A BRANCH-AND-CUT ALGORITHM FOR CHANCE-CONSTRAINED MULTI-AREA RESERVE SIZING

Jehum Cho, Anthony Papavasiliou

LIDAM Discussion Paper CORE
2022 / 18

UCLouvain

LIDAM
CORE | Louvain Institute of Data Analysis and Modeling in economics and statistics

CORE

Voie du Roman Pays 34, L1.03.01

B-1348 Louvain-la-Neuve

Tel (32 10) 47 43 04

Email: immaq-library@uclouvain.be

https://uclouvain.be/en/research-institutes/lidam/core/core-discussion-papers.html

A Branch-and-Cut Algorithm for Chance-Constrained Multi-Area Reserve Sizing

Jehum Cho
Anthony Papavasiliou
Center of Operations Research and Econometrics (CORE)
Université catholique de Louvain (UCL)
Louvain-la-Neuve, Belgium

***Abstract*—We implement an exact mixed-integer programming algorithm for the chance-constrained multi-area reserve sizing problem in the presence of transmission network constraints. The problem can be cast as a two-stage stochastic mixed integer linear program using sample approximation. Due to the complicated structure of the problem, existing methods attempt to find a feasible solution based on heuristics. However, a recent development of integer programming techniques allow us to reformulate the problem into a form where we can solve it to optimality. In this paper, we apply this integer programming approach to solve our problem to optimality and compare the results with that of the existing heuristics.**

***Index Terms*—Multi-area reserve sizing, chance constraints, probabilistic constraints, mixed-integer programming**

I. Introduction

The level of the cooperation of transmission system operators (TSOs) in European electricity markets is becoming higher including the coordinated international trade of reserve capacity within European day-ahead electricity markets (article 40-42 [1]). In this work, our focus is on the dimensioning of automatic (aFRR) and manual frequency restoration reserves (mFRR). aFRR resources are driven by automatic controllers which respond to system frequency, and are expected to obey a full activation time of 5 minutes in the future European-wide PICASSO platform for aFRR activation. mFRR resources are driven by manual activation, are used in order to relieve aFRR, and are expected to be fully activated within 15 minutes in the future European-wide MARI platform for mFRR activation.

There is a wide literature on the topic of dimensioning frequency restoration reserve using probabilistic criteria [2]. This literature is largely focused on European electricity markets (see, for instance, [3] for applications in Germany and [4] for applications in the Central Western Continental block), since the probabilistic criterion is enshrined in European legislation, in particular the System Operation Guideline (article 157 [5]). Note, however, that the aforementioned literature does not account for the impact of transmission constraints on the sharing and exchange of reserves.

Reserve exchange refers to the exclusive use of reserves by a system operator which is different from the system operator that hosts the reserves in question. Reserve sharing refers to the joint use of the same reserve capacity by multiple system operators in order to cover reserve requirements, which counts on the fact that the need to activate these reserves simultaneously is highly unlikely [6]. Our work can best be classified as relating to the sizing of reserve in the context of reserve sharing.

Our formulation bears certain similarities to the literature on reserve deliverability which has been developed in the context of US market operations. Note, however, that the literature on reserve deliverability often considers reserve requirements as being defined exogenously [7], and does not account for the endogenous sizing of reserve. Other reserve deliverability literature does account for endogenous sizing, however it does not account for chance constraints corresponding to a probabilistic criterion [8], [9], or relies on approximations [10] which are not required in the exact method that is presented in our work. Research which is concerned with the deliverability of balancing energy has also been developed in [11], however the actual reserve requirements in this work are also considered to be exogenous input.

Although the two-stage stochastic mixed-integer linear programming formulation in [6] captures many realistic requirements including the uncertainty driven by both imbalances and transmission capacity, their methodology can only find a feasible solution using a heuristic method based on an LP-relaxation of the original formulation. Under the philosophy of the sample approximation approach in [12], their formulation uses a so-called "big-M method" in order to make binary variables represent whether each sample violates the chance constraints.

This "big-M method" creates a substantial gap between the MILP solution and the LP-relaxation solution, preventing the branch-and-bound algorithm from obtaining an optimal solution especially with a high number of samples. In this work, being inspired by a recent development of integer programming [14], we reformulate this problem and apply a decomposition method with strong valid inequalities in order to solve the problem to optimality.

