to improve the efficiency of branching operations.

The remainder of the paper is organized as follows. Section II introduces the mathematical formulation of the non-convex OEGF model. Section III presents the solution methodology which includes the novel two-stage spatial branching strategy. Case studies are conducted in section IV, and discussions are given in section V. Finally, section VI concludes the paper.

II. Mathematical Modeling

A. Assumptions and Simplifications

Before the mathematical modeling, some assumptions and simplifications are made first: 1) The dynamics of natural gas pipelines is neglected, and the steady-state gas flow model is used [10], [13]. 2) A simplified compressor model where the expressions of fuel consumption are linearized is adopted [6], [12]. 3) A linearized OPF model with quadratic line losses is employed [8], [11] based on the assumptions that the voltage magnitude at each bus is close to 1, and the angle difference across each line is close to 0 (in the per-unit system). 4) The IEGSs are operated by a single entity.

B. The Non-Convex OEGF Model

The model is assumed to be applied for short-term or day-ahead scheduling of IEGSs. The objective of the model is to minimize the total costs of the system.

$$\begin{aligned}\min \quad & \sum_{t\in\Omega_t}\sum_{i\in\Omega_c}\left[C_i^a\left(p_{i,t}^c\right)^2+C_i^b\,p_{i,t}^c+C_i^c\right] \\ & +\sum_{t\in\Omega_t}\sum_{m\in\Omega_s}C_m^p f_{m,t}^s+\sum_{t\in\Omega_t}\sum_{i\in\Omega_b}C_i^s\left(w_{i,t}^p+w_{i,t}^q\right)\end{aligned} \tag{1}$$

where the first term represents the operating cost of conventional power generation (CPG), the second term is the natural gas production cost, and the last one denotes the penalty cost of load curtailment. Note that the parameters that used in this section are given in upper-case letters, while the variables are represented in lower-case letters.

Based on the OPF model in [8], [11], the electricity side of the OEGF problem can be formulated as follows.

$$\begin{aligned}\sum_{ji\in\alpha_i^+}\left(p_{ji,t}-p_{ji,t}^s\right)-\sum_{ik\in\alpha_i^-}p_{ik,t} \\ +p_{i,t}^c+p_{i,t}^u=D_{i,t}^p-w_{i,t}^p,\quad \forall i\in\Omega_b,\forall t\in\Omega_t\end{aligned} \tag{2}$$

$$\begin{aligned}\sum_{ji\in\alpha_i^+}\left(q_{ji,t}-q_{ji,t}^s\right)-\sum_{ik\in\alpha_i^-}q_{ik,t} \\ +q_{i,t}^c+q_{i,t}^u=D_{i,t}^q-w_{i,t}^q,\quad \forall i\in\Omega_b,\forall t\in\Omega_t\end{aligned} \tag{3}$$

$$p_{ij,t}=\frac{1}{2}G_{ij}\left(v_{i,t}-v_{j,t}\right)-B_{ij}\,\theta_{ij,t},\quad \forall ij\in\Omega_l,\forall t\in\Omega_t \tag{4}$$

$$q_{ij,t}=\frac{1}{2}B_{ij}\left(v_{j,t}-v_{i,t}\right)-G_{ij}\,\theta_{ij,t},\quad \forall ij\in\Omega_l,\forall t\in\Omega_t \tag{5}$$

$$P_{\{\cdot\},i}^{min}\le p_{i,t}^{\{\cdot\}}\le P_{\{\cdot\},i}^{max},\ \forall i\in\Omega_{\{\cdot\}},\forall t\in\Omega_t,\{\cdot\}=\{c,u\} \tag{6}$$

$$Q_{\{\cdot\},i}^{min}\le q_{i,t}^{\{\cdot\}}\le Q_{\{\cdot\},i}^{max},\ \forall i\in\Omega_{\{\cdot\}},\forall t\in\Omega_t,\{\cdot\}=\{c,u\} \tag{7}$$

$$P_{ij}^{min}\le p_{ij,t}\le P_{ij}^{max},\quad \forall ij\in\Omega_l,\forall t\in\Omega_t \tag{8}$$

$$Q_{ij}^{min}\le q_{ij,t}\le Q_{ij}^{max},\quad \forall ij\in\Omega_l,\forall t\in\Omega_t \tag{9}$$

$$\left(V_i^{min}\right)^2\le v_{i,t}\le\left(V_i^{max}\right)^2,\quad \forall i\in\Omega_b,\forall t\in\Omega_t \tag{10}$$

$$0\le w_{i,t}^p\le D_{i,t}^p,\ 0\le w_{i,t}^q\le D_{i,t}^q\quad \forall i\in\Omega_b,\forall t\in\Omega_t \tag{11}$$

$$p_{ij,t}^s=G_{ij}\,\theta_{ij,t}^2,\quad \forall ij\in\Omega_l,\forall t\in\Omega_t \tag{12}$$

$$q_{ij,t}^s=-B_{ij}\,\theta_{ij,t}^2,\quad \forall ij\in\Omega_l,\forall t\in\Omega_t \tag{13}$$

$$-P_{c,i}^{ramp}\le p_{i,t}^c-p_{i,t-1}^c\le P_{c,i}^{ramp},\quad \forall i\in\Omega_c,\forall t\in\Omega_t \tag{14}$$

$$-P_{u,i}^{ramp}\le p_{i,t}^u-p_{i,t-1}^u\le P_{u,i}^{ramp},\quad \forall i\in\Omega_u,\forall t\in\Omega_t \tag{15}$$

$$\theta_{ij,t}=\theta_{i,t}-\theta_{j,t},\quad \forall ij\in\Omega_l,\forall t\in\Omega_t \tag{16}$$

$$\theta_{Ref,t}=0,\quad \forall t\in\Omega_t \tag{17}$$

where equations (2) and (3) are power balance constraints at each bus. Equations (4) and (5) represent the linearized power flow on lines. Constraints (6) and (7) set the output ranges of CPGs and GFUs. The lower and upper bounds of power flow are presented in (8)-(9). Constraint (10) gives the boundaries of nodal voltage. The amount of load curtailment is restricted in (11). Equations (12)-(13) are the active and reactive power losses. The ramping limits of CPGs and GFUs are expressed in (14)-(15). Constraint (16) computes the angle difference of each line. Constraint (17) enforces the voltage phase-angle of reference bus to zero.

Here, a linearized OPF model with quadratic power losses [11] is embedded in the operating model. It can be viewed as an extension to the commonly used DC OPF by incorporating reactive power and voltage magnitude in the problem, which narrows the deviation from the real solution with a relatively low computational burden. The approximation process is given in Appendix A.

A steady-state natural gas flow model considering the slow dynamics of line-pack [10], [13] is given as follows.

$$\sum_{im\in\beta_m^+}(f_{im,t}^e-\tau_{im,t})-\sum_{mn\in\beta_m^-}f_{mn,t}^e+\sum_{im\in\gamma_m^+}f_{im,t}^+-\sum_{mn\in\gamma_m^-}f_{mn,t}^-+f_{m,t}^s-f_{m,t}^u=D_{m,t}^g,\quad\forall m\in\Omega_n,\forall t\in\Omega_t \tag{18}$$

$$\frac{f_{mn,t}}{|f_{mn,t}|}=\frac{f_{mn,t-1}}{|f_{mn,t-1}|},\quad\forall mn\in\Omega_p,\forall t\in\Omega_t \tag{19}$$

$$\sum_{mn\in\Omega_p}l_{mn,t}\geq L^{min},\quad\forall t\in\Omega_t \tag{20}$$

$$f_{mn,t}=\left(f_{mn,t}^-+f_{mn,t}^+\right)/2,\quad\forall mn\in\Omega_p,\forall t\in\Omega_t \tag{21}$$

$$W_{mn}f_{mn,t}\left|f_{mn,t}\right|=\pi_{m,t}^2-\pi_{n,t}^2,\quad\forall mn\in\Omega_p,\forall t\in\Omega_t \tag{22}$$

$$f_{mn,t}^--f_{mn,t}^+=l_{mn,t}-l_{mn,t-1},\quad\forall mn\in\Omega_p,\forall t\in\Omega_t \tag{23}$$

$$l_{mn,t}=K_{mn}\left(\pi_{m,t}+\pi_{n,t}\right)/2,\quad\forall mn\in\Omega_p,\forall t\in\Omega_t \tag{24}$$

$$\tau_{mn,t}=\eta_{mn}f_{mn,t}^e,\quad\forall mn\in\Omega_k,\forall t\in\Omega_t \tag{25}$$

$$H_{mn}^{min}\pi_{m,t}\leq\pi_{n,t}\leq H_{mn}^{max}\pi_{m,t},\ \forall mn\in\Omega_k,\forall t\in\Omega_t \tag{26}$$

$$-F_{s,m}^{ramp}\leq f_{m,t}^s-f_{m,t-1}^s\leq F_{s,m}^{ramp},\ \forall m\in\Omega_s,\forall t\in\Omega_t \tag{27}$$

$$p_{i,t}^u=T_m f_{m,t}^u,\quad\forall i,m\in\Omega_u,\forall t\in\Omega_t \tag{28}$$

$$\Pi_m^{min}\leq\pi_{m,t}\leq\Pi_m^{max},\quad\forall m\in\Omega_n,\forall t\in\Omega_t \tag{29}$$

$$-F_{mn}^{max}\leq f_{mn,t}\leq F_{mn}^{max},\quad\forall mn\in\Omega_p,\forall t\in\Omega_t \tag{30}$$

$$0\leq f_{mn,t}^e\leq F_{e,mn}^{max},\quad\forall mn\in\Omega_k,\forall t\in\Omega_t \tag{31}$$

$$0\leq f_{m,t}^u\leq F_{u,m}^{max},\quad\forall m\in\Omega_u,\forall t\in\Omega_t \tag{32}$$

$$F_{s,m}^{min}\leq f_{m,t}^s\leq F_{s,m}^{max},\quad\forall m\in\Omega_s,\forall t\in\Omega_t \tag{33}$$

where constraint (18) denotes the nodal gas balance equation. Equation (19) indicates that the direction of gas flow on each pipeline remains unchanged during the dispatch in short time scale [10]. Constraint (20) sets a minimum value for the line-pack at each time interval. Equation (21) refers to the average gas flow of each pipeline. The nonlinear Weymouth equation of each pipeline is presented in (22). Constraint (23) denotes the change of line-pack. Equation (24) gives the relationship between line-pack and nodal pressures. Equation (25) means that the gas consumption of compressors is a fixed percentage of the gas flowing through them. Constraint (26) corresponds to the limits of compression ratio. The ramping limits of gas suppliers are incorporated in (27). Constraint (28) indicates a linear GFU generation model, which couples the gas and the power systems. In (29)-(33), the ranges of gas pressures, gas flow of pipelines and compressors, active power generation of GFUs, and outputs of gas suppliers are given.

