A Branch-and-Cut Benders Decomposition Algorithm for Transmission Expansion Planning

Shengjun Huang, *Student Member, IEEE* and Venkata Dinavahi, *Senior Member, IEEE*

***Abstract*—The emergence of a great number of regional planning projects worldwide has considerably increased the complexity and relevance of transmission expansion planning, prompting intensive research and investigation on the formulation and solution. In this paper, the security constrained transmission expansion planning problem is addressed by a branch-and-cut Benders decomposition (BCBD) algorithm. It is a deterministic method where the global optimal solution can be guaranteed in a finite number of iterations. Based on this implementation framework, four acceleration strategies have been employed to enhance the performance. For the validation of accuracy and efficiency, the commercial solver Cplex running on the same platform is introduced for comparison, where four types of mixed-integer linear programming algorithms are discriminated by specifying two pairs of key settings, including dynamic searching and parallel implementation. The superiority of BCBD over Cplex has been validated by case studies, where five benchmark systems ranging from 6 to 300 buses are employed. In addition, performance analysis between BCBD and classical Benders decomposition has also been carried out to distinguish the contribution of branch-and-cut framework and acceleration strategies.**

***Index Terms*—Benders decomposition (BD), branch-and-cut (B&C) algorithm, mixed-integer linear programming (MILP), parallel processing, security constrained transmission expansion planning (SCTEP).**

I. Introduction

TRANSMISSION expansion planning (TEP) has regained its importance as a vital planning study especially in the context of the growing complexity of smart power systems [1]–[3]. Reliability and security concerns of electricity supply during this radical evolution of modern power systems is causing operators and engineers to find improved and efficient solutions to classical planning problems [4], [5]. Therefore, security constrained Transmission expansion planning (SCTEP) has emerged [6], [7], which determines how to expand and reinforce the transmission network in order to supply electricity to consumers in a secure and economic fashion, where the forecasted load growth over a specific time span and the available generation assets are available for decision making.

Two main types of models have been widely used to represent the transmission network in TEP studies: the dc and ac power flow models. The classical dc model is, in general, a mixed-integer, nonlinear, nonconvex, and NP-hard problem, which is utilized in TEP solution mostly when the security criterion is not considered [8]. Another widely utilized dc model is the disjunctive model, which is derived from the classical dc model by representing the integer decision variables with binary decision variables, and eliminating the nonlinear property by the introduction of linearizing constant (big-M), resulting in a mixed-integer linear programming (MILP) problem. The disjunctive model is mainly employed for the solution of SCTEP [6], [9]–[14]. Full ac model is considered only at a later stage of the planning process when the most attractive topologies have been determined [8].

Security, in a deterministic sense (which is the current common industry practice), is the capability of a power system to withstand a specified set of credible contingencies [6], [15]. Modeling security drastically increases the complexity of the resulting problem since the unavailability of system components needs to be characterized. One of the most extensively adopted criteria in the literature on SCTEP is the $N-1$ security criterion, which states that the system should be expanded in such a way that if a line was withdrawn the resulting system should still operate adequately [16], [17].

In this paper, SCTEP formulated by the disjunctive model is investigated with the consideration of transmission line $N-1$ security criterion. To tackle this tough MILP problem, lots of research efforts have been devoted, such as the direct MILP solution method [9] and the adjustable robust optimization approach [6]. However, both methods formulate all the scenarios/contingencies into one whole MILP problem, leading to the number of decision variables and constraints increases to a scale of thousands or even millions, which presents a major challenge for several popular commercial MILP solvers, such as Cplex, Gurobi, and Lingo. Therefore, decomposition strategies were investigated [10]–[14] to separate the complex full problem into several smaller subproblems, among which Benders decomposition (BD) [18] has received the most attention in the literature.

BD has a good reputation from its wide application for different kinds of optimization problems; nevertheless, there are few fields where BD can work very well without much additional

Manuscript received April 13, 2017; revised July 27, 2017; accepted November 16, 2017. Date of publication December 7, 2017; date of current version February 22, 2019.This work was supported by the Natural Sciences and Engineering Research Council of Canada. The work of S. Huang was supported by the China Scholarship Council under Grant 201403170337. *(Corresponding author: Shengjun Huang.)*

S. Huang is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton AB T6G 2V4, Canada, and is also with the College of Information System and Management, National University of Defense Technology, Changsha 410073, China (e-mail: shengjun@ualberta.ca).

V. Dinavahi is with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB T6G 2V4, Canada (e-mail: dinavahi@ualberta.ca).

Digital Object Identifier 10.1109/JSYST.2017.2775610

implementation work, i.e., enhancements to BD are almost always necessary. Within several attempts to solve SCTEP with BD, Asadamongkol and Eua-arporn [10] introduced a local search procedure to solve the master problem, with considerably decreased computational time reported compared with classical Benders decomposition (CLBD); however, additional constraints are applied to reduce the search space at the time of problem formulation, i.e., the original problem has been revised. Lumbreras and Ramos [11] introduced a lot of interesting improvements employed by a practical project using textual description, such as inexact master problem resolution mechanism, semirelaxed cuts for discrete decision variables, and the combination of monocut and multicut; nevertheless, few technical and mathematical details were exhibited. Jenabi *et al.* [12] introduced two valid inequalities to reduce the search space of the master problem, as well as multiple generation cuts and strong high density cut to boost the convergent efficiency. Alizadeh-Mousavi and Zima-Bočkarjova [13] proposed a set of appropriate Benders cuts specifically tailored for the binary decision variables, and also studied the effectiveness of standard and modified disjunctive model. Huang and Dinavahi [14] developed three acceleration strategies, resulting in significant reduction in both execution time and the number of iterations.

Although remarkable advances have been made in BD for finding the global optimal solution of SCTEP, some beneficial endeavors still need to be conducted, one of them is relieving the intense computational burden for the master problem. All the above-mentioned enhancements on BD, including our previous work [14], were implemented in the classical framework, i.e., iteratively solving the master problem (which is integer linear programming (ILP) or MILP) and the subproblem [which is linear programming (LP)] until the convergence criteria are met. It can be very expensive from a computational point of view since the solution of ILP and MILP is heavily involved, which is much more computationally intensive than the LP solution. Therefore, in this paper, instead of solving the MILP and ILP master problem, we integrate the BD process into a branch-and-cut (B&C) framework, resulting in a branch-and-cut Benders decomposition (BCBD) algorithm, where only the relaxed LP master problem needs to be solved at each node; thus, much computational effort can be saved, as well as the total execution time. Additionally, four acceleration strategies have also been utilized to enhance the solution performance. Comprehensive computational experiments conducted on five benchmark test systems indicate that BCBD (working in sequential mode) performs better for the majority of problems although some MILP solvers work in parallel with 24 threads.

The main contributions of this paper are as follows.

1) For the circumstance that more than one circuits can be built in each corridor, a nonanticipativity constraint satisfying contingency construction technique has been proposed.
2) In order to release the heavy computational burden of MILP or ILP master problem in the classical BD, an optimization technique for MILP has been investigated by integrating the CLBD into a B&C framework, resulting in the BCBD algorithm, where only LP solution is related for both master problem and subproblems. To the best of our knowledge, this kind of methodology has never been reported in the solution of SCTEP.
3) Four acceleration strategies have been employed in BCBD as the supplementary components to enhance the convergence efficiency as well as to restrict the solution space: two-phase method, multicut strategy, valid inequality, and optimal preconditioning.
4) Intensive comparison and performance analysis have been carried out between BCBD and other algorithms, including four types of MILP algorithm discriminated from Cplex solver and seven types of BD algorithm separated from BCBD and CLBD.

The rest of this paper is organized as follows. The SCTEP disjunctive model formulation and stochastic programming are presented in Section II. Section III describes the integration of BD into B&C framework, as well as the four acceleration strategies. Computational experiments and performance evaluations are reported and discussed in Section IV. Finally, conclusions and future work are highlighted in Section V.