II. Overview

For a network $\mathcal{G}(Z, E)$, let r_z denote the size of (positive) reserve for each area z. Given a positive integer n, let us denote $[n]$ as the set $\{1, \ldots, n\}$. For $i \in [N]$, δ_{iz} denotes the imbalance of sample i at the area z, and u_i denotes the binary variable representing whether r is in F_i, where F_i is the feasible region of r with sample i defined as (1). Here, p_z and f_e is the amount of balancing energy at z and the flow from z_1 to z_2 when $e = (z_1, z_2)$ under capacity constraints with T_e^+ and T_e^-.

$$
\begin{aligned}
F_i = \{r \in \mathbb{R}_+^{|Z|} : \exists (p, f) \text{ s.t.} & \\
p_z + \delta_{iz} = \sum_{e=(z,\cdot)\in E} f_e - \sum_{e=(\cdot,z)\in E} f_e, \quad & \forall z \in Z : \lambda_z \\
p_z \le r_z, \quad & \forall z \in Z : \pi_z \\
f_e \ge -T_e^-, \quad & \forall e \in E : \mu_e^- \\
f_e \le T_e^+, \quad & \forall e \in E : \mu_e^+ \}
\end{aligned} \tag{1}
$$

Our problem can be written with logical constraints (3) as follows.

$$\min \sum_{z\in Z} r_z \tag{2}$$

$$\text{s.t. } u_i = 0 \implies r \in F_i, \quad \forall i \in [N] \tag{3}$$

$$\sum_{i\in N} u_i \le \lfloor \epsilon N \rfloor \tag{4}$$

$$r \ge 0, u \in \{0,1\}^N \tag{5}$$

A common way to formulate this problem is to use "big-M method" in order to represent the logical expression (3). There can be several ways, but one way to formulate it is as follows:

$$
\begin{aligned}
\min & \sum_{z\in Z} r_z \\
\text{s.t. } & p_{iz} + \delta_{iz} = \sum_{e=(z,\cdot)\in E} f_{ie} - \sum_{e=(\cdot,z)\in E} f_{ie}, \quad \forall z \in Z, i \in [N] \\
& p_{iz} \le r_z + l_{iz}, \quad \forall z \in Z, i \in [N] \\
& l_{iz} \le \max\{0, -\delta_{iz}\} \cdot u_i, \quad \forall z \in Z, i \in [N] \\
& -T_e^- \le f_{ie} \le T_e^+, \quad \forall e \in E, i \in [N] \\
& \sum_{i\in N} u_i \le \lfloor \epsilon N \rfloor \\
& r \ge 0, u \in \{0,1\}^N
\end{aligned} \tag{6}
$$

Here, $l_{iz}, \forall z \in Z$ are equal to zero when u_i is equal to zero, which implies $r \in F_i$. When u_i is equal to 1, l_{iz} allows p_{iz} to satisfy the power balance constraint for the sample i even when r_z is equal to zero. In worst case scenario, p_{iz} has to be equal to $-\delta_{iz}$ in order to make the problem feasible. So, the the coefficient of u_i for the constraint linking l_{iz} and u_i has to be greater than equal to $\max\{0, -\delta_{iz}\}$. This type of constraints represents logical expressions with a large number M ($\max\{0, -\delta_{iz}\}$ for our problem); hence the name "big-M method". With this additional variable l which connects F_i with u_i, our problem can be formulated into linear programming. However, it has several drawbacks to solve (6) directly with Branch-and-Bound algorithm using a commercial solver. First, the problem is too large containing variables and constraints for all the sample set $[N]$. Second, it is well known that the LP-relaxation of a "big-M" style formulation is not tight, resulting in bad performance for Branch-and-Bound algorithm.

Instead of formulating (3) with a "big-M method", in this paper, we solve the problem using Benders decomposition. We reformulate (3) into a form of $(r, u) \in R$, where R is a polyhedron formed with the valid inequalities induced by Benders cuts. The Benders cuts for representing $r \in F_i$ can be obtained by solving an LP based on the dual form of F_i. Then we modify these cuts to represent the logical expression in (3). There are several layers of this modification. We start from the standard mixing set and incorporate the star inequalities in order to strengthen the cuts as introduced in Theorem 1 of [14]. We go one step further by using the extended formulation introduced in [13]. In section III, we elaborate these layers of modification of the cuts step by step. In section IV, we provide a diagram for all the procedures of Branch-and-Cut algorithm using these cuts from [14], which achieves an optimal solution. In section V, we show our computation results comparing our method and (6).