The original OEGF model in (1)-(33) is a non-convex and nonlinear problem because of (12), (13), (19) and (22). Since the objective and constraints are continuous, the original non-convex OEGF model is referred to as **[NLP]**:

$$\textbf{[NLP]}:\ \min\ \ (1),\quad\text{s.t.}\ \ (2)-(33) \tag{34}$$

where the existence of these non-convex constraints makes it hard to solve the problem to its exact solution.

C. Model Reformulation

To standardize the form of non-convexity, **[NLP]** is reformulated as an MIBLP at first. Here, a binary variable λ_{mn} is introduced to indicate the direction of gas flow. Let $\lambda_{mn}=1$ if the direction of the gas flowing through pipeline mn is the same as the forward direction, and $\lambda_{mn}=0$, otherwise. Note that the gas flow direction of each pipeline mn is assumed to remain unchanged during the operation [10]. Based on the Big-M method, for all $mn\in\Omega_p$ and $t\in\Omega_t$, constraint (19) can be reformulated as:

$$-M(1-\lambda_{mn})\leq f_{mn,t}\leq M\lambda_{mn} \tag{35}$$

As to the Weymouth constraint, the absolute value sign in (22) can also be removed. By imposing two sets of auxiliary variables $\sigma_{mn,t}^+$ and $\sigma_{mn,t}^-$, for all $mn\in\Omega_p$ and $t\in\Omega_t$, the Weymouth constraint can be reformulated as:

$$\sigma_{mn,t}^+=\pi_{m,t}+\pi_{n,t} \tag{36}$$

$$-M\lambda_{mn}\leq\sigma_{mn,t}^-+\pi_{m,t}-\pi_{n,t}\leq M\lambda_{mn} \tag{37}$$

$$-M(1-\lambda_{mn})\leq\sigma_{mn,t}^--\pi_{m,t}+\pi_{n,t}\leq M(1-\lambda_{mn}) \tag{38}$$

$$-M(1-\lambda_{mn})\leq\pi_{m,t}-\pi_{n,t}\leq M\lambda_{mn} \tag{39}$$

$$W_{mn}f_{mn,t}^2=\sigma_{mn,t}^+\sigma_{mn,t}^- \tag{40}$$

where $\sigma_{mn,t}^+$ is set to the sum of nodal pressures on pipeline mn, and $\sigma_{mn,t}^-$ represents the absolute value of the pressure difference between the two ends of pipeline mn.

Then, the quadratic equality constraints (12), (13) and (40) can be equivalently rewritten for all $ij\in\Omega_l$, $mn\in\Omega_p$ and $t\in\Omega_t$ as the following convex constraints:

$$p_{ij,t}^s\geq G_{ij}\,\theta_{ij,t}^2,\quad q_{ij,t}^s\geq-B_{ij}\,\theta_{ij,t}^2, \tag{41}$$

$$W_{mn}f_{mn,t}^2\leq\sigma_{mn,t}^+\sigma_{mn,t}^- \tag{42}$$

and the concave constraints:

$$p_{ij,t}^s\leq G_{ij}\,\theta_{ij,t}^2,\quad q_{ij,t}^s\leq-B_{ij}\,\theta_{ij,t}^2, \tag{43}$$

$$W_{mn}f_{mn,t}^2\geq\sigma_{mn,t}^+\sigma_{mn,t}^- \tag{44}$$

where the latter two concave constraints (43)-(44) are further reformulated by imposing three sets of auxiliary variables, i.e., $\sigma_{ij,t}^s$, $\sigma_{mn,t}^w$ and $\sigma_{mn,t}^a$, as follows:

$$p_{ij,t}^s\leq G_{ij}\,\sigma_{ij,t}^s,\quad q_{ij,t}^s\leq-B_{ij}\,\sigma_{ij,t}^s, \tag{45}$$

$$W_{mn}\sigma_{mn,t}^w\geq\sigma_{mn,t}^a \tag{46}$$

$$\sigma_{ij,t}^s=\theta_{ij,t}^2,\ \sigma_{mn,t}^w=f_{mn,t}^2,\ \sigma_{mn,t}^a=\sigma_{mn,t}^+\sigma_{mn,t}^- \tag{47}$$

where the non-convexity of the model is standardized by (47) as bilinear equality constraints in the form of $z=xy$ ($x\equiv y$ is possible). The reformulated model is referred to as **[BLP]** in this paper as:

$$\textbf{[BLP]}:\ \min\ \ (1) \tag{48}$$

$$\text{s.t.}\ \ (2)-(11),(14)-(18),(20)-(21),(23)-(33) \tag{49}$$

$$(35)-(39),(41)-(42),(45)-(47) \tag{50}$$

Note that models **[NLP]** and **[BLP]** are equivalent to each other, and equation (47) is the only non-convex part in **[BLP]** after the reformulation.

D. Matrix Form

For simplicity, the matrix form of the MIBLP-based OEGF model, i.e., the **[BLP]**, is given as follows.

$$\underset{(matrix\ form)}{\textbf{[BLP]}} : \min_{\{\boldsymbol{x}\in\mathbb{Z},\boldsymbol{y}\in\mathbb{R}\}} \boldsymbol{c}^T\boldsymbol{x}+\boldsymbol{d}^T\boldsymbol{y}+\boldsymbol{y}^T\boldsymbol{Q}\boldsymbol{y}+\boldsymbol{e} \quad (51)$$

$$\text{s.t.} \quad \boldsymbol{A}\boldsymbol{x}+\boldsymbol{B}\boldsymbol{y}\le\boldsymbol{f},\ \boldsymbol{L}\le\boldsymbol{y}\le\boldsymbol{U} \quad (52)$$

$$\|\boldsymbol{C}\boldsymbol{y}\|_2\le\boldsymbol{D}\boldsymbol{y} \quad (53)$$

$$\boldsymbol{y}_{r_n}=\boldsymbol{y}_{p_n}\boldsymbol{y}_{q_n},\quad \forall n\in\Delta \quad (54)$$

where $\boldsymbol{x}$ and $\boldsymbol{y}$ are binary and continuous variable vectors. To be specific, $\boldsymbol{x}$ corresponds to the binary variable λ_{mn}, and $\boldsymbol{y}$ refers to $[f^u_{m,t},\ f^s_{m,t},\ f_{mn,t},\ f^-_{mn,t},\ f^+_{mn,t},\ f^e_{mn,t},\ l_{mn,t},\ p^c_{i,t},\ q^c_{i,t},\ p^u_{i,t},\ q^u_{i,t},\ p^s_{ij,t},\ q^s_{ij,t},\ p_{ij,t},\ q_{ij,t},\ v_{i,t},\ w^p_{i,t},\ w^q_{i,t},\ \theta_{i,t},\ \theta_{ij,t},\ \tau_{mn,t},\ \pi_{m,t},\ \sigma^s_{ij,t},\ \sigma^w_{mn,t},\ \sigma^a_{mn,t},\ \sigma^-_{mn,t},\ \sigma^+_{mn,t}]^T$.

Besides, $\boldsymbol{c}$, $\boldsymbol{d}$, $\boldsymbol{Q}$, $\boldsymbol{e}$, $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$, $\boldsymbol{D}$ and $\boldsymbol{f}$ are the coefficient vectors or matrices, $\boldsymbol{L}$ and $\boldsymbol{U}$ denote the vectors of lower and upper bounds. Objective (51) refers to (1). Linear constraints in (52) correspond to (2)-(11), (14)-(18), (20)-(21), (23)-(33), (35)-(39) and (45)-(46). Conic constraint (53) is equivalent to the SOCP constraints in (41)-(42). Constraint (54) represents the bilinear equations in (47). Here, a triple of indices, namely (p_n, q_n, r_n), is introduced for each $n\in\Delta$, where Δ is a set of indices for the variables which appear in (47).

III. Solution Methodology

It is observed that **[NLP]** is a continuous nonlinear model and its MIBLP-based reformulation **[BLP]** is a mixed-integer programming. Although using a general-purpose solver (e.g., Ipopt for **[NLP]**, Gurobi and SCIP for **[BLP]**) might be one of the most obvious choices to solve the problem, this paper is devoted to challenging it by proposing a new solving strategy within the framework of sB&B.

The sB&B method is a divide-and-conquer technique that converts the solution space of the original problem to a rooted search tree. Convex relaxation is performed on each node of the tree. In each iteration, the lower and upper bounds of the optimal value of objective function are compared. If the predefined tolerance is not met, the problem will be partitioned into smaller sub-problems by dividing the feasible regions of continuous variables (spatial branching). Then, the new sub-problems are solved, if required, by recursive partitioning. If a node is proved not to contain the global optimum, then the associated branch in the sB&B search tree can be pruned. The process is continued until the global optimum is found in one of the derived sub-problems [27]. Based on this framework, an improved sB&B algorithm is developed.

