Submitted to *INFORMS Journal on Computing*
manuscript (Please, provide the mansucript number!)

Algorithm for the N-2 Security Constrained Unit Commitment Problem with Transmission Switching

Kathryn M. Schumacher, Amy E.M. Cohn
Industrial and Operations Engineering Department, University of Michigan, Ann Arbor, Michigan 48109, kaschu@umich.edu, amycohn@umich.edu

Richard Li-Yang Chen
Quantitative Modeling and Analysis, Sandia National Laboratories, Livermore, California 94551, rlchen@sandia.gov

Most power grid systems are operated to be N-1 *secure*, meaning that the system can withstand the failure of any one component. There is increasing interest in more stringent security standards, where the power grid must be able to survive the (near) simultaneous failure of k components (i.e., N-k). However, this improved reliability criterion significantly increases the number of contingency scenarios that must be considered when solving the *unit commitment problem*. Additional computational complexity is introduced when taking into account *transmission switching*. This relatively inexpensive method of redirecting power flows in the grid has been proposed as a way of introducing flexibility to help survive failure events. We present an algorithm for solving the unit commitment problem that simultaneously addresses both the challenges of the N-k security requirement and the use of transmission switching during operation. We analyze the algorithmic performance and present computational results for the IEEE24 and RTS-96 test systems for $k = 1$ and 2. We also include a discussion of how this approach might be extended to solve problems with $k \geq 3$.

Key words: unit commitment, transmission switching, robust power system operation, N-k security
History:

1. Introduction

Recent blackout events have highlighted the need to have a power grid that is robust and reliable (Liscouski and Elliot (2004), Srivastava et al. (2012)). Currently, the North American Electricity Reliability Corporation (NERC) requires that the power grid be N-1 *secure*, meaning that load must be fully met in the event that any single component fails (NERC (2011)). The rationale for this policy is that the failure of a single component is considered to be a much more likely event than the near simultaneous failure of multiple components, and thus only single failures are considered when making operational planning decisions. However, given that component failures are not independent events, the probability of near-simultaneous failures may be higher than is currently estimated. Thus, the possibility of multiple failures is worth considering when making planning decisions.

Federal directives (PPD-21 (2013)) emphasize the importance of the security of the power grid as a critical infrastructure, and highlight the need to protect against major disruptions. Consequently there has been significant interest in considering reliability standards that are more stringent than N-1, such as N-2 or N-3, or more generally, N-k, where the grid must be able to survive any simultaneous failure of k or fewer components. A failure event of one or more components is commonly called a *contingency*. The set of all contingencies under consideration greatly increases when multiple failures are included, and thus the task of making planning and operational decisions becomes much more challenging.

The *unit commitment problem* is the day-ahead planning problem in which generators are scheduled to be on or off for each hour of the planning horizon. The generators in the power grid have operational limits including constraints on their minimum up and down times and ramp rate limits. To meet the forecasted demand for a balancing area, the on and off statuses of the generators must be planned ahead.

One way of including security requirements in the unit commitment problem is by specifying operating reserve (Read (2010)) which requires, for example, that the excess capacity of the committed generators be at least as much as the capacity of the largest generator. However, a requirement of this type does not take into account transmission constraints. Excess generating capacity in the event of a failure is useless if the transmission constraints do not allow power to be transported to where it is needed. An N-k secure generator schedule specifically considers how the transmission network constraints impact the available recourse actions in the event of a failure.

As is common in most optimization literature on grid planning, the power flow model in this paper is based on steady state analysis. While incorporating the effects of system dynamics is practically important, it is beyond the scope of this paper. To model the steady-state transmission network constraints in a power system, the Alternating Current Optimal Power Flow (ACOPF) equations are the ideal way to represent the physical laws. When several simplifying assumptions are made regarding stable operation, the ACOPF equations reduce to the linear DC power flow (DCPF) equations. The ACOPF equations are highly nonlinear, and thus optimization models typically use the DCPF equations as a linear approximation of the ACOPF equations. The DCPF equations are commonly used both in the academic literature and in industry (Hedman et al. (2011a)) and are used in our model.