II. Problem Formulation

A. Disjunctive Model

As shown in a range of previous work [6], [9], [10], [12]–[14], the disjunctive model is given as follows:

$$\min \sum_{k=1}^{K} \sum_{(i,j)\in\mathcal{C}} c_{ij} n_{ij}^{k} + \sum_{s=1}^{|\mathcal{S}|} \left(P \sum_{i=1}^{|\mathcal{N}|} r_i^{(s)} \right) \tag{1}$$

subject to

$$n_{ij}^{k+1} \le n_{ij}^{k},\ k = 1,\ldots,K-1, \qquad ij \in \mathcal{C} \tag{2}$$

$$\sum_{ij\in\mathcal{E}} f_{ij}^{0(s)} + \sum_{k=1}^{K} \sum_{ij\in\mathcal{C}} f_{ij}^{k(s)} + g_i^{(s)} + r_i^{(s)} = d_i, \quad i \in \mathcal{N} \tag{3}$$

$$f_{ij}^{0(s)} - \gamma_{ij} n_{ij}^{0(s)} \left(\theta_i^{(s)} - \theta_j^{(s)} \right) = 0, \qquad ij \in \mathcal{E} \tag{4}$$

$$|f_{ij}^{k(s)} - \gamma_{ij} \left(\theta_i^{(s)} - \theta_j^{(s)} \right)| \le M \left(1 - n_{ij}^{k} \right), \qquad ij \in \mathcal{C} \tag{5}$$

$$|f_{ij}^{0(s)}| \le \bar{f}_{ij} n_{ij}^{0(s)}, \qquad ij \in \mathcal{E} \tag{6}$$

$$|f_{ij}^{k(s)}| \le \bar{f}_{ij} n_{ij}^{k}, \qquad ij \in \mathcal{C} \tag{7}$$

$$0 \le g_i^{(s)} \le \bar{g}_i, \qquad i \in \mathcal{N} \tag{8}$$

$$0 \le r_i^{(s)} \le d_i, \qquad i \in \mathcal{N} \tag{9}$$

$$n_{ij}^{k} \in \{0, 1\},\ k = 1,\ldots,K.$$

where

$\mathcal{C}$	set of candidate transmission lines;
$\mathcal{E}$	set of existing transmission lines;
$\mathcal{N}$	set of bus nodes;
$\mathcal{S}$	set of security contingencies/scenarios;
K	maximum no. of circuits can be built for each line;
M	constant used to linearize the power flow constraint of transmission line;

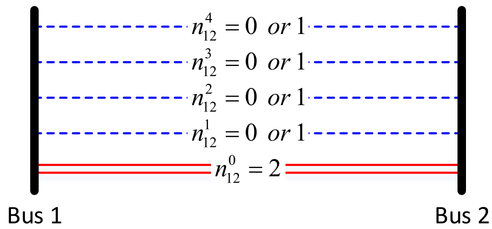


Fig. 1. Initiated and candidate circuits on corridor 1–2.

P	penalty factor for loss of load;
c_{ij}	construction cost for transmission line ij;
d_i	load demand at node i;
$f_{ij}^{0(s)}$	power flow of the original circuit ij at scenario s;
$f_{ij}^{k(s)}$	power flow of the kth parallel circuit ij at scenario s;
$\bar{f}_{ij}$	maximum power flow on transmission line ij;
$g_i^{(s)}$	amount of generation at node i at scenario s;
$\bar{g}_i$	maximum amount of generation at node i;
$n_{ij}^{0(s)}$	initial no. of circuits on corridor ij for scenario s;
n_{ij}^k	binary variables indicating whether the kth parallel circuit of corridor ij is built;
$r_i^{(s)}$	amount of load shedding at node i at scenario s;
γ_{ij}	susceptance of transmission line ij;
$\theta_i^{(s)}$	voltage angle of node i at scenario s.

The objective function (1) comprises construction cost and load curtailment penalty; constraint (2) is a valid inequality strategy utilized to refine the solution space by eliminating some equivalent solutions, which will be explained in Section III-C; constraint (3) is the load balance requirement for each bus, i.e., Kirchhoff's current law; constraints (4) and (5) represent the Kirchhoff's voltage law for existing and candidate circuits; power flow is limited by (6) and (7); and finally, the amount of generation and load curtailment are restricted by (8) and (9). The configuration of parameter n_{ij}^0 and binary decision variables $n_{ij}^k (k = 1, \ldots, K)$ are exemplified by Fig. 1 on corridor 1–2.

Scenario set $\mathcal{S}$ related to $N-1$ transmission line outage can be defined as either a full version for the convenience of formulation

$$\mathcal{S} = \mathcal{C} \tag{10}$$

or a refined one to save computational time:

$$\mathcal{S} = \left\{ ij | n_{ij}^0 + \sum_{k=1}^{K} n_{ij}^k > 0, ij \in \mathcal{C} \right\}. \tag{11}$$

For a given solution, it is sufficient only when it can withstand the loss of any $ij \in \mathcal{S}$. If (10) is utilized, the outage of all $ij \in \mathcal{C}$ should be checked; on the other hand, (11) only exams those corridors with original or newly built circuits.

All the components in (1)–(9) with the superscript $^{(s)}$ are scenario-dependent decision variables, which will be valued automatically in the process of problem solving, except for $n_{ij}^{0(s)}$, which is a parameter required to describe each scenario that needs to be predefined. Suppose $i_s j_s$ is the corresponding unavailable corridor in scenario s, then the process of $n_{ij}^{0(s)}$ assignment as well as contingency construction can be given by Algorithm 1.

Algorithm 1: Contingency Construction.

1: **for** Each considered contingency s **do**
2: **for** Each candidate and existing circuit ij **do**
3: **if** Considered circuit ij is corridor $i_s j_s$ **then**
4: **if** $n_{i_s j_s}^0 > 0$ **then**
5: Set $n_{ij}^{0(s)} = n_{i_s j_s}^0 - 1$.
6: **else**
7: Replace constraint (5) when $k = 1$ by $f_{ij}^{1(s)} = 0$.
8: **end if**
9: **else**
10: Set $n_{ij}^{0(s)} = n_{ij}^0$.
11: **end if**
12: **end for**
13: **end for**

The main logic of Algorithm 1 lies in the fact that the paralleled circuits are equivalent with each other. When a contingency on corridor $i_s j_s$ needs to be formulated, we first check whether there are any existing initial circuits; if yes, then model this scenario by reducing one initial circuit (step 5); otherwise, set the power flow of the first parallel link to be 0 (step 7), which means $n_{i_s j_s}^1$ is withdrawn. Obviously, step 7 satisfies the circumstance that no power flow needs to be taken by $i_s j_s$; otherwise, at least one parallel link needs to be built; however, it must start from $n_{i_s j_s}^2 = 1$ [where $n_{i_s j_s}^1$ still needs to be valued as 1 according to valid inequality (2)] since $f_{ij}^{1(s)} = 0$ due to step 7, which means the cost of $n_{i_s j_s}^1$ has been counted although it provides no contribution to the power flow. The above-mentioned algorithm is suitable for both definitions of $\mathcal{S}$ given in (10) and (11).

B. Stochastic Programming

The MILP disjunctive model of SCTEP is usually treated as the two-stage stochastic programming problem [10]–[14], which can be formulated as follows:

$$\begin{array}{llllllll}
\min\limits_{\boldsymbol{x},\boldsymbol{y}_s} & \boldsymbol{c}^T\boldsymbol{x} & + & P\boldsymbol{q}_1^T\boldsymbol{y}_1 + P\boldsymbol{q}_2^T\boldsymbol{y}_2 + \cdots + P\boldsymbol{q}_s^T\boldsymbol{y}_s & & & & (12)\\
\text{s.t.} & \boldsymbol{A}\boldsymbol{x} & & & & & \leq & \boldsymbol{b}\\
& \boldsymbol{T}_1\boldsymbol{x} & + & \boldsymbol{W}_1\boldsymbol{y}_1 & & & \leq & \boldsymbol{h}_1\\
& \boldsymbol{T}_2\boldsymbol{x} & + & & \boldsymbol{W}_2\boldsymbol{y}_2 & & \leq & \boldsymbol{h}_2\\
& \vdots & + & & & \cdots & \leq & \vdots\\
& \boldsymbol{T}_s\boldsymbol{x} & + & & & \boldsymbol{W}_s\boldsymbol{y}_s & \leq & \boldsymbol{h}_s\\
& & & \boldsymbol{x} \in \boldsymbol{X}, \boldsymbol{y}_1 \geq 0, \ldots, \boldsymbol{y}_s \geq 0 & & & &
\end{array}$$

where $\boldsymbol{c}$ and $\boldsymbol{q}_s$ are the cost vectors, $\boldsymbol{x}$ is a vector that denotes the integer decision variables restricted by a set of $\boldsymbol{X}$, $\boldsymbol{y}_s$ are the continuous variables for each scenario s, and $\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{T}_s, \boldsymbol{W}_s$, and $\boldsymbol{h}_s$ are coefficient matrices and vectors. The detailed configurations of decision variables $\boldsymbol{x}$ and $\boldsymbol{y}_s$ are given in Table I. The structures of coefficient matrices and vectors can be determined accordingly based on Table I and (1)–(9).