A. Convex Relaxation at Each Node

As mentioned above, although sB&B is a global optimization method, the sub-problem at each node of the search tree is still a convex programming (with smaller feasible region). The bilinear relaxation problem at each node is referred to as $\textbf{[SP]}^{(s)}$ in this paper and given by:

$$\textbf{[SP]}^{(s)} : \min\ (51) \quad (55)$$

$$\text{s.t.}\quad (52)-(53),\ \boldsymbol{L}^{(s)}\le\boldsymbol{y}\le\boldsymbol{U}^{(s)} \quad (56)$$

$$\boldsymbol{y}_{r_n}\ge\boldsymbol{L}^{(s)}_{q_n}\boldsymbol{y}_{p_n}+\boldsymbol{L}^{(s)}_{p_n}\boldsymbol{y}_{q_n}-\boldsymbol{L}^{(s)}_{p_n}\boldsymbol{L}^{(s)}_{q_n},\ \forall n\in\Delta \quad (57)$$

$$\boldsymbol{y}_{r_n}\ge\boldsymbol{U}^{(s)}_{q_n}\boldsymbol{y}_{p_n}+\boldsymbol{U}^{(s)}_{p_n}\boldsymbol{y}_{q_n}-\boldsymbol{U}^{(s)}_{p_n}\boldsymbol{U}^{(s)}_{q_n},\ \forall n\in\Delta \quad (58)$$

$$\boldsymbol{y}_{r_n}\le\boldsymbol{U}^{(s)}_{q_n}\boldsymbol{y}_{p_n}+\boldsymbol{L}^{(s)}_{p_n}\boldsymbol{y}_{q_n}-\boldsymbol{L}^{(s)}_{p_n}\boldsymbol{U}^{(s)}_{q_n},\ \forall n\in\Delta \quad (59)$$

$$\boldsymbol{y}_{r_n}\le\boldsymbol{L}^{(s)}_{q_n}\boldsymbol{y}_{p_n}+\boldsymbol{U}^{(s)}_{p_n}\boldsymbol{y}_{q_n}-\boldsymbol{U}^{(s)}_{p_n}\boldsymbol{L}^{(s)}_{q_n},\ \forall n\in\Delta \quad (60)$$

where the superscript (s) denotes the index of the child node created by branching, $\boldsymbol{L}^{(s)}$ and $\boldsymbol{U}^{(s)}$ represent the lower and upper bounds of $\boldsymbol{y}$ at the current node (at the root node, $\boldsymbol{L}^{(0)}$ and $\boldsymbol{U}^{(0)}$ are equal to $\boldsymbol{L}$ and $\boldsymbol{U}$ in (52)). The reformulation-linearization technique (RLT) is adopted to relax the bilinear constraint (54) to RLT inequalities (57)-(60).

It should be emphasized that $\boldsymbol{L}^{(s)}$ and $\boldsymbol{U}^{(s)}$ in each $\textbf{[SP]}^{(s)}$ are changeable (based on the spatial branching strategy in the following subsection), while the remaining coefficients of the model are the same as **[BLP]**.

B. Two-Stage Spatial Branching Strategy

A solution which is feasible for $\textbf{[SP]}^{(s)}$ but does not satisfy constraint (54) is referred to as a bilinear infeasible solution. At each branching node where the current solution is optimal but bilinear infeasible, spatial branching is adopted.

Within the framework of sB&B, spatial branching occurs on the domain of a selected continuous variable. The branching operation generates two new child nodes, each with a smaller domain and a tighter relaxation. For a certain variable $\boldsymbol{y}$ (for instance), the branching rule can be written as:

$$(\boldsymbol{y}_e<\delta)\vee(\boldsymbol{y}_e>\delta) \quad (61)$$

where δ corresponds to the branching point, and e represents the index of branching variable. Since the selection of these two positions has a significant impact on the performance of sB&B algorithm [25], a novel two-stage branching strategy is devised. In the first stage, the best branching point δ for each variable is determined. Then, in the second stage, the variable with the most suitable δ is selected for branching.

1) Stage-1 Branching Point Selection: Let $\boldsymbol{y}^*$ be the bilinear infeasible solution at the current node. Then, the bilinear error of the solution can be computed by:

$$\rho^*_n=|\boldsymbol{y}^*_{r_n}-\boldsymbol{y}^*_{p_n}\cdot\boldsymbol{y}^*_{q_n}|\quad\forall n\in\Delta \quad (62)$$

where ρ^*_n represents the error of the n-th bilinear equation in (54). Without loss of generality, we assume that the error of the n-th bilinear equation violates a predefined threshold. The purpose of stage-1 is to determine the best branching point δ for $\boldsymbol{y}_{p_n}$ (the case for $\boldsymbol{y}_{q_n}$ is analogous).

Since the error ρ^*_n might be greater or less than 0, and the branching point δ could be greater or less than $\boldsymbol{y}^*_{p_n}$, there are four cases that need to be discussed. Here, a positive variable μ is introduced to denote the distance between δ and $\boldsymbol{y}^*_{p_n}$. If $\delta>\boldsymbol{y}^*_{p_n}$, then $\delta=\boldsymbol{y}^*_{p_n}+\mu$, otherwise, $\delta=\boldsymbol{y}^*_{p_n}-\mu$. The key idea of spatial branching is to require that at least one of the RLT inequalities in (57)-(60) be violated by $(\boldsymbol{y}^*_{r_n},\boldsymbol{y}^*_{p_n},\boldsymbol{y}^*_{q_n})$ in both branches [25], so that the bilinear infeasible solution at the parent node can be cut off at its child nodes. With this in mind, we first give the feasible region for μ in the four cases based on an auxiliary variable ξ as follows.

Proposition 1. *For all $\xi > 0$, both downward and upward branches will be violated by the solution $(\boldsymbol{y}^*_{r_n}, \boldsymbol{y}^*_{p_n}, \boldsymbol{y}^*_{q_n})$ if μ takes the following values:*

$$\mu = \begin{cases} -\rho^*_n/(\xi - \boldsymbol{L}^{(s)}_{q_n} + \boldsymbol{y}^*_{q_n}), & \text{if } \rho^*_n < 0,\ \delta < \boldsymbol{y}^*_{p_n} \quad (63) \\ -\rho^*_n/(\xi + \boldsymbol{U}^{(s)}_{q_n} - \boldsymbol{y}^*_{q_n}), & \text{if } \rho^*_n < 0,\ \delta > \boldsymbol{y}^*_{p_n} \quad (64) \\ \rho^*_n/(\xi + \boldsymbol{U}^{(s)}_{q_n} - \boldsymbol{y}^*_{q_n}), & \text{if } \rho^*_n > 0,\ \delta < \boldsymbol{y}^*_{p_n} \quad (65) \\ \rho^*_n/(\xi - \boldsymbol{L}^{(s)}_{q_n} + \boldsymbol{y}^*_{q_n}), & \text{if } \rho^*_n > 0,\ \delta > \boldsymbol{y}^*_{p_n} \quad (66) \end{cases}$$

Proof. Since the proofs of (63)-(66) are similar, we take (63) as an example. By definition, the branching point for $\boldsymbol{y}_{p_n}$ is given by:

$$\delta = \boldsymbol{y}^*_{p_n} - \mu = \boldsymbol{y}^*_{p_n} + \rho^*_n/(\xi - \boldsymbol{L}^{(s)}_{q_n} + \boldsymbol{y}^*_{q_n}) \quad (67)$$

and it is obvious that $\boldsymbol{y}^*_{p_n}$ is violated in the downward branch because the upper bound of $\boldsymbol{y}_{p_n}$ is narrowed to $\boldsymbol{y}^*_{p_n} - \mu$. As to the upward branch, we suppose for a contradiction that the solution $(\boldsymbol{y}^*_{r_n}, \boldsymbol{y}^*_{p_n}, \boldsymbol{y}^*_{q_n})$ is in the solution set. Then, the RLT inequality (57) for the upward branch can be written as:

$$\boldsymbol{y}_{r_n} - \boldsymbol{L}^{(s)}_{q_n}\boldsymbol{y}_{p_n} - (\boldsymbol{y}^*_{p_n} - \mu)\boldsymbol{y}_{q_n} + \boldsymbol{L}^{(s)}_{q_n}(\boldsymbol{y}^*_{p_n} - \mu) \geq 0 \quad (68)$$

substitute $(\boldsymbol{y}^*_{r_n}, \boldsymbol{y}^*_{p_n}, \boldsymbol{y}^*_{q_n})$ into (68), we obtain:

$$\boldsymbol{y}^*_{r_n} - \boldsymbol{L}^{(s)}_{q_n}\boldsymbol{y}^*_{p_n} - \boldsymbol{y}^*_{p_n}\boldsymbol{y}^*_{q_n} + \mu\boldsymbol{y}^*_{q_n} + \boldsymbol{L}^{(s)}_{q_n}\boldsymbol{y}^*_{p_n} - \boldsymbol{L}^{(s)}_{q_n}\mu \geq 0 \quad (69)$$

replace $\boldsymbol{y}^*_{r_n} - \boldsymbol{y}^*_{p_n}\boldsymbol{y}^*_{q_n}$ with ρ^*_n, then, simplify the expression as follows:

$$\rho^*_n + \mu(\boldsymbol{y}^*_{q_n} - \boldsymbol{L}^{(s)}_{q_n}) \geq 0 \quad (70)$$

substitute (63) into (70), we have:

$$\rho^*_n \geq \rho^*_n(\boldsymbol{y}^*_{q_n} - \boldsymbol{L}^{(s)}_{q_n})/(\xi + \boldsymbol{y}^*_{q_n} - \boldsymbol{L}^{(s)}_{q_n}) \quad (71)$$

note that $\rho^*_n < 0$ is a pre-condition of (63), and it is obvious that $\boldsymbol{y}^*_{q_n}$ is greater than $\boldsymbol{L}^{(s)}_{q_n}$, therefore, the inequality in (71) is a contradiction, which completes the proof. □

Based on **Proposition 1**, we yield the feasible region for δ without the use of auxiliary variable ξ as follows.

Proposition 2. *For the bilinear infeasible $(\boldsymbol{y}^*_{r_n}, \boldsymbol{y}^*_{p_n}, \boldsymbol{y}^*_{q_n})$, the feasible region Ψ for the branching point δ is given by:*

$$\Psi = \left\{\delta \in \mathbb{R} : \boldsymbol{y}^*_{p_n} - \underline{\mu} < \delta < \boldsymbol{y}^*_{p_n} + \overline{\mu}\right\} \quad (72)$$

where

$$(\underline{\mu},\ \overline{\mu}) = \begin{cases} \left(\dfrac{-\rho^*_n}{\boldsymbol{y}^*_{q_n} - \boldsymbol{L}^{(s)}_{q_n}},\ \dfrac{-\rho^*_n}{\boldsymbol{U}^{(s)}_{q_n} - \boldsymbol{y}^*_{q_n}}\right), & \text{if } \rho^*_n < 0 \quad (73) \\ \left(\dfrac{\rho^*_n}{\boldsymbol{U}^{(s)}_{q_n} - \boldsymbol{y}^*_{q_n}},\ \dfrac{\rho^*_n}{\boldsymbol{y}^*_{q_n} - \boldsymbol{L}^{(s)}_{q_n}}\right), & \text{if } \rho^*_n > 0 \quad (74) \end{cases}$$