Researchers have been exploring new "smart grid" technologies that improve the flexibility and efficiency of the operation of the power grid. In addressing grid congestion, there is a paradox associated with the existence of transmission capacity. On one hand, an arc in the network, i.e., a transmission line, allows power to be transmitted from one node to another, and thus can be useful in transmitting power from the generators to the consumers. On the other hand, given the laws of physics that govern how power flows throughout the network (i.e., Kirchhoff's circuit laws), the existence of an arc imposes a constraint on the system. In certain situations, removing a line can be advantageous in redirecting the flows in the network.

In a transmission model which uses DCPF constraints, removing a transmission line corresponds to removing the DCPF constraint for that line. Specifically, these situations arise in networks where there are cycles. Physical laws require that, when multiple paths exist between nodes, power must flow along all available paths. One path may be a bottleneck which constrains the flow on other paths, thus removing a transmission line may increase throughput. Cycles are often purposely designed into the power network to ensure redundancy, so there are often situations in which temporarily removing a line would be useful. An example of this phenomenon is presented in Hedman et al. (2011a).

Transmission switching is a practice where operators may open circuit breakers to switch transmission lines out of service to redirect the flow of power. This additional degree of control over the network topology has the potential to reduce the costs of dispatching generators and improve survivability of a contingency event. However, this additional set of switching decisions also introduces algorithmic challenges by dramatically increasing the dimension of the problem.

In this paper, we consider a unit commitment problem where N-k security is required, and transmission switching is allowed. A problem with this structure could naturally be decomposed into a two-stage program with mixed binary variables in both stages. However, such a formulation cannot be solved by standard decomposition methods, due to the existence of integer variables in the second stage and the very large number of scenarios (see Table 1 for specific numbers of contingencies considered in our computational experiments). We present novel models and methods to address these challenges. Specifically, we developed a column-and-cut algorithm based on Benders decomposition where binary switching variables are progressively generated and an integer bilevel separation oracle, with binary variables in both levels, is used to identify the worst-case contingency scenario when transmission switching is permitted in the recourse dispatch. A new combinatorial cutting plane algorithm is presented to solve the integer bilevel separation oracle.

The outline of this paper is as follows. We first review the literature on solving power system operational problems with N-k security and on using transmission switching to control the power flows in a network in Section 2. We then formally define the N-k unit commitment problem with transmission switching in Section 3. In Section 4, the natural two-stage decomposition and then an alternative decomposition are presented. In Section 5, the Contingency Oracle is derived, which is used to identify unsurvivable contingencies. In Section 6, the complete algorithm is defined, and implementation details are described which improve run time. Computational experiments are presented in Section 7 for the IEEE24 and RTS-96 test systems which demonstrate the value of switching, and the cost tradeoff of increasing reliability for $k = 1$ and $k = 2$. We discuss the challenges of using $k = 3$ for these test instances, and we suggest alternative approaches that may allow $k \geq 3$ to be solved in a reasonable amount of time. Finally, we present conclusions and ideas for future work in Section 8.

2. Literature Review

There has been significant work on network interdiction problems, and on various other ways of analyzing vulnerabilities in the power grid. Bienstock and Verma (2010), Salmeron et al. (2004) and Salmeron et al. (2009) present theoretical and computational results on solving the bilevel power system interdiction problem. Pinar et al. (2010) propose that the worst-case power grid interdiction problem can be accurately approximated as a (minimum cut) network inhibition problem, whose mixed-integer formulation can be solved for realistically-sized networks. Fan et al. (2011) present a critical node detection method for solving the power grid interdiction problem, and an economic basis for evaluating the damage caused by contingency events.

Several papers solve planning or operational problems with the N-k security standard. Street et al. (2011) present a robust optimization framework for solving the single bus unit commitment problem (i.e., where transmission constraints do not exist) when survivablility is required for any simultaneous failure of up to k generators. Wang et al. (2012) formulate the N-k unit commitment problem, where generators or transmission lines may fail, as a two-stage program and propose a cutting plane algorithm that solves for an exact solution. Chen et al. (2014) present a more general N-k-ε constrained transmission and generation capacity expansion planning problem that takes in consideration the amount of load shedding allowable under various contingency sizes.

The potential of transmission switching to significantly reduce the cost of dispatching generators is explored in Fisher et al. (2008). Following this work, Hedman et al. (2008) address the drawbacks and explore further the benefits of transmission switching as a corrective mechanism. Hedman et al. (2009) consider how transmission switching affects the costs of dispatching generators N-1 securely, and find that not only would it be possible for N-1 security to be maintained when transmission switching is used, but that the economic dispatch cost savings due to transmission switching are sometimes greater with N-1 security requirements than without. Li et al. (2012) use a constraint programming approach to solve for switching actions that enable the power system to recover from a contingency event without redistributing generators.