TABLE I
DEFINITION OF DECISION VARIABLES $\boldsymbol{x}$ AND $\boldsymbol{y}_s$

Decision variables	Properties	Configurations
$\boldsymbol{x}$	Binary	$\left\{ \left[n_{ij}^{k} \right]_{ij \in \mathcal{C},\, k=1\ldots K} \right\}$
$\boldsymbol{y}_1$	Continuous	$\left\{ \left[f_{ij}^{0(1)} \right]_{ij \in \mathcal{E}}, \left[f_{ij}^{k(1)} \right]_{ij \in \mathcal{C},\, k=1\ldots K}, \left[g_i^{(1)} \right]_{i \in \mathcal{N}}, \left[r_i^{(1)} \right]_{i \in \mathcal{N}}, \left[\theta_i^{(1)} \right]_{i \in \mathcal{N}} \right\}$
$\boldsymbol{y}_2$	Continuous	$\left\{ \left[f_{ij}^{0(2)} \right]_{ij \in \mathcal{E}}, \left[f_{ij}^{k(2)} \right]_{ij \in \mathcal{C},\, k=1\ldots K}, \left[g_i^{(2)} \right]_{i \in \mathcal{N}}, \left[r_i^{(2)} \right]_{i \in \mathcal{N}}, \left[\theta_i^{(2)} \right]_{i \in \mathcal{N}} \right\}$
...	Continuous	...
$\boldsymbol{y}_s$	Continuous	$\left\{ \left[f_{ij}^{0(s)} \right]_{ij \in \mathcal{E}}, \left[f_{ij}^{k(s)} \right]_{ij \in \mathcal{C},\, k=1\ldots K}, \left[g_i^{(s)} \right]_{i \in \mathcal{N}}, \left[r_i^{(s)} \right]_{i \in \mathcal{N}}, \left[\theta_i^{(s)} \right]_{i \in \mathcal{N}} \right\}$

It is relatively straightforward to translate the MILP disjunctive model (1) into stochastic programming (12) by converting constraints (3)–(9) into $\boldsymbol{T}_s\boldsymbol{x} + \boldsymbol{W}_s\boldsymbol{y}_s \le \boldsymbol{h}_s$ for each scenario s and keeping the other constraints and objective function the same. The solution process for stochastic programming is usually implemented in an iterative manner. At each iteration, the master problem (first stage) is solved to obtain a temporary integer solution, which will be verified by subproblems (scenarios, referred as the second stage), where feasibility or optimality cuts may be generated according to specified rules, and then added to the master problem for the next iteration solution. The procedure terminates if no cuts can be extracted from the second stage. During the above-mentioned course, the nonanticipativity constraint [19] of stochastic programming prescribes that all the second stage subproblems should receive the same temporary solution from the first stage, which means each scenario is independent with the temporary solution. In this paper, scenario s is determined by Algorithm 1, where the intermediate solution $\bar{n}_{ij}^k$ is not related, i.e., nonanticipativity constraints are maintained.

It should be noted that, from the point of view of practical operation, if any loss of load $r_i^{(s)}$ is positive, the solution is invalid; however, mathematically speaking, with the introduction of $r_i^{(s)}$, the system is always feasible, i.e., any solution of $\bar{n}_{ij}^k$ will not violate any constraints since there always has at least a scheme of $r_i^{(s)} = d_i$ to compensate the load imbalance in (3). Therefore, the generated two-stage stochastic programming is a complete recourse problem, i.e., all the first stage solutions are feasible in the second stage.

III. Solution Methodology

A. Benders Decomposition

BD separates the original problem (12) into master problem [MP] and subproblem [SP] according to binary and continuous decision variables. In addition, the dual problem of [SP] should be generated and marked as [DP]:

$$\text{[MP]} \quad \min_{\boldsymbol{x}} \quad \boldsymbol{c}^T\boldsymbol{x} + Q \tag{13}$$

$$\text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} \le \boldsymbol{b}, \boldsymbol{x} \in \boldsymbol{X} \tag{14}$$

$$\text{[SP]} \quad \min_{\boldsymbol{y}_s} \quad \boldsymbol{q}_s^T\boldsymbol{y}_s \tag{15}$$

$$\text{s.t.} \quad \boldsymbol{W}_s\boldsymbol{y}_s \le \boldsymbol{h}_s - \boldsymbol{T}_s\bar{\boldsymbol{x}}, \boldsymbol{y}_s \ge 0, s \in \mathcal{S} \tag{16}$$

$$\text{[DP]} \quad \max_{\boldsymbol{u}_s} \quad (\boldsymbol{h}_s - \boldsymbol{T}_s\bar{\boldsymbol{x}})^T\boldsymbol{u}_s \tag{17}$$

$$\text{s.t.} \quad \boldsymbol{W}_s^T\boldsymbol{u}_s \le \boldsymbol{q}_s, \boldsymbol{u}_s \le 0, s \in \mathcal{S} \tag{18}$$

where $Q = \sum_{s\in\mathcal{S}} p_s Q_s$ is the weighted sum of objective values from each subproblem for scenario s, $\bar{\boldsymbol{x}}$ is the temporary solution of [MP], and $\boldsymbol{u}_s$ is the dual values for constraints in [SP].

As this is a complete recourse problem, [SP] is always feasible; thus, [DP] is bounded, and the objective function value of [DP] provides a valid lower bound for [SP] according to dual theory; therefore, an optimality cut can be generated:

$$Q_s \ge (\boldsymbol{h}_s - \boldsymbol{T}_s\boldsymbol{x})^T\bar{\boldsymbol{u}}_s, s \in \mathcal{S} \tag{19}$$

where $\bar{\boldsymbol{u}}_s$ is the optimal solution of [DP]. Since the feasibility of [SP] is always valid, feasibility cuts are not required in this problem. After the adding of optimality cut, the [MP] is evolved as follows:

$$\text{[MP]} \quad \min_{\boldsymbol{x}} \quad \boldsymbol{c}^T\boldsymbol{x} + Q \tag{20}$$

$$\text{s.t.} \quad \boldsymbol{A}\boldsymbol{x} \le \boldsymbol{b}, \boldsymbol{x} \in \boldsymbol{X} \tag{21}$$

$$Q \ge \sum_{s\in\mathcal{S}} p_s(\boldsymbol{h}_s - \boldsymbol{T}_s\boldsymbol{x})^T\bar{\boldsymbol{u}}_s^i, i = 1\ldots N \tag{22}$$

where N is the current iteration number.