Proof. Since the proofs of (73) and (74) are similar, we take (73) as an illustration. Based on **Proposition 1**, we see that μ is a monotonic function of ξ. If $\rho^*_n < 0$ and $\delta < \boldsymbol{y}^*_{p_n}$ hold, for all $\xi > 0$, the infimum and supremum of μ are given as follows:

$$\mu > \lim_{\xi \to \infty}(63) = 0,\ \mu < \lim_{\xi \to 0}(63) = -\rho^*_n/(\boldsymbol{y}^*_{q_n} - \boldsymbol{L}^{(s)}_{q_n}) \quad (75)$$

which yields the range of δ as $\Psi_+ = \left\{\boldsymbol{y}^*_{p_n} - \underline{\mu} < \delta < \boldsymbol{y}^*_{p_n}\right\}$. Similarly, in the case that $\rho^*_n < 0$ and $\delta > \boldsymbol{y}^*_{p_n}$ hold, the range of δ can be given by $\Psi_- = \left\{\boldsymbol{y}^*_{p_n} < \delta < \boldsymbol{y}^*_{p_n} + \overline{\mu}\right\}$ based on (64). As $\boldsymbol{y}^*_{p_n}$ is also feasible for branching [25], we have:

$$\Psi = \Psi_+ \cup \left\{\boldsymbol{y}^*_{p_n}\right\} \cup \Psi_- = \left\{\boldsymbol{y}^*_{p_n} - \underline{\mu} < \delta < \boldsymbol{y}^*_{p_n} + \overline{\mu}\right\} \quad (76)$$

which completes the proof of (73). The proof of (74) can be analogized from the analysis above. □

Based on **Proposition 2**, to balance the complexity of the two child nodes, the branching point δ for variable $\boldsymbol{y}_{p_n}$ can be determined within its feasible region Ψ as:

$$\delta = \arg\min_{y \in \Psi} \left| y - \frac{1}{2}(\boldsymbol{L}^{(s)}_{p_n} + \boldsymbol{U}^{(s)}_{p_n}) \right| \quad (77)$$

whose core idea is to choose the point in Ψ that is closest to the midpoint of the domain of $\boldsymbol{y}_{p_n}$.

2) Stage-2 Branching Variable Selection: After obtaining the branching points for all potential branching variables, the next step is to choose the best one to branch. Let Φ be a set branching variables at the current node, the traditional sB&B algorithm adopts reliability branching rule [28] to calculate a score for each candidate $\boldsymbol{y}_e \in \Phi$. However, the rule relies on the information of bilinear error (62) only, the information of SOCP relaxation error is discarded.

To address the issues, a novel branching variable selection strategy is proposed to incorporate the SOCP relaxation error into the selection of $\boldsymbol{y}_e$. First, a $n_c \times n_y$ matrix $\boldsymbol{E}$ is defined, where n_c is set to the number of conic relaxation constraints in **[P]**, and n_y is set to the size of the continuous variable $\boldsymbol{y}$. The elements of $\boldsymbol{E}$ are defined as:

$$\boldsymbol{E}_{ij} = \begin{cases} 1, & \text{if } \boldsymbol{C}_{ij} \neq 0; \\ 0, & \text{if } \boldsymbol{C}_{ij} = 0. \end{cases} \quad (78)$$

where $\boldsymbol{E}_{ij} = 1$ represents the variable $\boldsymbol{y}_j$ appears in the i-th conic relaxation constraint in **[P]**, and $\boldsymbol{E}_{ij} = 0$, otherwise.

Next, for the bilinear infeasible solution $\boldsymbol{y}^*$ at the current node, a penalty vector ζ is defined as:

$$\zeta = [\zeta_1, \zeta_2, ..., \zeta_{n_y}]^{\mathrm{T}} = \boldsymbol{E}^{\mathrm{T}} \cdot (\boldsymbol{D}\boldsymbol{y}^* - \|\boldsymbol{C}\boldsymbol{y}^*\|_2) \quad (79)$$

where the size of ζ is equal to n_y. As can be seen, for each optional branching variable $\boldsymbol{y}_e$, the corresponding ζ_e denotes the sum of the SOCP relaxation errors where $\boldsymbol{y}_e$ is involved in. Therefore, the penalty vector ζ can be used to improve the performance of the traditional reliability branching rule.

Specifically, for each candidate $\boldsymbol{y}_e \in \Phi$, let ϕ^-_e and ϕ^+_e be the pseudo-costs for downward and upward branching on $\boldsymbol{y}_e$ (which can be directly obtained from solvers). The reliability branching is adopted first to give a basic score $\widetilde{s_e}$ for $\boldsymbol{y}_e$ as follows, where ω_1 and ω_2 are two weight coefficients [28].

$$\widetilde{s_e} = \omega_1 \max\{\phi^-_e, \phi^+_e\} + \omega_2 \min\{\phi^-_e, \phi^+_e\} \quad (80)$$

Then, the SOCP relaxation error is further incorporated to modify the selection of branching variables. The core idea is to give priority to the variables which are active in the conic constraints with larger relaxation errors. Hence, an activation function, $\tanh(\zeta_e)$, is designed in this paper. The final score of branching on $\boldsymbol{y}_e$ can be computed by:

$$s_e = \widetilde{s_e} \cdot \tanh(\zeta_e) \quad (81)$$

The procedure of the proposed two-stage spatial branching strategy is introduced in **Algorithm 1**.

Algorithm 1: The Two-Stage Spatial Branching

Define $s \leftarrow 0$, $\Phi \leftarrow \varnothing$, $\boldsymbol{y}_{br} \leftarrow \boldsymbol{y}_0$, $\delta_{br} \leftarrow 0$;
for each $n \in \Delta$ with $|\boldsymbol{y}^*_{r_n} - \boldsymbol{y}^*_{p_n}\boldsymbol{y}^*_{q_n}| \geq \varepsilon$ **do**
  Get the value of δ_{p_n} by (77);
  Get the value of δ_{q_n} by (77) with p_n, q_n swapped;
  Update $\Phi \leftarrow \Phi \cup \{\boldsymbol{y}_{p_n}, \boldsymbol{y}_{q_n}\}$;
for each $\boldsymbol{y}_e \in \Phi$ **do**
  Compute the branching score s_e by (81);
  if $s_e \geq s$ **then**
    Update $\boldsymbol{y}_{br} \leftarrow \boldsymbol{y}_e$, $\delta_{br} \leftarrow \delta_e$, $s \leftarrow s_e$;
Return the branching position $\boldsymbol{y}_{br}$ and δ_{br}.

Algorithm 2: The Improved sB&B Algorithm

1 Set the thresholds of relaxation error ε and optimality gap ϵ based on user requirements;
2 Set $k \leftarrow 0$, $lb \leftarrow -\infty$, $ub \leftarrow +\infty$;
3 Initialize sets $N \leftarrow \{\textbf{[SP]}^{(0)}\}$ and $Z \leftarrow \{lb\}$;
4 **if** an incumbent $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ is available **then**
5 Solve $\textbf{[SP]}^{(0)}$ to get the solution $(\boldsymbol{x}^{(0)}, \boldsymbol{y}^{(0)})$ and the objective value $z^{(0)}$, update $lb \leftarrow z^{(0)}$;
6 Update ub by (51) using the incumbent $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$;
7 **else**
8 Initialize the incumbent by $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}) \leftarrow (\boldsymbol{0}, \boldsymbol{0})$;
9 **while** $(ub - lb)/ub \geq \epsilon$ and $N \neq \varnothing$ **do**
10 Pick out the $\textbf{[SP]}^{(s)}$ from N with the minimal $z^{(s)}$;
11 Update $N \leftarrow N \setminus \{\textbf{[SP]}^{(s)}\}$, $Z \leftarrow Z \setminus \{z^{(s)}\}$;
12 Use **Algorithm 1** to get the branching position;
13 Branch on $(\boldsymbol{y}_{br} < \delta_{br}) \vee (\boldsymbol{y}_{br} > \delta_{br})$ and create child nodes $\textbf{[SP]}^{(k+1)}$ and $\textbf{[SP]}^{(k+2)}$;
14 **for** each node $\textbf{[SP]}^{(s)}$ in $\{\textbf{[SP]}^{(k+1)}, \textbf{[SP]}^{(k+2)}\}$ **do**
15 **if** $\textbf{[SP]}^{(s)}$ is feasible **then**
16 Solve $\textbf{[SP]}^{(s)}$ to get its solution $(\boldsymbol{x}^{(s)}, \boldsymbol{y}^{(s)})$ and the objective value $z^{(s)}$;
17 **if** the solution is bilinear feasible **then**
18 **if** $z^{(s)} \leq ub$ **then**
19 $ub \leftarrow z^{(s)}$, $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}}) \leftarrow (\boldsymbol{x}^{(s)}, \boldsymbol{y}^{(s)})$;
20 **else**
21 $N \leftarrow N \cup \{\textbf{[SP]}^{(s)}\}$, $Z \leftarrow Z \cup \{z^{(s)}\}$;
22 Update $N \leftarrow N \setminus \{\textbf{[SP]}^{(s)} \mid z^{(s)} \geq ub,\ \forall s\}$, and $Z \leftarrow Z \setminus \{z^{(s)} \mid z^{(s)} \geq ub,\ \forall s\}$;
23 Update lb with the minimal $z^{(s)} \in Z$, $k \leftarrow k + 2$;
24 Return the best solution $(\hat{\boldsymbol{x}}, \hat{\boldsymbol{y}})$ and the minimum objective value, i.e. the upper bound ub.

C. Procedure of the Improved sB&B Algorithm

Based on the analysis above, the procedure of the proposed improved sB&B algorithm is detailed in **Algorithm 2**, where **Algorithm 1** is performed at all branching nodes.

The first line of the algorithm provides a user interface to control the accuracy levels of the solution (as a fully optimal and tight solution may not be a necessity for some real-world applications). Line-2 sets the parameters of counter k, lower bound lb, and upper bound ub. Line-3 refers to the node and value sets used in the sB&B algorithm. Line-4 to Line-8 are designed for root node processing if an incumbent (defined as the best solution found in the search) is available for warm-start. This step will be discussed in the following subsection. The main iteration of the sB&B algorithm starts at Line-9. In Line-10 and Line-11, we pick out the node with the minimal objective value first. Then, the two-stage spatial branching is performed (Line-12 to Line-13). Next, in Line-14 to Line-21, the new child sub-problems are solved. Once the feasibility is checked, the node which cannot provide a better solution will be added to the node set. Otherwise, the incumbent is updated. Line-22 prunes the node whose local optimum is worse than the incumbent, and Line-23 updates the new lower bound. If the optimality tolerance is not met, the algorithm will start the next iteration. Otherwise, Line-24 terminates the solving.