Analysis of how the worst-case power system interdiction models could be extended to include transmission switching is presented in Delgadillo et al. (2010) and in Zhao and Zeng (2011). Delgadillo et al. (2010) present a method for solving the worst-case electric grid interdiction problem with transmission switching allowed in the lower level problem by using Benders' decomposition within a restart framework. In Zhao and Zeng (2011) the authors present a tri-level reformulation of the bilevel interdiction problem with transmission switching in the lower level, which has an equivalent single level form that can be solved with a cutting plane algorithm.

Modifications of the unit commitment problem to incorporate transmission switching are presented in Hedman et al. (2010) and Khodaei et al. (2010). Khodaei et al. (2010) present a solution methodology that iterates between finding the best unit commitment decision and best transmission switching decisions, and apply this method for a handful of specific contingency events. Hedman et al. (2010) present a model for the N-1 secure unit commitment problem, where switching decisions are made for each time period, but the switching decisions are not changed in response to a contingency event. The authors present a heuristic method of solving this problem, which shows a cost savings of 3.7% in the unit commitment solution when transmission switching is employed, compared to when transmission switching is not used, for the RTS-96 test system. An economic analysis of the impact of transmission switching on the N-1 unit commitment problem is presented in O'Neill et al. (2010). All of these papers on transmission switching and the unit commitment problem also use the DCPF equations to model power flows.

The N-k secure unit commitment problem considered here could be classified as an adaptive robust problem (Ben-Tal et al. (2004)) with an uncertainty set defined as the set of all contingencies of size k or smaller. Herein we propose a formulation for the robust unit commitment problem which is similar in structure to a stochastic unit commitment problem with a finite number of contingency scenarios. However, our proposed algorithm does not require that all scenarios be explicitly enumerated, unlike other methods for solving two stage stochastic programs such as progressive hedging (Watson and Woodruff (2011)). When considering contingencies of size greater than 1, the number of contingencies is likely to be extremely large due to the combinatorial explosion, and thus it is necessary to develop a method which does not explicitly consider all scenarios. Our algorithm takes advantage of the specific structure of the problem and provides a tractable way of solving a problem that would otherwise be too large to solve with traditional methods. We also present several important implementation details which significantly impact the runtime of the overall algorithm, as demonstrated by our computational results.

3. Problem Definition

Our ultimate goal is to solve for a set of unit commitment decisions and generator dispatch decisions for normal operating conditions. These decisions should minimize the total cost of normal operation but must also be able to survive any contingency of size k or smaller, where a contingency is defined as the simultaneous failure of one or more components. In response to a contingency event, operators have the opportunity to redispatch generators which are already committed, and to switch transmission lines out as needed.

3.1. Assumptions

- **Components fail completely or not at all.** Partial failures are not considered.
- **Only transmission lines and/or generators may fail.** We model only generators, transmission lines (i.e., arcs), and buses (i.e., nodes) in our representation of the power grid. One or more generators may be on a single bus. During a contingency event, it is assumed that only transmission lines and/or generators may fail. The failed elements that define a contingency event are said to be contained within the contingency.
- **For a contingency of size ℓ, we define survival as meeting at least $(1-\varepsilon_\ell)$ fraction of the total demand.** The parameter ε_ℓ is defined for $\ell = 0, \ldots, k$ such that $0 \leq \varepsilon_0 \leq \cdots \leq \varepsilon_k \leq 1$. It is common to set $\varepsilon_0 = 0$ and $\varepsilon_1 = 0$.
- **Time is discretized at one hour intervals.** Multiple failures within the interval are considered simultaneous.
- **A contingency event is assumed to occur when the system is otherwise operating normally.** Contingency events in sequential time periods, or cascading failures are not considered. When a contingency occurs in a particular time period, the generators that were already committed in that time period can be redispatched, but generators that were not committed cannot be turned on.
- **To respond to a contingency in a given time period, the generator's output cannot be increased or decreased from the nominal output in that same time period by more than the ramp rate.**
- **For a single contingency, the post-contingency generator outputs are not linked across time periods.** When a contingency occurs, the primary concern is immediately finding a feasible power flow solution so that a blackout event will not occur. In subsequent time periods, the operator may take other actions to enable recovery including repairing broken components, bringing online generators that were previously uncommitted, etc. But for the purposes of the problem considered here, the only requirement is that a feasible power flow solution exists immediately following a contingency event. Subsequent time periods are not modeled, as it is assumed that once a stable solution has been found, the operator is able to recover using actions beyond just redispatching generators and switching transmission lines.