In each iteration, the lower bound LB is the objective value of [MP], and the upper bound UB is determined by the objective value of [DP], i.e., $\text{UB}' = \boldsymbol{c}^T\bar{\boldsymbol{x}} + \sum_{s\in\mathcal{S}} p_s(\boldsymbol{h}_s - \boldsymbol{T}_s\bar{\boldsymbol{x}})^T\bar{\boldsymbol{u}}_s$. The solution process of BD is given in Algorithm 2. After initialization, the [MP] is first solved, with temporary solution $\bar{\boldsymbol{x}}$, and objective value LB$'$ is obtained. Update LB into LB$'$ if LB$'$ $>$ LB. Based on $\bar{\boldsymbol{x}}$, the [DP] for each scenario s can be solved to generate optimality cuts (22) and UB$'$. Then, cuts are added into [MP] and UB is updated into UB$'$ if UB$'$ $<$ UB. The above-mentioned iterative process terminates if $|\text{UB} - \text{LB}| \le \epsilon$. The transferred data between [MP] and [DP] is their optimal solution $\bar{\boldsymbol{x}}$ and $\bar{\boldsymbol{u}}_s$. It should be noted that Algorithm 2 is a

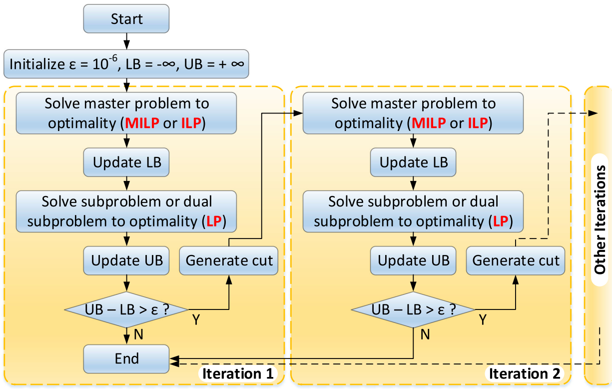


Fig. 2. Flowchart of BD within classical implementation framework.

Algorithm 2: Benders Decomposition (simplified version).

1: Set $\epsilon = 10^{-6}$, $LB = -\infty$ and $UB = +\infty$.
2: **while** $|UB - LB| > \epsilon$ **do**
3: Solve [MP] with all generated cuts to get a solution $\bar{\boldsymbol{x}}$ with objective value LB'.
4: **if** $LB' > LB$ **then**
5: Set $LB = LB'$.
6: **end if**
7: Solve [DP] with $\bar{\boldsymbol{x}}$ for each scenario s, denote the optimal solution as $\bar{\boldsymbol{u}}_s$.
8: Calculate $UB' = \boldsymbol{c}^T\bar{\boldsymbol{x}} + \sum_{s\in S} p_s(\boldsymbol{h}_s - \boldsymbol{T}_s\bar{\boldsymbol{x}})^T\bar{\boldsymbol{u}}_s$.
9: Generate cut (22) based on $\bar{\boldsymbol{u}}_s$, and add it into [MP].
10: **if** $UB' < UB$ **then**
11: Set $UB = UB'$.
12: **end if**
13: **end while**

simplified version of CLBD since the feasibility cuts are omitted. For more details of BD implementation on the power system, please refer to [20]. An illustrative flowchart of implementation framework is given in Fig. 2. Nevertheless, repeatedly solving [MP] to optimality, adding a cut and resolving it can be very expensive in terms of computational resources.

B. BCBD

In order to relieve the computational burden of [MP], a branch-and-check strategy was advocated in [21], where only one MILP search tree of [MP] was built and maintained, i.e., [MP] is solved into optimality by only once, whereas all the efforts spent on each node of the tree are solving the LP relaxed [MP], which is much easier and faster compared with MILP solution. In spirit of this strategy, we integrated BD into the B&C framework provided by ILOG–Cplex concert technology, resulting in the BCBD algorithm. The step by step implementation process is illustrated by Algorithm 3. It is obvious that the key steps of Algorithm 2 are line 3 and lines 7–9, which are all represented and highlighted in Algorithm 3. Other steps of Algorithm 3 are corresponding to the B&C process. The solution framework of BCBD is illustrated in Fig. 3. It can be seen that only LP is addressed in both [MP] and [DP].

Algorithm 3: BCBD Algorithm.

1: Add the original [MP] into tree L, set final solution $\boldsymbol{x}^* = null$ and value $v^* = +\infty$.
2: **while** L is not empty **do**
3: Select a node [MP] from L.
4: Solve the LP relaxation of [MP] to obtain an optimal solution $\bar{\boldsymbol{x}}$ with objective value v. (*corresponding to line 3 in* **Algorithm 2**)
5: **if** LP is infeasible **then**
6: Prune the node.
7: **else if** $v \geq v^*$ **then**
8: Prune the node.
9: **else if** $\bar{\boldsymbol{x}}$ is integer **then**
10: Solve [DP] based on $\bar{\boldsymbol{x}}$ and generate Benders cuts. (*corresponding to lines 7–9 in* **Algorithm 2**)
11: **if** No cuts generated **then**
12: Update $v^* = v$ and $\boldsymbol{x}^* = \bar{\boldsymbol{x}}$, prune the node.
13: **else**
14: Add the cuts to the LP relaxation and return to 4.
15: **end if**
16: **else**
17: **if** A candidate solution $\bar{\boldsymbol{x}}'$ is found **then**
18: Search for cutting planes that are violated by $\bar{\boldsymbol{x}}'$.
19: Solve [DP] based on $\bar{\boldsymbol{x}}'$ and generate Benders cuts. (*corresponding to lines 7–9 in* **Algorithm 2**)
20: If any cutting planes or Benders cuts are found, add them to the LP relaxation and return to 4.
21: **end if**
22: Choose one non-integral variable from $\bar{\boldsymbol{x}}$ to branch, create two nodes and add them to L.
23: **end if**
24: **end while**
25: **return** Final solution $\boldsymbol{x}^*$ and value v^*.

Large numbers of constraints will usually be involved in large-scale problems; however, many of them are redundant or at least not binding near the optimal solution, and any of them can be ruled out without prior information. One proper method to handle these constraints is to set them as lazy constraints and put them into a pool in Cplex. When a solution is generated, the solver will check if any lazy constraints are violated and, if so, adds them to the active set. Lazy constraints that were previously added but have not been binding for a while will be returned to the pool. In Algorithm 3, the lazy constraint with callback function is utilized, which is guaranteed to be checked every time Cplex B&C framework finds a candidate solution (see line 17), regardless of how the candidate is found (such as, node LP solution, rounding, and various heuristics).

C. Acceleration Strategies

To improve the solution performance, four acceleration strategies are employed in this paper. They are beneficial from different aspects: generating high-quality initiate points, increasing the number of cuts, restricting the solution space, and obtaining

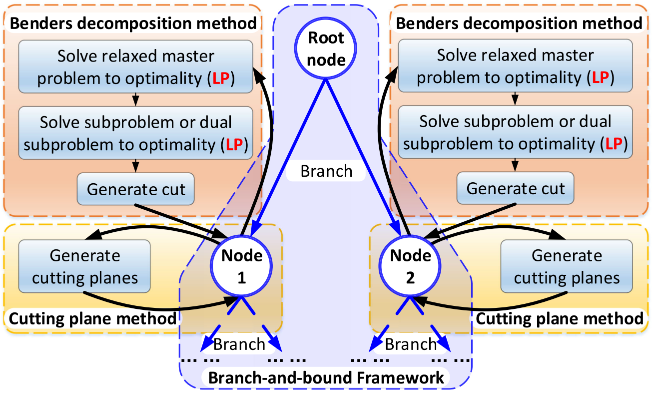


Fig. 3. Flowchart of BD within B&C implementation framework.

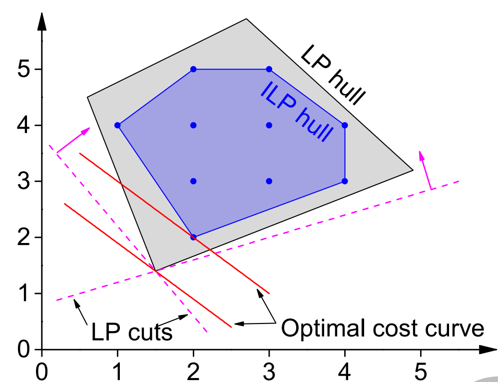


Fig. 4. ILP hull versus LP hull.

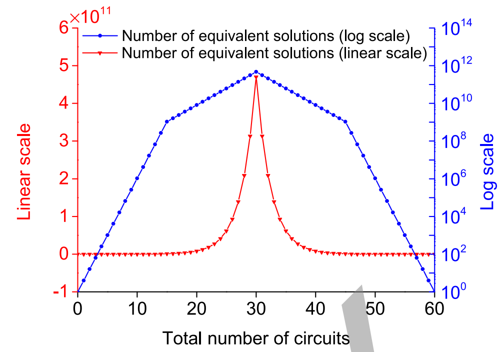


Fig. 5. Number of equivalent solutions for the Garver 6-bus system without valid inequality.

Algorithm 4: Two-Phase BCBD Algorithm.