D. Modern Implementation for the sB&B Algorithm

Although the procedures of the sB&B in **Algorithm 2** can be implemented programmatically, an obsolete programming approach may still cause slow convergence. The main reason is that, the solver treats each sub-problem $\textbf{[SP]}^{(s)}$ as a brand new optimization task, even though these sub-problems differ on only a few coefficients. Solving the problems with similar structures may contain considerable rework.

In this paper, a modern implementation approach is developed for the improved sB&B algorithm. As suggested in our previous work [29], the support for callbacks enables modern solvers to integrate customizable functionalities when dealing with mixed-integer programs. Therefore, the core idea of the implementation is to embed **Algorithm 2** into the traditional branch-and-cut algorithm of modern solvers to avoid solving all sub-problems independently. The implementation is based on a commercial solver CPLEX 20.1.0 [30], and **[BLP]** is the only problem delivered to the solver.

1) Callbacks: Since CPLEX is not an MIBLP solver, it is not aware of the existence of bilinear constraints (54), which means that the built-in branch-and-cut algorithm would not branch on any integer-feasible solution (even if it is a bilinear infeasible solution). Therefore, three specific callback classes are registered, namely "heuristics callback", "incumbent callback" and "branch callback".

The heuristics callback is used at the root, where Line-4 to Line-8 are built into this callback to set a feasible solution (if exists) for warm-start. Each time an integer-feasible solution is found, the incumbent callback will be invoked. Since it is the only chance to discard a solution for whatever the reason is, Line-15 to Line-21 are embedded in this callback to prevent any bilinear infeasible incumbent from updating. The branch callback is designed for providing branching decisions, which corresponds to Line-12 to Line-13. In our implementation, if the current node is not integer-feasible (which means CPLEX knows how to branch), we just let CPLEX branch and add its internal cuts as usual. Otherwise, the novel two-stage spatial branching strategy is adopted to overwrite the default one. As to the rest steps in **Algorithm 2**, we let CPLEX implement them automatically. After setting all these callback functions, the solver only needs to be called once.

2) Parameters of Solver: Due to the presence of callbacks, CPLEX needs to work on the original variable space without any dual pre-solve reductions. Therefore, three preprocessing parameters called "reduce", "aggregator" and "reformulation" are set to 1, 0, and 0, respectively. Besides, to prevent CPLEX from spending a significant amount of time performing computation at the root, the node heuristic needs to be turned off by setting the parameter "heuristicfreq" to -1.

3) Root Node Processing: The motivation for this process is based on empirical observations that mixed-integer models are solved fast once all integer variables are fixed to constant [26]. Therefore, a root node processing approach is proposed to find a feasible (could be sub-optimal, as is usually the case) solution at the root by fixing all binary variables λ_{mn} (which represent the gas flow directions) based on historical data, or simply let all $\lambda_{mn} = 1$ as default. Solve this modified model using **Algorithm 2** gives an incumbent at the root. Then, we can set these binary variables free and perform **Algorithm 2** again to obtain the exact solution. Since the incumbent at the root could be used as a criterion to prune fathomed nodes, the size of the sB&B search tree can be greatly reduced.

IV. Case Study

In this paper, the performance of the proposed algorithm is tested on two IEGSs. The first case study is implemented on a system which consists of a 39-bus power system and a 11-node gas system to illustrate the computational performance. Then, a large system which is composed of a South Carolina 500-bus power system and a Greek 134-node gas system is adopted to validate the scalability. The detailed parameters of the two systems are modified from the dataset of Matpower [31] and Gaslib [32] and can be downloaded from [33]. All tests are coded in Python 3.8 and tested on a PC with AMD Ryzen 3900X CPU and 32 GB of RAM.

To test the performance of our improved sB&B algorithm in solving the non-convex OEGF model, a total of 12 solving strategies are compared:

- M1: a Taylor series expansion based approximation [6]
- M2: a PWL-based approximation [5]
- M3: a conic relaxation without tightening strategies [19]
- M4: a tightened formulation based on convex hull [10]
- M5: a tight conic relaxation based on CCP [8]
- M6: a global optimization method based on AMP [26]
- M7: a commercial solver Gurobi 9.1.2 [21]
- M8: a continuous nonlinear solver Ipopt 3.14.0 [16]
- M9: a general-purpose solver SCIP 7.0.2 [22]
- M10: an sB&B with a guided 2-way partitioning [24]
- M11: an sB&B with a shifted branching strategy [25]
- M12: the proposed improved sB&B algorithm

where the characteristics of M1-M12 are given in Table I. The marks in the first and second lines denote their capabilities to guarantee the feasibility and optimality of solutions, and the types of problems solved by M1-M12 are also specified. It should be noted that **[NLP]** and **[BLP]** are equivalent to each other, which means that they share the same global optimum. However, not all of these solving strategies are applicable for non-convex optimization. Therefore, approximations are made for M1-M4, where M1 and M2 are modeled as mixed-integer linear programs, while M3 and M4 are formulated as mixed-integer SOCP problems. As to M5-M10, no approximation is made because their algorithms have the ability to cope with the non-convex constraints. The solvers used by M1-M12 are listed in the last line as well, and M8 (Ipopt) is adopted as the benchmark for the rest methods.

To quantify the feasibility and optimality of solutions, two indices, i.e., E_f and E_o, are introduced correspondingly. For each obtained solution and its objective value, E_f is given by the maximal imbalance among (12), (13) and (22), and E_o is defined as the relative gap between the objective value of the benchmark M8 and the obtained one.

Since a fully feasible and optimal OEGF solution may not be a necessity for real-world applications, the performance is tested under three accuracy levels, namely low, medium and high. For each level, the thresholds ε and ϵ are set to (2.5%, 2.5%), (1%, 1%) and (0.5%, 0.5%), respectively. For M1-M4 whose feasibility and optimality are uncontrollable, the above thresholds are set free and are used to check E_f and E_o after the solution is obtained. For M5, only the threshold ε is predefined. For M6-M12, the parameters of their algorithms are adjusted accordingly based on the values of ε and ϵ.

A. Single-Period OEGF calculation

Solving simple problems is a necessary first step to test the performance of our sB&B algorithm. In this section, a single-period OEGF problem, which can be easily derived from the original **[NLP]** (or **[BLP]**) by setting $|\Omega_t| = 1$, is adopted. As to those time-related constraints, i.e. (14), (15), (23) and (27), we simply discard them. A total of 24 time intervals are tested for both IEGSs at each accuracy level. The runtime limit is set to 3600 seconds.

1) 39-bus-11-node Testing IEGS: In the testing IEGS, the effectiveness of the proposed algorithm is demonstrated first. Table II illustrates the average values of E_f and E_o at each accuracy level. As can be observed, M1-M4, which are four approximation-based methods, cannot meet the requirements of high accuracy level (where M1 even exceeds the thresholds of medium accuracy level). Based on the tightening strategies of CCP and AMP, M5 and M6 ensure the feasibility of their solutions at each accuracy level. The general-purpose solvers M7-M9 are verified to be trustworthy approaches to deal with non-convex OEGF models due to their high-quality solutions. Two traditional sB&B algorithms M10-M11 also successfully solve the problem to an acceptable tolerance. As to M12, the solutions obtained by our improved sB&B algorithm are very close to those of M8 (the benchmark Ipopt), and the relative error of the objective function value is no more than 0.02% at each accuracy level, which demonstrates that our strategy is able to effectively solve the non-convex OEGF model.

A further comparison on the solvability of M1-M12 on 24 time intervals at all accuracy levels is conducted then. Here, solvability is defined as the number of time intervals that have been successfully solved. To be specific, if the best values of E_f and E_o a method can get within the runtime limit do not violate the thresholds ε and ϵ, the corresponding time interval

TABLE I
COMPARISON OF THE CHARACTERISTICS OF SOLVING STRATEGIES M1-M12

Method	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
Feasibility	×	×	×	×	✓	✓	✓	✓	✓	✓	✓	✓
Optimality	×	×	×	×	×	✓	✓	✓	✓	✓	✓	✓
Problem Type	**[NLP]**	**[NLP]**	**[BLP]**	**[BLP]**	**[BLP]**	**[BLP]**	**[BLP]**	**[NLP]**	**[BLP]**	**[BLP]**	**[BLP]**	**[BLP]**
Approximation	Linear	Linear	Conic	L&C	No	No	No	No	No	No	No	No
Solver	CPLEX	CPLEX	CPLEX	CPLEX	Gurobi	Gurobi	Gurobi	Ipopt	SCIP	Gurobi	Gurobi	CPLEX

[1] L&C means that both linear and conic approximations are used.

TABLE II
EXACTNESS OF M1-M12 IN THE SINGLE-PERIOD OEGF CALCULATION OF 39-BUS-11-NODE TESTING IEGS (ON AVERAGE)

Accuracy	Low		Medium		High	
	E_f (%)	E_o (%)	E_f (%)	E_o (%)	E_f (%)	E_o (%)
M1	1.99	↓ 2.47	1.99	↓ 2.47	1.99	↓ 2.48
M2	0.42	↑ 0.50	0.42	↑ 0.43	0.39	↑ 0.35
M3	0.53	↓ 0.01	0.53	↓ 0.01	0.53	↓ 0.02
M4	0.52	↓ 0.01	0.52	↓ 0.01	0.52	↓ 0.02
M5	0.12	↑ 0.02	0.04	↑ 0.01	0.02	↑ 0.01
M6	0.43	↑ 0.03	0.32	↑ 0.05	0.15	↑ 0.03
M7	0.47	↑ 0.03	0.32	↑ 0.05	0.10	↑ 0.04
M8	0.16	benchmark	0.11	benchmark	0.06	benchmark
M9	0.19	↑ 0.05	0.11	↑ 0.05	0.09	↑ 0.02
M10	0.14	↑ 0.06	0.10	↑ 0.01	0.03	↑ 0.02
M11	0.15	↑ 0.06	0.08	↑ 0.02	0.04	↑ 0.02
M12	0.15	↑ 0.02	0.10	↑ 0.01	0.06	↑ 0.01

[1] The symbol "↓ (↑)" denotes that the corresponding value is less (greater) than the benchmark.