• **Transmission switching may be used in response to a contingency event but not during normal operation.** Previous studies (Fisher et al. (2008), Hedman et al. (2010)) have shown that transmission switching can be employed during normal operation to reduce the cost of dispatching generators by optimizing the network topology to allow the most efficient generators to meet demand. Other studies (Hedman et al. (2011b), Li et al. (2012)) have indicated that switching might also be used as recourse action, to help redirect flows in response to a contingency event to satisfy as much demand as possible. We focus here on the effect of transmission switching on system reliability, and thus we consider only the latter use case, in which transmission switching is used in response to a contingency event to improve the network's ability to survive the contingency. But it is trivial to extend our model to allow switching during normal operation.

• **No cost is assigned to post-contingency response decisions.** When a contingency occurs, the primary goal is to ensure feasibility, not to minimize the cost of operation. Thus, generator dispatch decisions under normal operation appear in the objective function, but post-contingency dispatch decisions do not. The decision to switch a line in or out of service is also not assigned any cost.

3.2. Explicit Formulation

Here we present the explicit formulation of the N-k unit commitment problem with transmission switching. For notational conciseness and clarity we present the explicit formulation of the problem using matrix notation. The full, detailed formulation is presented in the Online Supplement.

Let the $\mathcal{C}$ be the set of all contingencies of size k or smaller. Each contingency $\boldsymbol{c} \in \mathcal{C}$ is a binary vector of length equal to the number of generators and transmission lines, where $c_j = 1$ indicates that element j has failed. More formally, let $\mathcal{E}$ represent the set of transmission lines, and $\mathcal{G}$ represent the set of generators. The set of all contingencies $\mathcal{C} = \left\{\boldsymbol{c} \in \{0,1\}^{|\mathcal{E}|+|\mathcal{G}|} \mid \mathbf{1}^\top \boldsymbol{c} \leq k\right\}$ where $\mathbf{1}$ is an appropriately sized unit vector.

Let scalar c be the index of a given contingency, that is $c \in \{0, 1, \ldots, |\mathcal{C}|-1\}$. The set $\mathcal{C}$ contains the 0-contingency, $\boldsymbol{c} = \mathbf{0}$, where no components have failed, i.e., normal operation. Let $c = 0$ be the index of the no failure 0-contingency. With a slight abuse of notation, let $|c|$ be the size (i.e. number of failed elements) of contingency c. $\mathcal{T}$ is the set of 24 1-hr time periods, and b^t is the total load in time period t.

The vectors of variables used in this problem are:

- $\boldsymbol{x}^t$ binary unit commitment decisions including on/off and start-up/shut-down statuses of each generator in time period t
- $\boldsymbol{p}^{tc}$ generator dispatch decisions in time period t during contingency c
- $\boldsymbol{f}^{tc}$ operational decisions in time period t during contingency c including line flows, node phase angles, and load shedding at each node
- $\boldsymbol{w}^{tc}$ binary transmission switching variables in time period t during contingency c

Let $\boldsymbol{p}^0$ be a concatenation of $\boldsymbol{p}^{t0}$ for all $t\in\mathcal{T}$, and $\boldsymbol{x}$ be a concatenation of $\boldsymbol{x}^t$ decisions for all $t\in\mathcal{T}$. The complete formulation is as follows:

$$\min_{\boldsymbol{x},\boldsymbol{p},\boldsymbol{f},\boldsymbol{w}} \quad \boldsymbol{d}_x^\top \boldsymbol{x} + \boldsymbol{d}_p^\top \boldsymbol{p}^0 \tag{1a}$$

$$\text{s.t.} \quad U\boldsymbol{x} + Q\boldsymbol{p}^0 \leq \boldsymbol{q} \tag{1b}$$

$$A\boldsymbol{f}^{t0} + G\boldsymbol{p}^{t0} \leq \boldsymbol{r}^t \quad \forall t\in\mathcal{T} \tag{1c}$$