1: *Phase 1:*
2: Remove integrality constraints on all variables.
3: Solve the problem using **Algorithm 2**, and keep all the generated cuts.
4: *Phase 2:*
5: Reintroduce integrality constraints on the master problem variables.
6: Add all cuts generated from *Phase 1* into master problem, and solve the problem using **Algorithm 3**.

tighter cuts. Their performance on CLBD framework has been reported in [14], [22]; however, implementation on the developed BCBD framework has never been revealed before. For simplicity, only the first method is depicted in detail. The mathematical formulation and further explanation of the other strategies are given in [14].

1) Two-Phase Method: As shown in Fig. 4, the convex hull of the feasible region of the MILP is always contained within the LP relaxation; thus, all added LP cuts are valid for MILP, which leads to the two-phase method [22]. In the two-phase method, the MILP problem is relaxed into LP and solved to optimality by BD in *Phase 1*, all the generated cuts are sent to *Phase 2* where the integrality property of MILP is considered. The general implementation framework is given in Algorithm 4. It has been proved by experiments that this method plays a major role in making the MILP start from a high-quality point (global optimal of LP relaxation), especially when the MILP has a small integrality gap.

2) Multicut Method: Instead of returning only one cut at each iteration to [MP], as shown in (22) for the CLBD, a multicut strategy can be employed to enhance the convergence efficiency by generating one cut from each scenario [11], [12], [23], [24], i.e., multiple cuts are introduced for each iteration.

3) Valid Inequality: In SCTEP, the circuits built in parallel are similar; thus, there will be several equivalent optimal solutions, which may introduce complexity during the solution process. Results for the Garver 6-bus system is shown in Fig. 5, where the maximum number of equivalent solutions is as large as 4.70×10^{11}. Therefore, in order to save computational efforts for solution searching and make the optimal solution logically unique, valid inequality (2) is employed for all systems and algorithms considered in this paper.

4) Optimal Preconditioning: One of the preconditions for the optimality of SCTEP solution is $Q = 0$ in (20), which means no loss of load is tolerable. Since p_s is positive, then $Q = 0$ is equivalent with $Q_s = 0$ for all $s \in \mathcal{S}$. Therefore, this optimality precondition can be embedded into the Benders cut generation (19) by forcing $Q_s = 0$.

IV. Computational Experiments

To evaluate the performance of the proposed algorithm, we implemented our method in C++ and embedded it within the ILOG–Cplex concert technology framework, based on ILOG–Cplex 12.5.1. The implementation platform is a 64-b Windows desktop with 32 GB RAM and 2 Intel Xeon E5-2620 CPUs at 2.10 GHz (12 cores for each).

In our experiment, the Cplex MILP solver is employed for comparison. In order to achieve the best performance from the MILP solver, two key modes are investigated: 1) sequential and parallel implementation mode; and 2) traditional and dynamic B&C search pattern. Therefore, four solvers are identified in total: *MILP_dynamic (1 thread)*, *MILP_dynamic (24 threads)*, *MILP_traditional (1 thread)*, and *MILP_traditional (24 threads)*. All the other parameters of MILP solver are kept at their default settings, such as primal

TABLE II
SCALES AND COMPLEXITY OF CONSIDERED BENCHMARK TEST SYSTEMS

Items	6-bus	24-bus	46-bus	118-bus	300-bus
No. of buses: n_b	6	24	46	118	300
No. of candidate branches: n_c	15	41	79	186	411
No. of parallel circuits: K	4	2	3	2	1
No. of scenarios: $\lvert\mathcal{S}\rvert$	15	41	79	186	411
No. of binary variables: Kn_c	60	82	237	372	411
No. of continuous variables: $\lvert\mathcal{S}\rvert \cdot ((K+1)n_c + 3n_b)$	1395	7995	35 886	169 632	707 742
No. of equality constraints: $\lvert\mathcal{S}\rvert \cdot (n_b + n_c)$	315	2665	9875	56 544	292 221
No. of inequality constraints: $\lvert\mathcal{S}\rvert \cdot (5Kn_c + 4n_b n_c + n_c)$	10 125	179 867	1248 200	16 709 868	203 718 726
Data resources	[8]	[27]	[8]	[28]	[28]

TABLE III
COMPUTATIONAL RESULTS FOR TEST SYSTEMS WITH FIVE DIFFERENT TYPES OF METHODS

Algorithms	6-bus		24-bus		118-bus		46-bus		300-bus	
	Time (s)	Gap	Time (s)	Gap	Time (s)	Gap	Time (h)	Gap	Time (h)	Gap
MILP_dynamic (1 thread)	2.97	99.97%	2,201.59	32.75%	1,861.64	99.89%	48.00	18.53%	48.00	27.49%
MILP_dynamic (24 threads)	3.52	99.97%	**246.31**	**0.00%**	1,487.08	99.89%	32.27	17.42%	48.00	18.90%
MILP_traditional (1 thread)	5.72	99.97%	1,230.69	31.99%	709.75	99.89%	48.00	30.25%	48.00	33.16%
MILP_traditional (24 threads)	5.19	99.97%	317.80	27.98%	481.33	99.89%	33.55	16.65%	48.00	20.71%
BCBD_traditional (1 thread)	**0.47**	**0.00%**	1,516.10	21.07%	**62.65**	**0.00%**	**15.61**	**0.00%**	**48.00**	**5.41%**

heuristics, branching variable selection, and next node selection. On the other hand, since the lazy constraint callback function does not support the dynamic search and may not be thread safe, our algorithm is only implemented in sequential mode with the traditional pattern: *BCBD_traditional (1 thread)*.

A tolerance of $\epsilon = 10^{-6}$ is employed for all experiments. All algorithms are forced to terminate after exceeding a maximum run time of 48 h. Big-M is determined by the following equation in accordance with [13], [25], and [26]:

$$M_{ij} = 2\bar{\theta}\gamma_{ij} \tag{23}$$

where $\bar{\theta}$ is the maximum bus voltage angle.

A. Test Bed

In order to investigate the full potential of the considered methods and algorithms, five classical benchmark systems of different sizes are employed: the Garver 6-bus system, the IEEE 24-bus test system, the South Brazilian 46-bus system, the IEEE 118-bus test system, and the IEEE 300-bus test system. An overview of the scales and complexity are illustrated in Table II, as well as the available data resources.

The full feasible region is considered as the input for all algorithms. It should be pointed out that, for IEEE 118- and 300-bus test systems, the maximum capacity of each line has been reduced to 40% of the capacity given originally to increase the complexity of the problem [13]. Additionally, the original data set of [28] does not contain the price information; in this paper, an assumption on the transmission line investment cost, similar to [13], is adopted:

$$c_{ij} = 1000 L_{ij} \bar{f}_{ij} \tag{24}$$

where L_{ij} is the length of circuit ij.

B. Results

Unlike other algorithms with parameters, which need to be tuned before implementation, BCBD as well as the other four MILP solvers are totally parameter free. The comprehensive results of the five test systems are given in Table III, where the gap is defined by (25) and sampled when the fastest algorithm terminates:

$$\text{Gap} = (\text{UB} - \text{LB})/\text{UB}. \tag{25}$$

Table III is interpreted as follows according to different systems. Detailed performance comparison between various algorithms within one system will be given in Section IV-B3.

1) Garver 6-Bus System: Although K is valued as 4, the solution space of this small-scale system is still limited; thus, all methods achieve the convergence in seconds. With the help of B&C framework and acceleration strategies, the BCBD is almost 10× faster to arrive at the global optimal solution when compared with MILP solvers. Dynamic MILP algorithms significantly perform better than the traditional ones in this system.