TABLE III
SOLVABILITY OF M1-M12 IN THE SINGLE-PERIOD OEGF CALCULATION OF 39-BUS-11-NODE TESTING IEGS

Method		M1	M2	M3	M4	M5	M6
Accuracy Level	Low	21 / 3	24 / 0	21 / 3	21 / 3	24 / 0	24 / 0
	Medium	0 / 24	23 / 1	21 / 3	21 / 3	24 / 0	24 / 0
	High	0 / 24	13 / 11	20 / 4	20 / 4	24 / 0	24 / 0
Method		M7	M8	M9	M10	M11	M12
Accuracy Level	Low	24 / 0	24 / 0	24 / 0	24 / 0	24 / 0	24 / 0
	Medium	24 / 0	24 / 0	24 / 0	24 / 0	24 / 0	24 / 0
	High	24 / 0	24 / 0	24 / 0	24 / 0	24 / 0	24 / 0

[1] The number on the left-hand (right-hand) side of "/" denotes the number of solved (unsolved) instances.

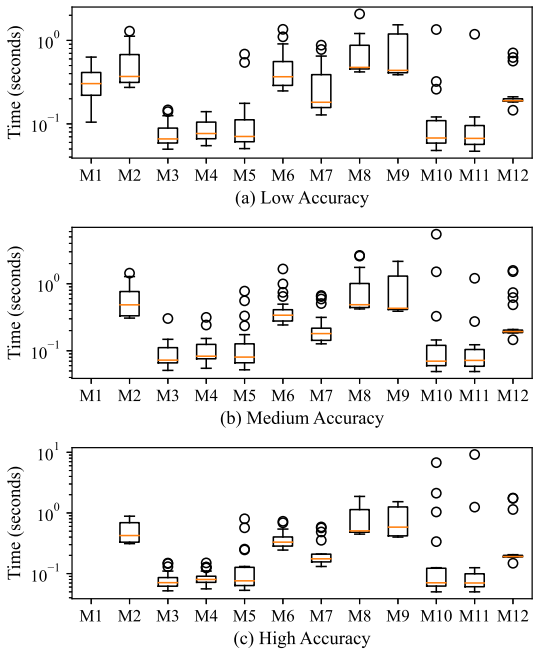

Fig. 1. Box-plot of computing time of M1-M12 in the single-period OEGF calculation of 39-Bus-11-Node testing IEGS.

will be marked as successfully solved. The results are shown in Table III. As can be seen, M5-M12 (including our sB&B algorithm) solve all 24 single-period OEGF problems at each accuracy level, which further verifies the effectiveness of the proposed sB&B algorithm.

Fig. 1 compares the runtime of M1-M12 for solving all 24 instances at each accuracy level (only the successfully solved ones are taken into account). The computing time is depicted in box-plot. The box extends from the lower to upper quartile values of the data, with a line at the median. The whiskers on the two sides of the box show the range of the data. Besides, outlier points are given in circles. Note that the vertical axis is given in logarithmic coordinate. As can be seen, the runtime of M1-M12 ranges from 10^{-1} seconds to 10 seconds. Among them, we leave M1-M4 out of the discussion because of their low performance in solvability. The proposed sB&B algorithm (M12) has a similar efficiency to Gurobi (M7), and is slightly faster than Ipopt (M8) and SCIP (M9). Although M5, M10-M11 seem to be faster than ours, the difference in computing time is on the order of 10^{-1} seconds.

It can be summarized from the comparisons above that our sB&B algorithm is competitive with general-purpose solvers in computational efficiency while ensuring the feasibility and optimality of the solutions. For a small test case of this size, the computing time is on the order of 0.1 seconds. A detailed log file for all data above can be found in [33].

2) 500-Bus-and-134-Node Large-Scale IEGS: To verify the scalability of the proposed algorithm, a large IEGS modified from real-world parameters is adopted. Similar to the testing case above, the effectiveness, solvability and computing time of M1-M12 are compared as follows.

Table IV shows the effectiveness of all 12 methods at each accuracy level, where the average values of E_f and E_o are

TABLE IV
EXACTNESS OF M1-M12 IN THE SINGLE-PERIOD OEGF CALCULATION OF 500-BUS-134-NODE LARGE-SCALE IEGS (ON AVERAGE)

Accuracy	Low		Medium		High	
	E_f (%)	E_o (%)	E_f (%)	E_o (%)	E_f (%)	E_o (%)
M1	1.56	↓ 1.86	1.56	↓ 2.28	1.56	↓ 2.39
M2	2.18	↑ 1.31	2.18	↑ 0.93	2.18	↑ 0.76
M3	0.62	↓ 1.46	0.62	↓ 0.69	0.62	↓ 0.44
M4	1.09	↓ 1.27	1.09	↓ 0.70	1.09	↓ 0.45
M5	0.27	↑ 0.74	0.15	↑ 0.26	0.17	↑ 0.10
M6	0.37	↑ 0.82	0.38	↑ 0.25	0.24	↑ 0.21
M7	0.71	↑ 0.68	0.39	↑ 0.32	0.36	↑ 0.09
M8	0.05	benchmark	0.08	benchmark	0.06	benchmark
M9	1.60	↑ 0.17	0.66	↑ 0.11	0.33	↑ 0.04
M10	0.26	↑ 0.29	0.12	↑ 0.14	0.13	↑ 0.08
M11	0.12	↑ 0.28	0.12	↑ 0.11	0.12	↑ 0.07
M12	0.15	↑ 0.06	0.12	↑ 0.09	0.10	↑ 0.08

[1] The symbol "↓ (↑)" denotes that the corresponding value is less (greater) than the benchmark.

TABLE V
SOLVABILITY OF M1-M12 IN THE SINGLE-PERIOD OEGF CALCULATION OF 500-BUS-134-NODE LARGE-SCALE IEGS

Method		M1	M2	M3	M4	M5	M6
Accuracy Level	Low	24 / 0	20 / 4	20 / 4	17 / 7	23 / 1	23 / 1
	Medium	0 / 24	0 / 24	15 / 9	9 / 15	22 / 2	23 / 1
	High	0 / 24	0 / 24	10 / 14	5 / 19	21 / 3	14 / 10
Method		M7	M8	M9	M10	M11	M12
Accuracy Level	Low	24 / 0	24 / 0	24 / 0	24 / 0	24 / 0	24 / 0
	Medium	24 / 0	24 / 0	24 / 0	24 / 0	24 / 0	24 / 0
	High	24 / 0	24 / 0	23 / 1	23 / 1	23 / 1	24 / 0

[1] The number on the left-hand (right-hand) side of "/" denotes the number of solved (unsolved) instances.

listed to illustrate the feasibility and optimality of the obtained solutions. As can be seen, M1-M4 fail to solve the instances at medium and high accuracy levels, which again shows that approximation-based methods are not reliable for solving the non-convex OEGF models. As to M5-M12, numerical results show that their indices of feasibility and optimality satisfy the thresholds of ε and ϵ. Take our improved sB&B algorithm at the high accuracy level as an example, its imbalance error E_f is only 0.1%, which is lower than the threshold ($\varepsilon = 0.5\%$), and its objective value is only 0.08% higher than that of M8, which is also lower than the threshold of ($\epsilon = 0.5\%$).

Since the average values given above are still insufficient to fully demonstrate the solving status at each time interval, the solvability of M1-M12 at each accuracy level is further listed in Table V. It can be seen that most methods do not perform as well on the large-scale IEGS as they do on test cases. The proposed sB&B algorithm (M12), along with M7 and M8, is the only three methods that successfully solve all 3×24 time intervals. Besides, M9-M11 are another three well-performed solving strategies with only one unsolved instance at the high accuracy level. By comparison, M5-M6 are not as effective as M7-M12, because their number of unsolved time intervals is relatively large, especially for M6 at the high accuracy level.

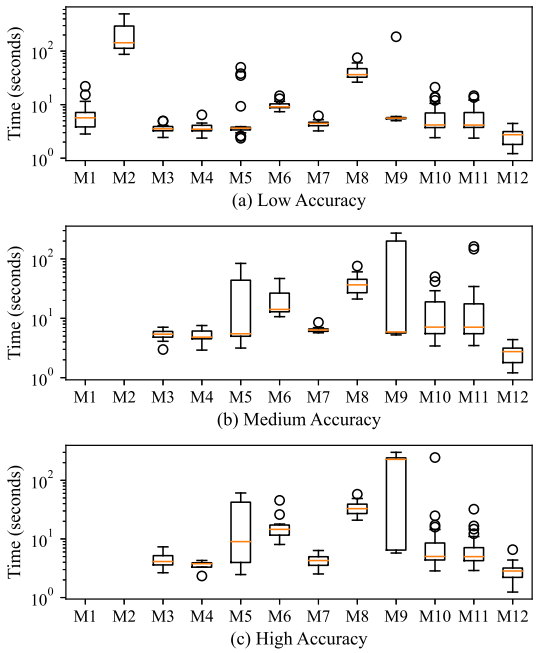

Fig. 2. Box-plot of computing time of M1-M12 in the single-period OEGF calculation of 500-Bus-134-Node large-scale IEGS.

As to M1-M4, they show poor performances in dealing with the large-scale system. The comparison above verifies that the improved sB&B algorithm has the ability to solve large non-convex OEGF problems to their exact solutions.

The computing time of M1-M12 is depicted in Fig. 2 (only the successfully solved instances are drawn). The meaning of the elements in the figure is the same as in Fig. 1. As can be observed, M1 and M2 can only solve the problems at the low accuracy level, leaving all the instances at medium and high levels unsolved. For M3 and M4, although they do solve some of the instances within 10 seconds, the success is achieved by chance that the relaxed solutions happen to be feasible. For the instances at the low accuracy level, the computing time of M5 is around 5 seconds. However, the time increases rapidly with the improvement of accuracy level. The results of M6 (AMP) indicate that the newly-added binary variables in its piecewise McCormick relaxation bring additional computational burden to the solving process. As to the general-purpose solvers M7-M9, Gurobi (M7) has the highest efficiency, with an average solving time of about 5 seconds. At each accuracy level, the computing time of Ipopt (M8) stabilizes at about 30 seconds, while the time of SCIP (M9) fluctuates greatly. By the use of the guided 2-way and the shifted branching strategies, M10-M11 also successfully solve most of the time intervals within 10 seconds. By comparison, our improved sB&B algorithm (M12) takes the shortest time (around 3 seconds on average) to solve each instance.

Based on the analysis of large-scale IEGS above, it can be seen that the proposed sB&B algorithm achieves even better performance compared to the test case. With the expansion of the system scale, the superiority of the proposed algorithm in computational efficiency becomes more obvious. For detailed results, please refer to [33].