$$A\boldsymbol{f}^{tc} + B\text{diag}(\boldsymbol{1}-\boldsymbol{c})\boldsymbol{w}^{tc} + G\boldsymbol{p}^{tc} \leq H\boldsymbol{c} + \boldsymbol{r}^t \quad \forall \boldsymbol{c}\in\mathcal{C}\setminus\{\boldsymbol{0}\},\ t\in\mathcal{T} \tag{1d}$$

$$Y\boldsymbol{p}^{tc} - D\text{diag}(\boldsymbol{1}-\boldsymbol{c})\boldsymbol{x}^t \leq \boldsymbol{0} \quad \forall \boldsymbol{c}\in\mathcal{C},\ t\in\mathcal{T} \tag{1e}$$

$$\boldsymbol{h}^\top \boldsymbol{f}^{tc} \leq \varepsilon_{|c|} b^t \quad \forall \boldsymbol{c}\in\mathcal{C},\ t\in\mathcal{T} \tag{1f}$$

$$W(\boldsymbol{p}^{tc} - \boldsymbol{p}^{t0}) \leq V\boldsymbol{c} + \boldsymbol{s} \quad \forall \boldsymbol{c}\in\mathcal{C}\setminus\{\boldsymbol{0}\},\ t\in\mathcal{T} \tag{1g}$$

$$\boldsymbol{p}\geq\boldsymbol{0}, \quad \boldsymbol{x} \text{ binary} \tag{1h}$$

$$\boldsymbol{w}^{tc} \text{ binary } \forall \boldsymbol{c}\in\mathcal{C}\setminus\{\boldsymbol{0}\},\ t\in\mathcal{T} \tag{1i}$$

The objective function (1a) minimizes the total cost of operating the generators including start-up and shut-down costs and fuel costs under normal operating conditions (i.e., the 0-contingency). We assume a linear fuel cost function, but a piecewise linear approximation of a quadratic cost curve could also be used, as is common with generator fuel costs (Zhu (2009)). Constraint set (1b) defines the requirements for the unit commitment variables including start-up, shut-down, and minimum up and down time, as well as the ramping constraints on the power dispatch variables under normal operation, which restrict the increase or decrease in the power output in consecutive time periods to obey limitations imposed by the equipment. Constraints (1c) and (1e) define the operational constraints in the 0-contingency, and constraints (1d) and (1e) define the operational constraints in contingency c. Constraint set (1d) includes power flow balance, DCPF constraints on available transmission lines, power flow as a function of nodal phase angle differences, and capacities on line flows. When a line is contained in a contingency, the power flow on that line is forced to be 0, and the DCPF constraints for that line are not enforced. If a line is not contained in a contingency, but it is switched out, the power flow is similarly set to 0 and the DCPF constraints relaxed. Note that some constraints in this set depend on the particular time period (e.g., flow balance depends on time-dependent forecasted loads) and some constraints depend on the contingency (e.g., line capacities depend on whether the line is contained in a contingency). The primary constraints in set (1d) are given as follows. Unless otherwise stated, for the rest of the paper c,e,n, and t are defined over sets $\mathcal{C},\mathcal{E},\mathcal{N}$, and $\mathcal{T}$, respectively. The relevant variable and parameter definitions are as follows: p_g^{tc} is the power output at generator g; f_e^{tc} is the power flow on transmission element e; q_n^{tc} is the unsatisfied demand at bus n ; s_n^{tc} is the undelivered supply at bus n (q_n^{tc} and s_n^{tc} are not related to $\boldsymbol{q}$ and $\boldsymbol{s}$ used in compact formulation); b_n^t is the load at bus n; b_e is the electrical susceptance on line e; θ_n^{tc} is the phase angle of bus n; F_e is the power flow capacity of transmission line e. For all variables, the superscripts (tc) can be read as, in time t in the contingency c.

$$\sum_{g\in\mathcal{G}_n} p_g^{tc} + \sum_{e\in E_n^{\text{in}}} f_e^{tc} - \sum_{e\in E_n^{\text{out}}} f_e^{tc} + q_n^{tc} - s_n^{tc} = b_n^t \quad \forall n,t,c$$
$$-M(c_e + w_e^{tc}) \le b_e\left(\theta_j^{tc} - \theta_i^{tc}\right) - f_e^{tc} \le M(c_e + w_e^{tc}) \quad \forall e=(i,j),t,c$$
$$-F_e(1-c_e)(1-w_e^{tc}) \le f_e^{tc} \le F_e(1-c_e)(1-w_e^{tc}) \quad \forall e,t,c$$