2) IEEE 24-Bus Test System: This is a medium-scale system, all algorithms are able to achieve the global optimal solution within 40 min. It can be concluded from Table III that parallel implementation with 24 threads works much better than the sequential ones, but it is hard to distinguish which one performs better than another for the dynamic and the traditional mode. The convergence curves for different algorithms are illustrated in Fig. 6. BCBD converges faster than all the other four MILP solvers in the early stage, which is due to the two-phase method; however, the MILP solvers with 24 threads suddenly converged into the global optimal solution in the later stage, which is due to the built-in heuristics. Different with some efficient heuristics designed for specific problems, the built-in heuristics in Cplex

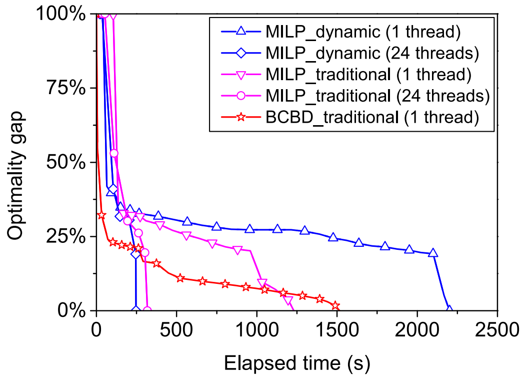


Fig. 6. Behavior of optimality gap for the IEEE 24-bus test system.

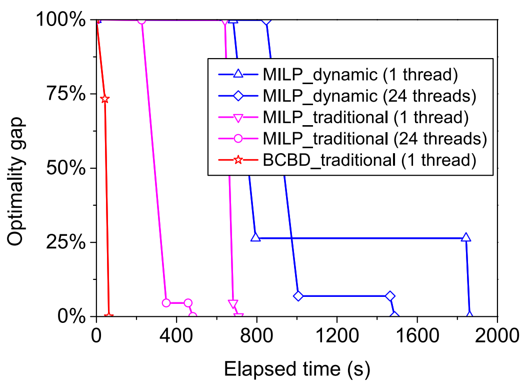


Fig. 7. Behavior of optimality gap for the IEEE 118-bus test system.

are designed for a general purpose; thus, their performance for SCTEP is not stable.

3) IEEE 118-Bus Test System: Although this system has a large solution space due to a large number of decision variables, it is not so much difficult when compared with the IEEE 24-bus system from the aspect of solution time. The reason is that the configuration is more sufficient for the 118-bus system, i.e., the number of circuits needed to be built is limited. Different from the results of the former system, BCBD shows its advantage over the other four MILP solvers in this system, where a speedup of 29.71, 23.74, 11.33, and 7.68 is gained, respectively. When BCBD terminates, an optimality gap of 99.89% is still holding for all the other methods, although two of them run in parallel with 24 threads.

It can be seen from Fig. 7 that several MILP solvers experience a long flat at the beginning, indicating that it is very hard to find a feasible solution for this problem. Another interesting phenomenon is the sharp decrease in the final stage, which is due to the special structure of this problem that it may be easy to derive the global optimal solution from lots of feasible solutions. BCBD can find a high-quality feasible solution with less effort by the help of all generated cuts from *Phase 1*; thus, it performs dramatically well for this problem. This can also be observed from the fact that the convergence process of BCBD is similar with the last stages of *MILP_traditional (1 thread)*.

4) South Brazilian 46-Bus System: This is a medium-scale real system, but its solution is more difficult than the 118-bus system. The fastest algorithm BCBD requires 15.61 h to meet the gap of 0.00%; in contrast, two MILP solvers could not even converge after running for 48 h: a gap of 22.80% and 11.22% is still remaining. It should be noted that parallel implementation plays a major role for MILP solvers in the solution of this test system.

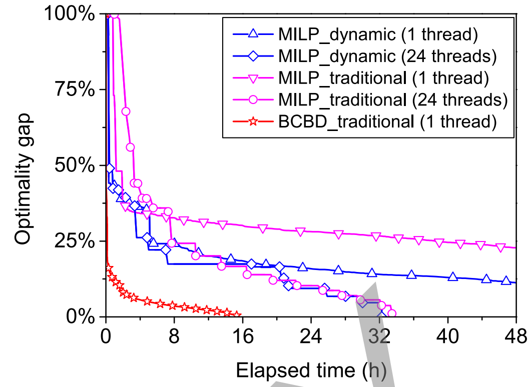


Fig. 8. Behavior of optimality gap for the South Brazilian 46-bus system.

TABLE IV
RANK TABLE FOR THE PERFORMANCE OF FIVE TYPES OF METHODS

Algorithms	6-bus	24-bus	118-bus	46-bus	300-bus	Sum
A	2	5	5	4	4	20
B	3	1	4	2	2	12
C	5	3	3	5	5	21
D	4	2	2	3	3	14
E	1	4	1	1	1	8

A: MILP_dynamic (1 thread) B: MILP_dynamic (24 threads)
C: MILP_traditional (1 thread) D: MILP_traditional (24 threads)
E: BCBD_traditional (1 thread)

The sequential BCBD is 16 h faster than the successfully converged MILP solvers, although they are implemented in parallel with 24 threads. Fig. 8 demonstrates the behavior of optimality gap for the 46-bus system. It can be seen that it takes almost 16 h for MILP solvers working in parallel mode to converge into a gap of 15%; however, it is just hundreds of seconds for BCBD, which is greatly due to the two-phase acceleration strategy. Both BCBD and the other two parallel MILP solvers spent nearly 16 h to fulfill the rest searching process, which proves that integrating BD into B&C framework is competitive when compared with parallel MILP solvers for really tough problems.

5) IEEE 300-Bus Test System: This large-scale system provides great challenges for all five methods since no method can finish the searching within 48 h. Nevertheless, the superiority of BCBD can still be identified by the finally achieved gaps. Although the last gap is only 5.41% for BCBD, it may takes tens of hours to arrive at 0.00% according to the previous experience for 46-bus system shown in Fig. 8. In terms of MILP solvers with larger termination gaps, requirement on the execution time and effort is much higher.

C. Discussion

1) Qualitative Evaluation: In order to distinguish which pattern and mode of MILP solver performs better for SCTEP, a qualitative evaluation is introduced in Table IV, where the

Circuit	ij	1-2	1-3	1-4	1-5	1-6	2-3	2-4	2-5	2-6	3-4	3-5	3-6	4-5	4-6	5-6	Execution Time (s)				
Cost	c_{ij}	40	38	60	20	38	20	40	31	30	59	20	48	63	30	61	A	B	C	D	E
$K=1$ (Infeasible)	n_{ij}^1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0.51	0.74	0.86	0.81	0.07
$K=2$ $(C=200)$	n_{ij}^2	0	0	0	0	0	0	0	0	1	0	1	0	0	1	0	1.94	2.23	2.70	2.66	0.24
	n_{ij}^1	0	0	0	0	0	0	1	0	1	0	1	0	0	1	0					
$K=3$ $(C=180)$	n_{ij}^3	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2.61	3.03	4.61	4.47	0.33
	n_{ij}^2	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0					
	n_{ij}^1	0	0	0	0	0	1	0	0	1	0	1	0	0	1	0					
$K=4$ $(C=180)$	n_{ij}^4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2.97	3.52	5.72	5.19	0.47
	n_{ij}^3	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0					
	n_{ij}^2	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0					
	n_{ij}^1	0	0	0	0	0	1	0	0	1	0	1	0	0	1	0					

A: MILP_dynamic (1 thread) B: MILP_dynamic (24 threads) C: MILP_traditional (1 thread)
D: MILP_traditional (24 threads) E: BCBD_traditional (1 thread)

Fig. 9. Solution configuration and execution time for different K values with the Garver 6-bus system.

number in each column represents the performance rank of the corresponding algorithm for each problem, i.e., the fastest one ranks 1 and the slowest one ranks 5. By adding the numbers for each row, there is $8_{(\text{row}E)} < 12_{(\text{row}B)} < 14_{(\text{row}D)} < 20_{(\text{row}A)} < 21_{(\text{row}C)}$. It can be concluded that BCBD performs better than the others across the five test systems. Two more conclusions can also be drawn for different versions of the MILP solver: 1) Parallel implementation works better than sequential, since $12_{(\text{row}B)} < 20_{(\text{row}A)}$ and $14_{(\text{row}D)} < 21_{(\text{row}C)}$ in dynamic and traditional modes, respectively; 2) dynamic mode performs better than traditional, as $20_{(\text{row}A)} < 21_{(\text{row}C)}$ and $12_{(\text{row}B)} < 14_{(\text{row}D)}$ in sequential and parallel environments, respectively.