TABLE VI
COMPARISON OF M1-M12 FOR THE DAY-AHEAD OPERATION OF 39-BUS-11-NODE TESTING IEGS

Method	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
E_f (%)	—	—	—	—	0.000	0.550	0.450	0.016	0.311	0.199	—	0.048
E_o (%)	—	—	—	—	↑ 0.29	↑ 0.07	↑ 0.01	benchmark	↑ 0.00	↑ 0.07	—	↑ 0.01
Time (s)	—	—	—	—	226.59	432.82	187.20	42.02	170.23	2791.08	—	28.98

[1] The symbol "—" indicates that the corresponding method cannot give a qualified solution within a limited runtime.
[2] The symbol "↑" denotes that the corresponding value is greater than the benchmark.

TABLE VII
COMPARISON OF M1-M12 FOR THE DAY-AHEAD OPERATION OF 500-BUS-134-NODE LARGE-SCALE IEGS

Method	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
E_f (%)	—	—	—	—	—	—	—	0.059	—	—	—	0.754
E_o (%)	—	—	—	—	—	—	—	benchmark	—	—	—	↑ 0.03
Time (s)	—	—	—	—	—	—	—	745.33	—	—	—	516.64

[1] The symbol "—" indicates that the corresponding method cannot give a qualified solution within a limited runtime.
[2] The symbol "↑" denotes that the corresponding value is greater than the benchmark.

B. Day-Ahead IEGS Operation

Here, the original day-ahead IEGS operation problem with an hourly electricity-and-gas demand curve is used to further test the performance of our algorithm. Since the single-period OEGF problem cannot take line-pack dynamics and ramping constraints into consideration, the original **[NLP]** (or **[BLP]**) is a more accurate representation of reality. Besides, the time scale is set to one hour in this paper, and the runtime limit is set to 3600 seconds.

Considering that day-ahead IEGS operation is not sensitive to the accuracy of solutions, the following tests are conducted at the medium accuracy level. All detailed results of the day-ahead operation can be downloaded from [33].

1) 39-bus-11-node Testing IEGS: Similar to the study of single-period OEGF problems, we start with a simple testing IEGS. The results of the day-ahead operation based on all 12 solving strategies are compared in Table VI, where the values of feasibility index E_f, optimality index E_o, and computing time are given only if the corresponding method can provide a qualified solution that meets the requirements of medium accuracy level. As can be seen, our sB&B algorithm (M12), along with M5-M10, is the only 7 methods that successfully solve the problem. Specifically, the CCP algorithm (M5) takes 226.59 seconds to find an extremely tight (E_f = 0%) solution which is verified to be only 0.29% larger than the best-known one. The AMP algorithm (M6) takes 432.82 seconds to give a converged solution, which is the second-longest among the 7 methods. The commercial solver Gurobi (M7) makes full use of its built-in sB&B algorithm to solve the problem in 187.20 seconds. The open-source solver Ipopt (M8) demonstrates its high efficiency by solving the super large nonlinear problems in 42.02 seconds, which is nearly a quarter of the time spent in Gurobi. Benefit from a built-in Ipopt solver for continuous programming, it takes the solver SCIP (M9) 170.23 seconds to converge, which is even faster than Gurobi. The traditional sB&B algorithm (M10) is the least efficient solving strategy, taking nearly 47 minutes to get the solution. By comparison, our sB&B algorithm (M12) is the fastest approach to get the optimal day-ahead operation solution. The computing time is only 28.98 seconds. Note that the improvement in computing speed does not impair the accuracy of the solution, as the data in Table VI shows that the maximal imbalance of non-convex constraints (E_f) is only 0.048%, and the relative error of the best known objective value is 0.01%.

Moreover, the comparison above further illustrates the importance of the modern implementation. The main reason for the poor performance of iteration-based M5, M6, and M10 is that their iteration process contains a lot of rework. Take the traditional sB&B algorithm M10 as an example, it treats each sB&B branching node as an independent problem, even though the node is almost the same as its parent. By comparison, our method implements the sB&B algorithm in a modern manner which integrates all sB&B nodes into a single search tree, so that our improved sB&B algorithm is nearly 100 times faster than the traditional one.

2) 500-bus-134-node Large-Scale IEGS: Then, the test is extended to a large-scale IEGS to verify the scalability of our sB&B algorithm. The day-ahead operation of this system is a super large non-convex programming model that is 24 times larger than the size of the single-period OEGF problem. The newly added nonlinearity and non-convexity bring additional challenges to global optimization algorithms.

The results of M1-M12 are given in Table VII. As can be observed, general-purpose solver Ipopt (M8) and our sB&B algorithm (M12) are the only two methods that successfully solve the problem. Moreover, our improved sB&B algorithm is even 228.69 seconds faster than Ipopt, which demonstrates that it can effectively address the non-convexity of super large-scale non-convex OEGF problems.

V. DISCUSSION

In this paper, we develop an improved sB&B algorithm to cope with a joint optimal energy flow problem for electricity and natural-gas systems. The problem has been given a lot of attention in recent years due to its non-convexity, and it is for the first time that an sB&B algorithm is specially designed for

the OEGF problem. The main contribution of this paper is a novel OEGF-specific two-stage spatial branching strategy for accelerating the traditional sB&B algorithm and achieving a balance among feasibility, optimality, and efficiency.

We compare our approach with another 11 solving strategies. The results show that our improved sB&B algorithm is competitive with the state-of-the-art general-purpose solvers, and as the size of the model increases, the superiority of our approach becomes even more significant.

The main drawback of the study could be an approximated representation of the OEGF problem. Specifically, we adopt a linearized OPF model with quadratic line losses terms in the electricity side of the IEGS. As to the natural-gas system, a steady-state gas flow model with line-pack dynamics is used. From the perspective of modeling, the proposed OEGF model can be further refined. For example, the models of GFUs and compressor stations can be formulated using a more accurate piecewise linearization method [5]. Further, the dynamics of pipelines can be modeled by partial differential equations in a more accurate manner [3]. However, the improvement of the accuracy does not change the non-convex nature of the OEGF problem. Thus, from a computational perspective, we choose to represent the OEGF problem as simply as possible, so that our contributions to the sB&B algorithm can be highlighted. As a matter of fact, although the proposed OEGF model is an approximated one, such approximation has been widely used in the short-term or day-ahead operation of IEGS [10], [12], [13], [15], which verifies the rationality of the approximation. Certainly, our sB&B algorithm can be easily extended to the models with more accurate formulations.

VI. Conclusion

This paper proposes an improved sB&B algorithm for the non-convex OEGF models, where the feasibility and optimality of the solutions can be ensured within a flexible threshold. A novel two-stage spatial branching strategy is developed to accelerate the convergence of the algorithm by improving the efficiency of branching operations.

Case studies are conducted on a test 39-bus-11-node IEGS and a large-scale 500-bus-134-node IEGS. The performance of the proposed algorithm is compared with another 11 solving strategies on both single-period and day-ahead operation problems. Numerical results verify that our sB&B algorithm is able to solve the non-convex OEGF problem for the exact solutions and is competitive with the state-of-the-art general-purpose solvers. Further, with the increase of model size, our improved sB&B algorithm outperforms the rest 11 methods in solving efficiency.

Appendix A
Linearization of Power Flow Expression

The linearized OPF adopted in this paper is an extension to the traditional DC OPF model. A detailed discussion of the linearization can be found in [11].

To avoid ambiguity and misunderstanding, an independent nomenclature is used here and given as follows. Let p_{ij}^l, q_{ij}^l be the active and reactive power flow from bus i to bus j, let v_i be the voltage magnitude of bus i, let θ_i be the voltage phase angle of bus i, let θ_{ij} be the angle difference between bus i and bus j, and G_{ij}, B_{ij} be the conductance and susceptance of branch ij. Then, the branch flow can be expressed as:

$$p_{ij}^l = G_{ij}(v_i^2 - v_i v_j \cos\theta_{ij}) - B_{ij} v_i v_j \sin\theta_{ij} \tag{82}$$

$$q_{ij}^l = B_{ij}(v_i v_j \cos\theta_{ij} - v_i^2) - G_{ij} v_i v_j \sin\theta_{ij} \tag{83}$$

Then, the second order Taylor series expansion method is used. Assuming that the value of θ_{ij} is a small number and is very close to zero, we have the following approximations:

$$\sin\theta_{ij} \approx \theta_{ij}, \quad \cos\theta_{ij} \approx 1 - \frac{\theta_{ij}^2}{2} \tag{84}$$

Next, we assume that the voltage magnitude v is close to 1.0 p.u., the following approximations can be obtained:

$$v_i v_j \theta_{ij} \approx \theta_{ij}, \quad v_i v_j \theta_{ij}^2 \approx \theta_{ij}^2 \tag{85}$$

By substituting (84)-(85) into (82)-(83), the following equations are given:

$$p_{ij}^l = G_{ij}(v_i^2 - v_i v_j) - B_{ij}\theta_{ij} + \frac{1}{2} G_{ij}\theta_{ij}^2 \tag{86}$$

$$q_{ij}^l = B_{ij}(v_i v_j - v_i^2) - G_{ij}\theta_{ij} - \frac{1}{2} B_{ij}\theta_{ij}^2 \tag{87}$$

Note that $(v_i - v_j)^2$ is very close to 0 and is far less than $v_i - v_j$, the expression $v_i^2 - v_i v_j$ can be approximated as:

$$\begin{aligned} v_i^2 - v_i v_j &= v_i^2 - \left[\frac{v_i^2 + v_j^2}{2} - \frac{(v_i - v_j)^2}{2}\right] \\ &= \frac{v_i^2 - v_j^2}{2} - \frac{(v_i - v_j)^2}{2} \approx \frac{v_i^2 - v_j^2}{2} \end{aligned} \tag{88}$$

Regarding v_i^2 and v_j^2 as independent variables, and substituting (88) into (86) and (87), the linearized branch flow can be written as:

$$p_{ij}^l \approx G_{ij}\frac{v_i^2 - v_j^2}{2} - B_{ij}\theta_{ij} + \frac{1}{2} G_{ij}\theta_{ij}^2 \tag{89}$$

$$q_{ij}^l \approx B_{ij}\frac{v_j^2 - v_i^2}{2} - G_{ij}\theta_{ij} - \frac{1}{2} B_{ij}\theta_{ij}^2 \tag{90}$$

where the first two terms in (89) and (90) are the active and reactive branch flow at the sending end of line ij, i.e., bus i as follows:

$$p_{ij} \approx \frac{1}{2} G_{ij}\left(v_i^2 - v_j^2\right) - B_{ij}\theta_{ij} \tag{91}$$

$$q_{ij} \approx \frac{1}{2} B_{ij}\left(v_j^2 - v_i^2\right) - G_{ij}\theta_{ij} \tag{92}$$

and the active and reactive power losses on line ij are given by p_{ij}^s and q_{ij}^s as follows:

$$p_{ij}^s = p_{ij}^l + p_{ji}^l = G_{ij}\theta_{ij}^2 \tag{93}$$

$$q_{ij}^s = q_{ij}^l + q_{ji}^l = -B_{ij}\theta_{ij}^2 \tag{94}$$

As can be observed, the derived branch flow equations (91) and (92) are the primitive form of (4) and (5), where reactive power and voltage magnitudes are incorporated to extend the traditional DC OPF model. As to the line losses in (93) and (94), they are the original form of (12) and (13). It can be seen that the quadratic line losses are the only nonlinear terms in the proposed OPF model.