Constraint set (1c) contains the same operational constraints as in (1d) except that for the 0-contingency the switching variables are not included, because in this model switching is not allowed during normal operation. Constraint set (1e) defines the bounds on the power output at each generator. The power output at a generator is restricted to be 0 if either the generator is not committed, or if the generator has failed in a particular contingency. Otherwise, the power output at a committed generator must be within the upper and lower output bounds. The full description of constraints (1e) are given as follows. $P_g^{\min}$ and $P_g^{\max}$ are the minimum and maximum generator outputs, respectively.

$$P_g^{\min}(1-c_g)x_g^t \le p_g^{tc} \le P_g^{\max}(1-c_g)x_g^t \quad \forall g,t,c$$

Constraint set (1f) requires that the total loss-of-load be less than a specified threshold, where the threshold is a function of the size of the contingency. The detailed formulation of (1f) is given as follows. $\hat{q}^{tc}$ is the amount by which the total unsatisfied demand exceeds the allowed amount of unsatisfied demand in time period t in contingency c.

$$\sum_{n\in\mathcal{N}} q_n^{tc} - \hat{q}^{tc} \le \varepsilon_{|c|} b^t \quad \forall g,t,c$$

Constraint (1g) specifies that the redispatched power outputs must obey ramping limits relative to the 0-contingency power dispatch decisions prior to the contingency event, where the vector s contains the ramping limits, and the term $V\boldsymbol{c}$ relaxes the limit on the post-contingency dispatch for a generator that is contained in a contingency. The detailed formulation of (1g) is given as follows. $R_g^{\min}$ is the ramp-down limit and $R_g^{\max}$ is the ramp-up limit.

$$-R_g^{\min} - P_g^{\max} c_g \le p_g^{tc} - p_g^{t0} \le R_g^{\max} \quad \forall g,t,c$$

4. Problem Decomposition

The full mixed-integer formulation (1) is typically very challenging to solve because the set of all contingencies $\mathcal{C}$ is very large even for moderately sized networks if $k>1$. The total number of contingencies is $\sum_{\ell=1}^{k}\binom{|\mathcal{E}|+|\mathcal{G}|}{\ell}$, which is on the order of $(|\mathcal{E}|+|\mathcal{G}|)^k$, assuming $(|\mathcal{E}|+|\mathcal{G}|)\gg k$. Therefore, we explore decomposition procedures that allow us to solve this mixed integer linear program.

4.1. Natural Two-Stage Decomposition

The natural decomposition of this problem follows from defining the set of scenarios to be all contingency-time period pairs, the first stage variables to be $\boldsymbol{x}$, $\boldsymbol{p}^0$, and $\boldsymbol{f}^0$, and the second stage variables to be $\boldsymbol{p}^{tc}$, $\boldsymbol{f}^{tc}$ and $\boldsymbol{w}^{tc}$. The two-stage problem is then:

$$\begin{aligned}
\min_{\boldsymbol{x},\boldsymbol{p}^0,\boldsymbol{f}^0} \quad & \boldsymbol{d}_x^\top \boldsymbol{x} + \boldsymbol{d}_p^\top \boldsymbol{p}^0 \\
\text{s.t.} \quad & \text{constraints (1b) - (1c)} \\
& Y\boldsymbol{p}^{t0} - D\boldsymbol{x}^t \leq \boldsymbol{0} \quad \forall t \in \mathcal{T} \\
& \boldsymbol{h}^\top \boldsymbol{f}^{t0} \leq \boldsymbol{0} \quad \forall t \in \mathcal{T} \\
& \boldsymbol{p}^0 \geq \boldsymbol{0}, \quad \boldsymbol{x} \text{ binary} \\
& \mathcal{F}^{tc}(\boldsymbol{x},\boldsymbol{p}^0) \text{ nonempty} \quad \forall t \in \mathcal{T},\ \boldsymbol{c} \in \mathcal{C} \setminus \{\boldsymbol{0}\}
\end{aligned} \tag{2}$$