III. Benders Decomposition

A. Standard Benders Cuts

Introduced by Benders in [15], the standard Benders Cuts are used in order to decompose a large problem into smaller problems by dividing variables into different groups. So, it is often used for solving stochastic programming with recourse whose variables can be easily grouped by stages. In this way, the second stage variables can be decomposed into even smaller problems for each scenario. They have two types: Benders Optimality Cuts and Benders Feasibility Cuts. The former connect the effects of the second stage variables to the objective function with the first stage variables, and the latter are used for representing the feasible region of the first stage variables depending on second stage variables. Since the second stage variables (p, f) in our problem do not show up in the objective function (2), and what we need is to check $r \in F_i$, we only utilize Benders Feasibility Cuts throughout the paper.

For our problem, Benders Feasibility Cuts can be obtained by solving (7), which is the dual problem of the feasibility checking problem for F_i when $\hat{r}$ is given.

$$
\begin{aligned}
v_i(\hat{r}) = \max & - \sum_{z\in Z} (\hat{r}_z \pi_z + \delta_{iz} \lambda_z) - \sum_{e\in E} (T_e^+ \mu_e^+ + T_e^- \mu_e^-) \\
\text{s.t. } & \pi_z - \lambda_z = 0, \quad \forall z \in Z \\
& \lambda_{z_t} - \lambda_{z_s} - \mu_e^+ + \mu_e^- = 0, \quad \forall e = (z_s, z_t) \in E \\
& \pi, \lambda \in [0,1]^{|Z|}, \mu^+, \mu^- \in [0,1]^{|E|}
\end{aligned} \tag{7}
$$

Observe that the dual variables $\pi, \lambda, \mu^+, \mu^-$ correspond to certain constraints of (1). If $v_i(\hat{r}) > 0$, then $\hat{r} \notin F_i$. Let $(\hat{\pi}, \hat{\lambda}, \hat{\mu}^+, \hat{\mu}^-)$ be an optimal extreme point solution of (7). Then, for

$$
\begin{aligned}
\alpha &= \hat{\pi}, \\
\beta &= -\delta_i \hat{\lambda} - T^+ \hat{\mu}^+ - T^- \hat{\mu}^-,
\end{aligned} \tag{8}
$$

$\alpha \hat{r} < \beta$ and $\alpha r \ge \beta$, for all $r \in F_i$. We call this $\alpha r \ge \beta$ as a Benders Feasibility Cut.

B. Modified Benders Cuts[1]

The standard Benders Cuts are not directly applicable to our problem. The Benders Feasibility Cuts represent $r \in F_i$, but what we need is the cuts which represent the logical expression in (3). For that, we need the following optimization problem for each scenario[2]:

$$h_i(\alpha) := \min\{\alpha r | r \in F_i\}. \tag{9}$$

Here, $\alpha r + h_i(\alpha)u_i \geq h_i(\alpha)$ is a valid inequality for (3), because if $u_i = 0, r \in F_i$, then $\alpha r \geq h_i(\alpha)$ by definition of $h_i(\alpha)$. Let us define the mixing set with these valid inequalities as follows:

$$\begin{aligned} P(\alpha) := \{(r,u) \in \mathbb{R}_+^{|Z|} \times \{0,1\}^N :& \\ \alpha r + h_i(\alpha)u_i \geq h_i(\alpha), \forall i \in [N]\}.& \end{aligned} \tag{10}$$

Let us denote Π as the set of all the possible α, which are extreme points of (7). Notice that for our problem F_i for all $i \in [N]$ shares the common constraints set of (7), resulting in the same set of extreme points; hence the same Π for $F_i, \forall i \in [N]$.

Theorem 1. $\bigcap_{\alpha \in \Pi} P(\alpha)$ *is equivalent to* $\{(r,u) : (3), (5)\}$.

Proof. Since all the inequalities in the set $P(\alpha)$ are valid for (3), $\{(r,u) : (3), (5)\} \subseteq P(\alpha)$ for all $\alpha \in \Pi$. For $(\hat{r}, \hat{u})$ such that $\hat{u}_k = 0$ and $\hat{r} \notin F_k$, $\exists(\alpha, \beta)$ s.t. $\alpha\hat{r} < \beta$ but $\alpha\hat{r} \geq \beta$ is valid for F_k. By definition of $h_k(\alpha)$, for all $r \in F_k, \alpha r \geq h_k(\alpha) \geq \beta > \alpha\hat{r}$. So, $(\hat{r}, \hat{u})$ is violated by $\alpha r + h_k(\alpha)u_k \geq h_k(\alpha)$. Thus, $\bigcap_{\alpha \in \Pi} P(\alpha) \subseteq \{(r,u) : (3), (5)\}$. ☐