REFERENCES

[1] A.Martinez-Mares and C. R.Fuerte-Esquivel, "A unified gas and power flow analysis in natural gas and electricity coupled networks," *IEEE Trans. Power Syst.*, vol. 27, no. 4, pp. 2156–2166, 2012.

[2] C.Shao, M.Shahidehpour, X.Wang, X.Wang, and B.Wang, "Integrated planning of electricity and natural gas transportation systems for enhancing the power grid resilience," *IEEE Trans. Power Syst.*, vol. 32, no. 6, pp. 4418–4429, 2017.

[3] J.Fang, Q.Zeng, X.Ai, Z.Chen, and J.Wen, "Dynamic optimal energy flow in the integrated natural gas and electrical power systems," *IEEE Trans. Sustain. Energy*, vol. 9, no. 1, pp. 188–198, 2018.

[4] A.Moreira, D.Pozo, A.Street, and E.Sauma, "Reliable renewable generation and transmission expansion planning: Co-optimizing system's resources for meeting renewable targets," *IEEE Trans. Power Syst.*, vol. 32, no. 4, pp. 3246–3257, 2017.

[5] T.Ding, Y.Xu, W.Wei, and L.Wu, "Energy flow optimization for integrated power-gas generation and transmission systems," *IEEE Trans. Ind. Informat.*, vol. 16, no. 3, pp. 1677–1687, 2020.

[6] C.Ordoudis, P.Pinson, and J. M.Morales, "An integrated market for electricity and natural gas systems with stochastic power producers," *Eur. J. Oper. Res.*, vol. 272, no. 2, pp. 642–654, 2019.

[7] M.Farivar and S. H.Low, "Branch flow model: Relaxations and convexification-part I," *IEEE Trans. Power Syst.*, vol. 28, no. 3, pp. 2554–2564, 2013.

[8] L.Yang, Y.Xu, H.Sun, and X.Zhao, "Two-stage convexification-based optimal electricity-gas flow," *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 1465–1475, 2020.

[9] Q.Li and V.Vittal, "Convex hull of the quadratic branch ac power flow equations and its application in radial distribution networks," *IEEE Trans. Power Syst.*, vol. 33, no. 1, pp. 839–850, 2018.

[10] S.Chen, A. J.Conejo, R.Sioshansi, and Z.Wei, "Operational equilibria of electric and natural gas systems with limited information interchange," *IEEE Trans. Power Syst.*, vol. 35, no. 1, pp. 662–671, 2020.

[11] Z.Yang, H.Zhong, A.Bose, T.Zheng, Q.Xia, and C.Kang, "A linearized OPF model with reactive power and voltage magnitude: A pathway to improve the MW-only DC OPF," *IEEE Trans. Power Syst.*, vol. 33, no. 2, pp. 1734–1745, 2018.

[12] S.D. Manshadi and M. E.Khodayar, "Coordinated operation of electricity and natural gas systems: A convex relaxation approach," *IEEE Trans. Smart Grid*, vol. 10, no. 3, pp. 3342–3354, 2019.

[13] S.Chen, A. J.Conejo, R.Sioshansi, and Z.Wei, "Unit commitment with an enhanced natural gas-flow model," *IEEE Trans. Power Syst.*, vol. 34, no. 5, pp. 3729–3738, 2019.

[14] H.Gao, J.Liu, and L.Wang, "Robust coordinated optimization of active and reactive power in active distribution systems," *IEEE Trans. Smart Grid*, vol. 9, no. 5, pp. 4436–4447, 2018.

[15] A. R.Sayed, C.Wang, J.Zhao, and T.Bi, "Distribution-level robust energy management of power systems considering bidirectional interactions with gas systems," *IEEE Trans. Smart Grid*, vol. 11, no. 3, pp. 2092–2105, 2020.

[16] A.Wächter and L. T.Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical Programming*, vol. 106, no. 1, pp. 25–57, 2006.

[17] J.Qiu, J.Zhao, H.Yang, and Z. Y.Dong, "Optimal scheduling for prosumers in coupled transactive power and gas systems," *IEEE Trans. Power Syst.*, vol. 33, no. 2, pp. 1970–1980, 2018.

[18] W.Dai, J.Yu, Z.Yang, H.Huang, W.Lin, and W.Li, "A static equivalent model of natural gas network for electricitygas co-optimization," *IEEE Trans. Sustain. Energy*, vol. 11, no. 3, pp. 1473–1482, 2020.

[19] C.Wang, W.Wei, J.Wang, L.Bai, Y.Liang, and T.Bi, "Convex optimization based distributed optimal gas-power flow calculation," *IEEE Trans. Sustain. Energy*, vol. 9, no. 3, pp. 1145–1156, 2018.

[20] A. M.Gleixner, T.Berthold, B.Mller, and S.Weltge, "Three enhancements for optimization-based bound tightening," *J. Glob. Optim.*, vol. 67, no. 4, pp. 731–757, 2016.

[21] Gurobi Optimization, LLC, "Gurobi optimizer reference manual," 2021. [Online]. Available: http://www.gurobi.com

[22] G.Gamrath, D.Anderson *et al.*, "The SCIP Optimization Suite 7.0," Optimization Online, Technical Report, March 2020. [Online]. Available: http://www.optimization-online.org/DB_HTML/2020/03/7705.html

[23] L.Liberti and C. C.Pantelides, "An exact reformulation algorithm for large nonconvex nlps involving bilinear terms," *J. Glob. Optim.*, vol. 36, no. 2, pp. 161–189, 2006.

[24] D.Gerard, M.Kppe, and Q.Louveaux, "Guided dive for the spatial branch-and-bound," *J. Glob. Optim.*, vol. 65, no. 1, pp. 685–711, 2017.

[25] M.Fischetti and M.Monaci, "A branch-and-cut algorithm for mixed-integer bilinear programming," *Eur. J. Oper. Res.*, vol. 282, no. 2, pp. 506–514, 2020.

[26] H.Nagarajan, M.Lu, S.Wang, R.Bent, and K.Sundar, "An adaptive, multivariate partitioning algorithm for global optimization of nonconvex programs," *J. Glob. Optim.*, vol. 74, no. 4, pp. 639–675, 2019.

[27] C.Pozo, G.Guillen-Gosalbez, A.Sorribas, and L.Jimenez, "A spatial branch-and-bound framework for the global optimization of kinetic models of metabolic networks," *Ind. Eng. Chem. Res.*, vol. 50, no. 9, pp. 5225–5238, 2011.

[28] P.Belotti, J.Lee, L.Liberti, F.Margot, and A.Wchter, "Branching and bounds tightening techniques for non-convex MINLP," *Optim. Methods Softw.*, vol. 24, no. 4-5, pp. 597–634, 2009.

[29] P.Liu, Z.Wu, W.Gu, P.Yu, J.Du, and X.Luo, "A novel acceleration strategy for N-1 contingency screening in distribution system," in *2020 IEEE Power & Energy Society General Meeting (PESGM)*, 2020, pp. 1–5.

[30] IBM ILOG, "IBM ILOG CPLEX Optimization Studio 20.1.0," 2020. [Online]. Available: https://www.ibm.com/docs/en/icos/20.1.0

[31] R. D.Zimmerman, C. E.Murillo-Snchez, and R. J.Thomas, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Trans. Power Syst.*, vol. 26, no. 1, pp. 12–19, 2011.

[32] M.Schmidt, D.Aßmann *et al.*, "GasLib – A library of gas network instances," *Data*, vol. 2, no. 4, p. article 40, 2017.

[33] P.Liu, "Scientific data of parameters and results," 2021. [Online]. Available: https://github.com/TesLarry/open-scientific-data/tree/main/journal-articles/01-TPWRS-00162-2021

Pengxiang Liu (Student Member, IEEE) received the B.S. degree in electrical engineering from Zhengzhou University, China, in 2017, and the M.Eng. degree in electrical engineering from Southeast University, China, in 2019. He is currently working toward the Ph.D. degree from the School of Electrical Engineering, Southeast University, China.

In the study period during 2021-2022, he will be a visiting Ph.D. student with the Department of Electrical and Electronic Engineering, Imperial College London. His research interests include multi-energy system optimization, operations research, and machine learning.

Zhi Wu (Member, IEEE) received the B.Eng. degree in mathematics and the M.Sc. degree in electrical engineering from Southeast University, China, in 2009 and 2012, respectively, and the Ph.D. degree from the University of Birmingham, U.K., in 2016.

He is currently working as an Associate Professor with the School of Electrical Engineering, Southeast University. His research interests include renewable energy, planning, and optimization techniques.

Wei Gu (Senior Member, IEEE) received the B.S. and Ph.D. degrees in electrical engineering from Southeast University, Nanjing, China, in 2001 and 2006, respectively.

From 2009 to 2010, he was a Visiting Scholar with the Department of Electrical Engineering, Arizona State University, Tempe, AZ, USA. He is currently a Professor with the School of Electrical Engineering, Southeast University. He is the Director of the Institute of Distributed Generations and Active Distribution Networks. His research interests include distributed generations and micro-grids, and integrated energy system. He is an Editor for the IEEE Transactions on Power Systems, the IET Energy Systems Integration, and the Automation of Electric Power Systems (China).

Yuping Lu (Senior Member, IEEE) received the Ph.D. degree in electrical engineering from the City University, U.K., in 2003.

He is currently working as a Professor with the School of Electrical Engineering, Southeast University. His research interests include power system protection (especially digital relaying of generator-transformer unit) and protection-and-control techniques of distribution systems.