2) Sensitivity Analysis for K: As shown in Table II, K has a large difference on the number of binary variables Kn_c, continuous variables $|\mathcal{S}| \cdot ((K+1)n_c + 3n_b)$, and inequality constraints $|\mathcal{S}| \cdot (5Kn_c + 4n_b n_c + n_c)$. Therefore, a sensitivity analysis on the solution configuration and efficiency for different K values based on the Garver 6-bus system is conducted in this section. Fig. 9 depicts all the results, which can be analyzed from two aspects.

1) Solution configuration: If $K = 1$, there is no feasible solution even if all candidate circuits are built. One reason is that the requirement of large number of circuits on key corridors, such as 4–6, is very hard to be replaced by other corridors. When $K = 2$, global optimal solution is accessible with a total cost of \$200 m. As K increases to 3, one more circuit can be built on corridor 4–6, which results in a cost reduction of \$20 m. Nevertheless, the cost cannot be reduced further by increasing K since $n_{ij}^4 = 0$ for all $ij \in \mathcal{C}$, which means the saturation point for the Garver 6-bus system is $K = 3$.
2) Solution efficiency: Discussion for $K = 1$ is skipped since there is no feasible solution. Comparing $K = 4$ with $K = 3$ and $K = 2$, it can be seen that the execution time increases for all algorithms as K increases, although $K = 4$ and $K = 3$ have the same optimal solution configuration. The reason is that a large K value represents larger solution space.

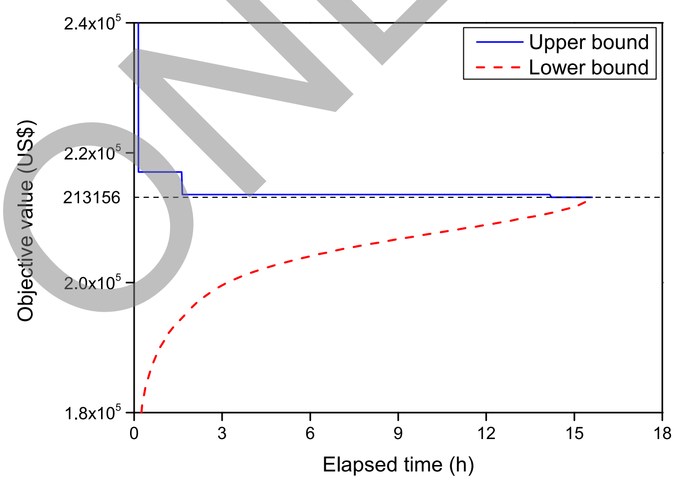


Fig. 10. Lower and upper bounds for BCBD of the South Brazilian 46-bus system.

D. Performance Analysis

Although the superiority of BCBD over MILP solvers has been revealed in the above-mentioned discussion, which component (B&C framework or acceleration strategies) facilitates the computational improvement is still not recognized. Therefore, further detailed comparison has been carried out in this section to figure out that issue.

Lower bound is employed to depict the convergence process. As shown in Section III-A, the lower bound is determined by the objective value of master problem. With the adding of cuts at each iteration, the master problem becomes more constrained; therefore, the objective value will increase monotonically, which provides a good parameter to describe the convergence characteristic. On the other hand, the upper bound fluctuates heavily since it is related with subproblems, where a large penalty may be triggered irregularly. Fig. 10 shows the lower and upper bounds for BCBD of the 46-bus system, where the fluctuated upper bound is flattened by lines 10–12 in Algorithm 2. Compared with the upper bound with long flat intervals, the lower bound provides more information about the convergence process.

TABLE V
DIFFERENT TYPES OF ALGORITHMS FOR PERFORMANCE ANALYSIS

Algorithms	Description
Alg.1	MILP_dynamic (24 threads)
Alg.2	CLBD without acceleration strategies
Alg.3	BCBD without acceleration strategies
Alg.4	CLBD without multicut strategy
Alg.5	BCBD without multicut strategy
Alg.6	CLBD with full acceleration strategies
Alg.7	BCBD with full acceleration strategies

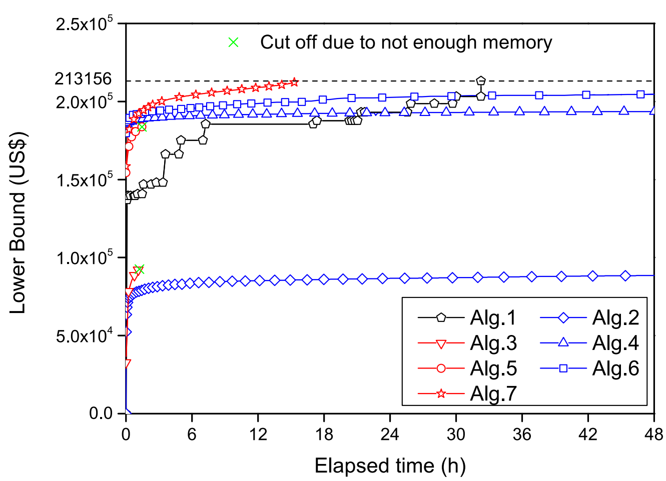


Fig. 11. Convergence properties of different algorithms for the South Brazilian 46-bus system.

As shown in Table III and Figs. 6–8, the solution processes of 6-, 24-, and 118-bus systems are short and not stable (e.g., frequently sharp decrease), whereas the execution time of 300-bus system is too long and the convergence is not guaranteed. Therefore, the 46-bus system is finally determined as the test bed due to its moderate convergence process and time consumption.

In order to distinguish the performance enhancement achieved from different components, seven types of algorithms are separated from BCBD and CLBD for comparison, which are listed in Table V. Fig. 11 illustrates the growth trend of the lower bound for Alg.1–7 on the 46-bus system, where the following findings can be observed.

1) Alg.7 is faster than Alg.1, showing that the B&C framework and acceleration strategies are successful for the MILP solution of SCTEP.
2) Alg.7 terminates at 16 h, whereas Alg.6 cannot find the global optimal until 48 h, indicating that the B&C framework plays an important role in the performance improvement.
3) Alg.6 converges faster than Alg.7 at the very beginning, the reason is that the incumbents in the searching tree of BCBD are not yet the optimum solution of the master problem; therefore, the cuts generated from subproblems cannot cut the feasible region efficiently. For CLBD, the optimal solution from master problem is always utilized.
4) Alg.6 performances better than Alg.1 in the first 30 h; however, MILP solver suddenly jumps to the global optimal at 33 h due to the dynamic search mechanism, which is also an evidence that the MILP solver has been greatly advanced in the last decades [9].
5) It has been revealed that BD may not work well without much additional enhancements; therefore, Alg.2 and Alg.3 show a weak performance.
6) Alg.4 and Alg.5 present good performance compared to Alg.2 and Alg.3 respectively, proving that the other three acceleration strategies except multicut contribute a lot in the iterative process.
7) The acceleration acquired by the introduction of multicut strategy is depicted by the fact that Alg.7 and Alg.6 perform better than Alg.5 and Alg.4.
8) One drawback of BCBD is that it may run out of memory when a huge B&C searching tree must be maintained, as shown by Alg.3 and Alg.5. On the other hand, CLBD has lower requirements on the computational resources even if it runs for 48 h.

To sum up, both the B&C framework and acceleration strategies contribute a lot to boost the efficiency of BCBD, making it perform better than the MILP solver and CLBD. Note that the searching tree of BCBD should be trimmed efficiently, otherwise it will grow to be very large and run out of the memory; therefore, the acceleration strategies are paramount for the success of BCBD. In terms of CLBD, the master problem is solved independently at each iteration, and the occupied memory will be freed when solving the subproblems.

V. CONCLUSION

To cope with the SCTEP problem, a BCBD algorithm is proposed by integrating BD into the B&C framework. Different with the CLBD framework, where the master problem is MILP, the BCBD replaces the MILP with LP, resulting in great reduction in computational resource and execution time. Four acceleration strategies have also been investigated to enhance the convergence efficiency as well as to restrict the solution space. Comprehensive computational experiments between BCBD and commercial MILP solver Cplex are conducted on five benchmark test systems ranging from 6 to 300 buses. Although parallel computing with 24 threads is enabled for some MILP solvers, the superiority of BCBD has been validated for the majority of systems. Detailed performance analysis has also been conducted to distinguish the performance improvements from B&C framework and acceleration strategies, where seven types of algorithms separated from BCBD and CLBD are involved. The results indicate that both the B&C framework and acceleration strategies contribute a lot to boost the efficiency of BCBD. Future work will be expanded to the combination of SCTEP and the generation expansion planning, the economic dispatch, and the unit commitment. In addition, new challenges confronted by modern power systems, such as the uncertainty from the intermittent renewable generators and transmission losses, will also be investigated.