Theorem 1 implies that the logical expression (3) can be replaced by the modified Benders Cuts. These cuts can be strengthened by including the cardinality constraint (4). For the simplicity of notation, let us denote $q = \lfloor \epsilon N \rfloor$, and for all $\alpha \in \Pi$, σ^α is a permutation of N integers such that

$$h_{\sigma_1^\alpha}(\alpha) \geq h_{\sigma_2^\alpha}(\alpha) \geq \cdots \geq h_{\sigma_N^\alpha}(\alpha).$$

Then, $\alpha r + (h_{\sigma_i^\alpha}(\alpha) - h_{\sigma_{q+1}^\alpha}(\alpha))u_{\sigma_i^\alpha} \geq h_{\sigma_i^\alpha}(\alpha)$ for all $i \in [q]$ is valid to $\{(r,u) : (3), (4), (5)\}$. This is because for at least one of the $q+1$ largest values of $h_i(\alpha)$, $u_i = 0$ and this implies $\alpha r \geq h_{\sigma_{q+1}^\alpha}(\alpha)$ is always valid. This argument is from Lemma 1 of [14]. Let us define the mixing set with these new valid inequalities as follows:

$$\begin{aligned} &P'(\alpha) := \{(r,u) \in \mathbb{R}_+^{|Z|} \times \{0,1\}^N : (4), \\ &\alpha r + (h_{\sigma_i^\alpha}(\alpha) - h_{\sigma_{q+1}^\alpha}(\alpha))u_{\sigma_i^\alpha} \geq h_{\sigma_i^\alpha}(\alpha), \forall i \in [q]\}. \end{aligned} \tag{11}$$

Theorem 2. $\bigcap_{\alpha \in \Pi} P'(\alpha)$ *is equivalent to* $\{(r,u) : (3), (4), (5)\}$.

[1]This idea comes from [14], but in this paper we show alternative interpretation of these cuts by emphasizing the link with Standard Benders Cuts and proving Theorem 1 to show equivalence of the mixing set (10) and the logical expression (3).

[2]Observe that F_i is nonempty, the recession cone of F_i is the positive orthant, whose dual cone is also the positive orthant. Since $\alpha = \hat{\pi}$ is a vector in the positive orthant as well, $h_i(\alpha)$ exists and is finite.

Proof. It is easy to see that $\{(r,u) : (3), (4), (5)\} \subseteq P'(\alpha)$ for all $\alpha \in \Pi$, as in the proof of Theorem 1. Similarly, for $(\hat{r}, \hat{u})$ such that $\hat{u}_k = 0$ and $\hat{r} \notin F_k$, $\exists(\alpha, \beta)$ s.t. $\alpha\hat{r} < \beta$ but $\alpha\hat{r} \geq \beta$ is valid for F_k. Here, we need to consider two cases. If $h_{\sigma_{q+1}^\alpha}(\alpha) \geq h_k(\alpha)$, since $h_{\sigma_i^\alpha}(\alpha) - (h_{\sigma_i^\alpha}(\alpha) - h_{\sigma_{q+1}^\alpha}(\alpha))\hat{u}_{\sigma_i^\alpha} \geq h_{\sigma_{q+1}^\alpha}(\alpha) \geq h_k(\alpha) \geq \beta > \alpha\hat{r}$ for all $i \in [q]$. If $h_{\sigma_{q+1}^\alpha}(\alpha) < h_k(\alpha)$, then $k = \sigma_i^\alpha$ for some $i \in [q]$, and $h_{\sigma_k^\alpha}(\alpha) - (h_{\sigma_k^\alpha}(\alpha) - h_{\sigma_{q+1}^\alpha}(\alpha))\hat{u}_{\sigma_i^\alpha} \geq h_{\sigma_{q+1}^\alpha}(\alpha) \geq h_k(\alpha) \geq \beta > \alpha\hat{r}$. So, there exists some inequalities in (11) which violate $(\hat{r}, \hat{u})$. Thus, $\bigcap_{\alpha \in \Pi} P'(\alpha) \subseteq \{(r,u) : (3), (4), (5)\}$. ☐

Theorem 2 shows that we can use these stronger valid inequalities to represent the logical expression (3) thanks to the cardinality constraint (4). Notice the difference between (10) and (11). The former has N inequalities whereas the latter has q inequalities, and generally $q << N$ since we consider $\epsilon \leq 0.01$. Now, we apply the star inequalities of [16], or equivalently the mixing inequalities of [17] as in [14].

Theorem 3. *([14],[16],[17]) Let* $T = \{t_1, t_2, \ldots, t_l\} \subseteq \{\sigma_1^\alpha, \ldots, \sigma_q^\alpha\}$ *be such that* $h_{t_i}(\alpha) \geq h_{t_{i+1}}(\alpha)$ *for* $i \in [l]$, *where* $h_{t_{l+1}}(\alpha) = h_{\sigma_{q+1}^\alpha}(\alpha)$. *Then the inequality*

$$\alpha r + \sum_{i=1}^{l}(h_{t_i}(\alpha) - h_{t_{t+1}}(\alpha))u_{t_i} \geq h_{t_1}(\alpha) \tag{12}$$

if valid for $P'(\alpha)$.

Clearly, (12) dominate the inequalities in (11), and they are known for their strength because they define the convex hull of the mixing set $P'(\alpha)$ without cardinality constraint (4). See ([14],[16],[17]) for more details.

Also, the inequalities in (11) are special cases of (12) when T is singleton. So, $Q(\alpha)$ is equivalent to $P'(\alpha)$, where

$$Q(\alpha) := \{(r,u) \in \mathbb{R}_+^{|Z|} \times \{0,1\}^N : (4), (12)\}. \tag{13}$$

Even though, the number of constraints in (12) grows exponentially with respect to q, for a given $(\hat{r}, \hat{u})$ separation of these inequalities can be accomplished efficiently ([16],[17]). So, in [14], they used these inequalities for their Branch-and-Cut algorithm with the separation problem. In this paper, we move a bit more forward by using an extended formulation from [13].

C. Extended Formulation

Let us first define the extended formulation of Q with q additional binary variables w^α as follows:

$$EQ(\alpha) := \{(r,u,w) \in \mathbb{R}_+^{|Z|} \times \{0,1\}^{N+q} : (4), (15) - (17)\}, \tag{14}$$

where

$$\alpha r + \sum_{i=1}^{q}(h_{\sigma_i^\alpha}(\alpha) - h_{\sigma_{i+1}^\alpha}(\alpha))w_i^\alpha \geq h_{\sigma_1^\alpha} \tag{15}$$

$$w_i^\alpha - w_{i+1}^\alpha \geq 0, \qquad \forall i \in [q-1] \tag{16}$$

$$u_{\sigma_i^\alpha} - w_i^\alpha \geq 0, \qquad \forall i \in [q]. \tag{17}$$

We can show that $EQ(\alpha)$ is also equivalent to $Q(\alpha)$ in the sense as follows.

Theorem 4. *([13])* $Proj_{(r,u)}(EQ(\alpha)) = Q(\alpha)$. *Moreover, the projection of the linear relaxation of* $EQ(\alpha)$ *is equal to the linear relaxation of* $Q(\alpha)$.

So, instead of solving the separation problem for (12), by utilizing the extended formulation $EQ(\alpha)$, we can have the same effect as adding the whole exponential family of valid inequalities (12).

IV. Branch-and-Cut Algorithm

So far, we have explored ways to represent $\{(r,u) : (3),(4),(5)\}$ by modifying the standard Benders Cuts for F_i. Since the size of Π is too big to handle with a one-shot optimization problem, to solve our problem to optimality, a procedure updating α which cuts off incumbent solutions is necessary. This procedure should be combined with the Branch-and-Bound algorithm to get an integer solution for u. Traditionally, this type of combination is called a Branch-and-Cut algorithm because in the course of Branch-and-Bound algorithm, certain cuts (in our paper Benders Cuts) are added. One of the Branch-and-Cut algorithms using (12) for a general two-stage stochastic problem is well documented in [14]. Here, we provide a diagram to describe our Branch-and-Cut algorithm specialized for our problem which is a slightly modified version using $EQ(\alpha)$.

A. Outline

Figure 1 shows the diagram illustrating the procedures of the Branch-and-Cut algorithm for our problem. In this subsection, we briefly explain the notation and the general flow of the algorithm. Formal descriptions for the Master Problem and the Separation Problem can be found in the following subsections.

Branch-and-Bound algorithm is basically a tree-search method where we utilize the information of upper bounds and lower bounds of incumbent problems (denoted as Master Problem in the diagram) in order to narrow down the searching space. In Figure 1, $OPEN$ denotes the set of nodes that we still need to explore. At each node of the searching tree (l), we fix binary variables to the certain value (0 or 1); $N_0(l)$ is the set of variables fixed to 0 at node l, $N_1(l)$ the set of variables fixed to 1 at node l. The linear relaxation of the incumbent problem is solved at each node, which provides a lower bound (denoted as lb) for the original optimization problem. At the leaf of the tree or in the middle of the tree by coincidence, sometimes we get an (mixed) integer solution which gives an upper bound for the underlying optimization problem (denoted as U in our diagram) if that solution is feasible to F_i for all $i \in [N]$. Notice that when the lower bound for a certain node is higher than the current upper bound (U), it implies that we no longer need to explore that node. In this manner, the searching space gets narrowed down. This is what the rhombus ($lb \geq U$?) does in the diagram.