REFERENCES

[1] H. Mavalizadeh, A. Ahmadi, F. H. Gandoman, P. Siano, and H. A. Shayanfar, "Multiobjective robust power system expansion planning considering generation units retirement," *IEEE Syst. J.*, vol. 12, no. 3, pp. 2664–2675, Sep. 2018.

[2] E. A. M. Cesea, T. Capuder, and P. Mancarella, "Flexible distributed multienergy generation system expansion planning under uncertainty," *IEEE Trans. Smart Grid*, vol. 7, no. 1, pp. 348–357, Jan. 2016.

[3] A. Hajebrahimi, A. Abdollahi, and M. Rashidinejad, "Probabilistic multiobjective transmission expansion planning incorporating demand response resources and large-scale distant wind farms," *IEEE Syst. J.*, vol. 11, no. 2, pp. 1170–1181, Jun. 2017.

[4] S. Dehghan, N. Amjady, and A. J. Conejo, "Reliability-constrained robust power system expansion planning," *IEEE Trans. Power Syst.*, vol. 31, no. 3, pp. 2383–2392, May 2016.

[5] J. D. Molina, J. Contreras, and H. Rudnick, "A risk-constrained project portfolio in centralized transmission expansion planning," *IEEE Syst. J.*, vol. 11, no. 3, pp. 1653–1661, Sep. 2017.

[6] A. Moreira, A. Street, and J. M. Arroyo, "An adjustable robust optimization approach for contingency-constrained transmission expansion planning," *IEEE Trans. Power Syst.*, vol. 30, no. 4, pp. 2013–2022, Jul. 2015.

[7] V. S. K. M. Balijepalli and S. A. Khaparde, "A holistic approach for transmission system expansion planning studies: An Indian experience," *IEEE Syst. J.*, vol. 5, no. 2, pp. 199–212, Jun. 2011.

[8] R. Romero, A. Monticelli, A. Garcia, and S. Haffner, "Test systems and mathematical models for transmission network expansion planning," *Proc. IEE Gener., Transmiss., Distrib.*, vol. 149, no. 1, pp. 27–36, Jan. 2002.

[9] H. Zhang, V. Vittal, G. T. Heydt, and J. Quintero, "A mixed-integer linear programming approach for multi-stage security-constrained transmission expansion planning," *IEEE Trans. Power Syst.*, vol. 27, no. 2, pp. 1125–1133, May 2012.

[10] S. Asadamongkol and B. Eua-arporn, "Application of Benders decomposition to transmission expansion planning with N-1 security constraints," *IEEJ Trans. Elect. Electron. Eng.*, vol. 6, no. 2, pp. 127–133, Mar. 2011.

[11] S. Lumbreras and A. Ramos, "Transmission expansion planning using an efficient version of Benders' decomposition. A case study," in *Proc. IEEE PowerTech*, Grenoble, France, Jun. 2013, pp. 1–7.

[12] M. Jenabi, S. M. T. Fatemi Ghomi, S. A. Torabi, and S. H. Hosseinian, "Acceleration strategies of Benders decomposition for the security constraints power system expansion planning," *Ann. Oper. Res.*, vol. 235, no. 1, pp. 337–369, Dec. 2015.

[13] O. Alizadeh-Mousavi and M. Zima-Bočkarjova, "Efficient Benders cuts for transmission expansion planning," *Elect. Power Syst. Res.*, vol. 131, pp. 275–284, Feb. 2016.

[14] S. Huang and V. Dinavahi, "Security constrained transmission expansion planning by accelerated benders decomposition," in *Proc. North Amer. Power Symp.*, Denver, CO, USA, Sep. 2016, pp. 1–6.

[15] A. J. Wood, B. F. Wollenberg, and G. B. Sheble, *Power Generation, Operation, and Control*, 3rd ed. New York, NY, USA: Wiley, 2013.

[16] I. de J. Silva, M. J. Rider, R. Romero, A. V. Garcia, and C. A. Murari, "Transmission network expansion planning with security constraints," *IEE Proc. Gener., Transmiss. Distrib.*, vol. 152, no. 6, pp. 828–836, Nov. 2005.

[17] K. W. Hedman, M. C. Ferris, R. P. O'Neill, E. B. Fisher, and S. S. Oren, "Co-optimization of generation unit commitment and transmission switching with N-1 reliability," *IEEE Trans. Power Syst.*, vol. 25, no. 2, pp. 1052–1063, May 2010.

[18] J. F. Benders, "Partitioning procedures for solving mixed-variables programming problems," *Numer. Math.*, vol. 4, no. 1, pp. 238–252, 1962.

[19] C. C. Carøe and R. Schultz, "Dual decomposition in stochastic integer programming," *Oper. Res. Lett.*, vol. 24, no. 1, pp. 37–45, Feb. 1999.

[20] M. Shahidehopour and Y. Fu, "Benders decomposition: Applying benders decomposition to power systems," *IEEE Power Energy Mag.*, vol. 3, no. 2, pp. 20–21, Mar. 2005.

[21] E. S. Thorsteinsson, "Branch-and-check: A hybrid framework integrating mixed integer programming and constraint logic programming," in *Proc. Int. Conf. Princ. Pract. Constraint*, Berlin, Germany, 2001, vol. 2239, pp. 16–30.

[22] J. F. Cordeau, G. Stojković, F. Soumis, and J. Desrosiers, "Benders decomposition for simultaneous aircraft routing and crew scheduling," *Transp. Sci.*, vol. 35, no. 4, pp. 375–388, Nov. 2001.

[23] F. You and I. E. Grossmann, "Multicut Benders decomposition algorithm for process supply chain planning under uncertainty," *Ann. Oper. Res.*, vol. 210, no. 1, pp. 191–211, Nov. 2013.

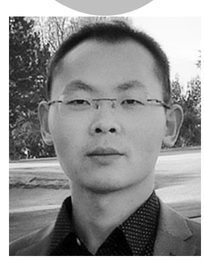
[24] F. D. Munoz, B. F. Hobbs, and J.-P. Watson, "New bounding and decomposition approaches for MILP investment problems: Multi-area transmission and generation planning under policy constraints," *Eur. J. Oper. Res.*, vol. 248, no. 3, pp. 888–898, Feb. 2016.

[25] L. Bahiense, G. C. Oliveira, M. Pereira, and S. Granville, "A mixed integer disjunctive model for transmission network expansion," *IEEE Trans. Power Syst.*, vol. 16, no. 3, pp. 560–565, Aug. 2001.

[26] M. Rahmani, R. Romero, and M. J. Rider, "Strategies to reduce the number of variables and the combinatorial search space of the multistage transmission expansion planning problem," *IEEE Trans. Power Syst.*, vol. 28, no. 3, pp. 2164–2173, Dec. 2013.

[27] R. Fang and D. J. Hill, "A new strategy for transmission expansion in competitive electricity markets," *IEEE Trans. Power Syst.*, vol. 18, no. 1, pp. 374–380, Feb. 2003.

[28] R. D. Zimmerman, C. E. Murillo-Sanchez, 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, Feb. 2011.



Shengjun Huang (S'14) received the B.Sc. and M.Sc. degree in management science and engineering from the National University of Defense Technology, Changsha, China, in 2011 and 2013, respectively, and is currently working toward the Ph.D. degree at the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada.

His research interests include mixed-integer linear programming, decomposition algorithms, large-scale power systems, and parallel computing.

Venkata Dinavahi (SM'08) received the Ph.D. degree in electrical and computer engineering from the University of Toronto, Toronto, ON, Canada, in 2000.

He is a Professor with the Department of Electrical and Computer Engineering, University of Alberta, Edmonton, AB, Canada. His research interests include real-time simulation of power systems and power electronic systems, large-scale system simulation, and parallel and distributed computing.