When the lower bound of the incumbent optimization problem is lower than U, it is still worth exploring that node. So, unless it happens to find an (mixed) integer solution, we branch the node by selecting a non-binary solution $\hat{u}_k$ and adding two new nodes to $OPEN$, where u_k is either fixed to 0 or 1 (see the bottom rectangle of the diagram).

Fig. 1. A Diagram for Branch-and-Cut Algorithm

If $\hat{u}$ is binary, the current solution is a candidate for being feasible to the original optimization problem. But, we still need to check if the current solution is feasible to F_i for all $i \in [N]$. That is what Separation Problem does in the diagram. It checks if there is any $k \in [N]$ such that $\hat{r} \notin F_k$ and $u_k = 0$. CUTFOUND is a boolean whose value is TRUE if there exists such k and FALSE if there is none. When CUTFOUND = TRUE, Separation Problem returns α which cuts off the current solution. This α is added to $\bar{\Pi}$ to update Master Problem. When CUTFOUND = FALSE, it means that the current solution is feasible to the original optimization problem, so we update U with the optimal objective function value for the incumbent problem Master Problem (l), which is lb. This process continues until there are no unexplored nodes left in the set $OPEN$. Then, the value U when the algorithm terminates is the optimal objective function value for our problem, and it is possible to store the optimal solution when we update U.

B. Master Problem

Here, we formalize what we solve as Master Problem. Master Problem (l) solves as follows:

$$\begin{aligned}
&\min \sum_{z\in Z} r_z \\
&\text{s.t. } \sum_{i\in N} u_i \le q \\
&\quad (15)-(17), \quad \forall\alpha \in \bar{\Pi} \\
&\quad u_k = 0, k \in N_0(l), u_k = 1, k \in N_1(l) \\
&\quad r \ge 0, u \in [0,1]^N, w^\alpha \in [0,1]^q, \forall \alpha \in \bar{\Pi}.
\end{aligned} \tag{18}$$

For the initial Master Problem (0), $\bar{\Pi}, N_0(l), N_1(l) = \emptyset$. So the optimal solution is $r_z = 0, \forall z \in Z$. On the course of the Branch-and-Cut Algorithm, the set $\bar{\Pi}$ is continually updated, so that (18) becomes close to the original optimization problem until it finds the optimal solution.

After we solve (18), when it is infeasible Master Problem (l) returns $lb = \infty$, which results in that the optimal U becomes ∞. This implies our original problem is infeasible. When (18) finds a feasible optimal solution, it returns the the optimal objective function value as lb, and corresponding optimal solution $(\hat{r}, \hat{u})$. Notice that the extended variable w is only used for tightening (18), and it is no longer used in the remaining procedure.

C. Separation Problem

As briefly stated in the outline, Separation Problem $(\hat{r}, \hat{u})$ checks if the current solution is feasible to the original optimization problem, and if not it returns α which cuts off this solution.

This is done by solving $(7), \forall i \in [N]$ such that $\hat{u}_i = 0$. When $\hat{u}_i = 1$, it implies that $\hat{r}$ does not have to be feasible to F_i for that certain solution. If $v_i(\hat{r}) > 0$, then $\hat{r} \notin F_i$ and $\alpha = \hat{\pi}$ for (7). Notice that this procedure can be parallelized for each i. If Separation Problem finds k such that $v_k(\hat{r}) > 0$, then it returns a boolean CUTFOUND as TRUE with corresponding α, otherwise it returns CUTFOUND as FALSE.

When CUTFOUND = TURE, α is added to $\bar{\Pi}$. Here, it is possible to find just one single such α or instead screen all $i \in [N]$ such that $\hat{u}_i = 0$ and add all α for all k such that $v_k(\hat{r}) > 0$). Notice that when we update Master Problem, we need to calculate $h_{\sigma_i^\alpha}, \forall i \in [N]$ and this can be also parallelized.

V. Computational Results

We compare the formulation of (6) and our Branch-and-Cut algorithm. For the underlying Branch-and-Bound algorithm, we used a commercial solver Gurobi 9.1. Here, in order to compare the power of different formulations, we did not use parallelization for Separation Problem. As simulation data, four different networks in Figure 2 are used for the computational experiments. For each network, we tested three levels of sample size : 3000,4000, and 5000. For each sample $i \in [N]$, δ_{iz} is sampled from a normal distribution with mean = 0, standard deviation = 100 for each zone $z \in Z$. For capacity constraints[3], $[T^+_{e_1}, T^+_{e_2}, T^+_{e_3}] = [50, 80, 20], [T^-_{e_1}, T^-_{e_2}, T^-_{e_3}] = [40, 30, 90]$. For each setting, we test 100 instances and check how many instances are solved to optimality within the time limit (1800 seconds).

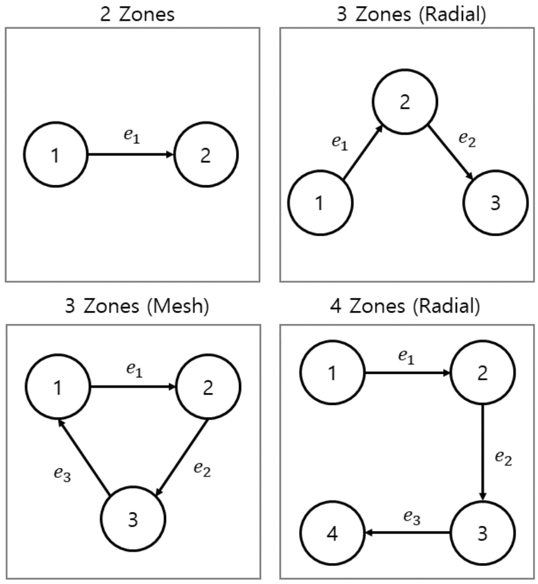

Fig. 2. Networks for Simulation

Figure 3 shows the result of the computational experiments. It shows the number of unsolved instances among 100 test instances by two different methods (BigM:(6), B&C:Branch-and-Cut Algorithm) for different networks in Figure 2. Observe that in every setting, the number of unsolved instances by B&C is lower than that by BigM. Even when BigM cannot solve more than 90% of instances, B&C solves more than 80% of instances. Notice that even though the effect of stronger valid inequalities (tight formulation of (14)) enhanced the performance of Branch-and-Bound drastically, the bottleneck of the B&C is Separation Problem. If we use parallelization for Separation Problem calculation $v_i(\hat{r})$ or $h_{\sigma_i^\alpha}$, this performance can be further improved.

VI. Conclusion

In this paper we present an exact integer programming method for solving the chance constrained formulation of the multi-area reserve dimensioning problem subject to network constraints. The formulation is aligned with the requirements of European legislation for reserve sizing with the possibility of reserve exchange between areas, as articulated in the System

[3]The capacity data could have been chosen to be random, but for the simplicity of presentation we chose fixed numbers for our simulation.

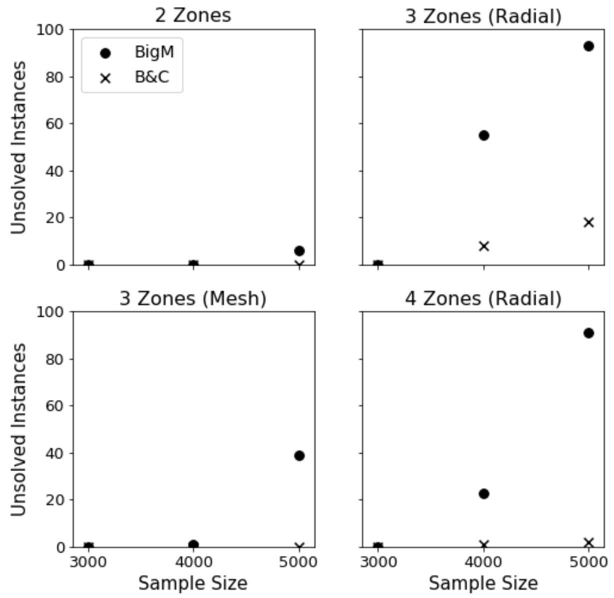

Fig. 3. Comparison between Big-M Formulation and Branch-and-Cut Algorithm for Different Networks

Operation Guideline [5]. Our model of the reserve dimensioning problem employs binary variables for indicating whether a certain imbalance scenario results in failure to serve load or not, with scenarios that are not covered resulting in an activation of the underlying binary variable. The aforementioned implication is formulated using the Modified Benders Cuts derived from the Standard Benders Feasibility Cuts. We further tighten the mixing set induced by the Modified Benders Cuts by utilizing the mixing inequalities. Finally, a Branch-and-Cut algorithm is introduced which can address this formulation.

Our method is tested against the big-M formulation of the problem [6]. Numerical experiments on various network configurations demonstrate that our method is able to both furnish an optimal solution which naturally outperforms the formulation from [6]. The technique developed in the paper can be used more broadly in chance constrained optimization formulations It is worth noting that chance-constrained formulations have found numerous applications recently in the power systems operation literature, e.g. in the context of optimal power flow [18], [19], and their interplay with pricing [20]. Our work thus identifies a specific application where these chance constraints are explicitly enshrined in European power systems legislation, and develops an exact method for tackling a class of such problems.

References

[1] European Commission, *Commission regulation (EU) 2017/2195 of 23 november 2017 establishing a guideline on electricity balancing*, Tech. Rep., 2017., https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=CELEX:32017R2195&from=EN

[2] M. Bucksteeg, L. Niesen, C. Weber, *Impacts of dynamic probabilistic reserve sizing techniques on reserve requirements and system costs*, IEEE Transactions on Sustainable Energy, 7(4), 2016.

[3] D. Jost, M. Speckmann, F. Sandau, R. Schwinn, *A new method for day-ahead sizing of control reserve in germany under a 100% renewable energy sources scenario*, Electric Power Systems Research, 119, 2015.

[4] J. De Haan, M. Gibescu, D. Klaar, W. Kling, *Sizing and allocation of frequency restoration reserves for LFC block cooperation*, Electric Power Systems Research, 128, 2015.

[5] European Commission, *Commission regulation (EU) 2017/1485 of 2 august 2017 establishing a guideline on electricity transmission system operation*, Tech. Rep., 2017.

[6] A. Papavasiliou et al., *Multi-area reserve dimensioning using chance-constrained optimization*, UCL Tech. Rep., 2021. https://ap-rg.eu/wp-content/uploads/2021/06/SvkReserveDimensioning.pdf

[7] T. Zheng, E. Litvinov, *Contingency-based zonal reserve modeling and pricing in a co-optimized energy and reserve market*, IEEE transactions on Power Systems, 23(2), 2008.

[8] Y. Chen, P. Gribik, J. Gardner, *Incorporating post zonal reserve deployment transmission constraints into energy and ancillary service co-optimization*, IEEE transactions on Power Systems, 29(2), 2013.

[9] F. Wang, Y. Chen, *Market implications of short-term reserve deliverability enhancement*, IEEE transactions on Power Systems, 36(2), 2021.

[10] B. Park, Z. Zhou, A. Botterud, P. Thimmapuram, *Probabilistic zonal reserve requirements for improved energy management and deliverability with wind power uncertainty*, IEEE transactions on Power Systems, 35(6), 2020.

[11] L. Roald, S. Misra, T. Krause, G. Andersson, *Corrective control to handle forecast uncertainty: A chance constrained optimal power flow*, IEEE transactions on Power Systems, 32(2), 2016.

[12] J. Luedtke, S. Ahmed, *A sample approximation approach for optimization with probabilistic constraints*, SIAM Journal on Optimization, 19(2), 2008.

[13] J. Luedtke, S. Ahmed, G.L. Nemhauser *An integer programming approach for linear programs with probabilistic constraints*, Math. Program., 122(2), 2010.

[14] J. Luedtke, *A branch-and-cut decomposition algorithm for solving chance-constrained mathematical programs with finite support*, Math. Program., 146(1), 2014.

[15] J.F. Benders, *Partitioning procedures for solving mixed-variables programming problems*,Numerische mathematik, 4(1), 1962.

[16] A. Atamtürk, G.L. Nemhauser, M.W. Savelsbergh, *The mixed vertex packing problem*, Math. Program., 89(1), 2000.

[17] O. Günlük, Y. Pochet, *Mixing mixed-integer inequalities*, Math. Program., 90(3), 2001.

[18] D. Bienstock, M. Chertkov, S. Harnett, *Chance-constrained optimal power flow: Risk-aware network control under uncertainty*, Siam Review, 56(3), 2014.

[19] B. Li, R. Jiang, J.L. Mathieu, *Distributionally robust chance-constrained optimal power flow assuming unimodal distributions with misspecified modes*, IEEE Transactions on Control of Network Systems, 6(3), 2019.

[20] X. Kuang et al., *Pricing chance constraints in electricity markets*, IEEE Transactions on Power Systems, 33(4), 2018.