The second stage feasibility problem for a particular contingency-time period pair is defined by the polyhedron $\mathcal{F}^{tc}(\boldsymbol{x},\boldsymbol{p}^0)$. If this polyhedron is nonempty for a first stage solution $(\boldsymbol{x},\boldsymbol{p}^0)$, for all contingencies and time periods, then all contingencies are survivable. This polyhedron will be referred to as the Unsurvivability Authenticator (UA).

$$\text{(UA)} \qquad \mathcal{F}^{tc}(\boldsymbol{x},\boldsymbol{p}^0) = \begin{cases} A\boldsymbol{f}^{tc} + B\text{diag}(\boldsymbol{1}-\boldsymbol{c})\boldsymbol{w}^{tc} + G\boldsymbol{p}^{tc} \leq H\boldsymbol{c} + \boldsymbol{r}^t \\ Y\boldsymbol{p}^{tc} \leq D\text{diag}(\boldsymbol{1}-\boldsymbol{c})\boldsymbol{x}^t \\ \boldsymbol{h}^\top \boldsymbol{f}^{tc} \leq \varepsilon_{|c|} b^t \\ W\boldsymbol{p}^{tc} \leq V\boldsymbol{c} + \boldsymbol{s} + W\boldsymbol{p}^{t0} \\ \boldsymbol{p}^{tc} \geq \boldsymbol{0} \\ \boldsymbol{w}^{tc} \text{ binary} \end{cases} \tag{3}$$

In this two-stage formulation, there exist binary variables $\boldsymbol{w}^{tc}$ in the second stage problem (3). Standard decomposition methods for solving stochastic programs cannot be used when there are integer variables in the second stage. There exist several methods for solving these types of problems using integer L-shaped method, disjunctive cuts or Fenchel cuts (Laporte and Louveaux (1993), Ntaimo (2013), Sen and Sherali (2006), Sherali and Fraticelli (2002)). These approaches involve generating cutting planes for the second stage to iteratively describe the convex hull, and thus tend to be computationally intensive and not scalable. In more recent work, Gade et al. (2014) present a decomposition algorithm with parametric Gomory cuts to interactively tighten approximations of the second-stage integer programs. Yuan and Sen (2009) addresses computational speed-ups that may be possible in cut generation associated with decomposition-based branch-and-cut methods and discusses bottleneck issues. Lubin et al. (2013) presents a parallel dual decomposition algorithm using a new formulation that permits parallel solution of the master program by using structure-exploiting interior-point solvers.

We instead take advantage of the specific structure of our problem, and suggest a novel reformulation in which the binary switching variables $\boldsymbol{w}^{tc}$ are added to the first stage. The authors in Khodaei et al. (2010) employ a related technique in their model of a transmission expansion planning problem with transmission switching. However, the algorithm proposed by Khodaei et al. (2010) is an explicit one in which feasibility under each contingency scenario is verified by solving a mixed-integer linear program. This is in contrast to our approach in which the full set of contingency scenarios are screened implicitly and switching variables are generated dynamically as violated contingencies are identified.

Once the switching variables have been moved into the first stage, the reformulated problem has a linear second stage problem, and thus a Benders' decomposition could be applied. However, this reformulation would result in a very large number of variables in the first stage problem (one switching variable for each transmission line for each contingency for each time period). We propose a procedure for solving this reformulation in which the switching variables are dynamically generated for the first stage problem on an as-needed basis. With this approach, the number of switching variables contained in the first stage problem is initially zero and grows slowly as cutting planes are added.

4.2. Reformulation and Cutting Plane Algorithm

In our reformulation, the master problem remains almost the same as (2) except that there exists a set of binary vectors of variables $\boldsymbol{w}^{tc}$ for all $t \in \mathcal{T}$ and $c \in \mathcal{C}$, and second stage feasibility is enforced instead with:

$$\mathcal{F}^{tc}(\boldsymbol{x}, \boldsymbol{p}^0, \boldsymbol{w}^{tc}) \text{ nonempty} \quad \forall t \in \mathcal{T},\ \boldsymbol{c} \in \mathcal{C} \setminus \{\boldsymbol{0}\} \tag{4}$$

We employ a Benders based approach where this second stage feasibility requirement (4) is initially relaxed, and then gradually enforced by adding Benders' feasibility cuts to the master problem.

In traditional Benders' decomposition, cutting planes would be generated for the master problem by solving a subproblem for each time period, for each contingency, in each iteration. Due to the large size of the set of contingencies when $k > 1$, this procedure is not viable because it would take an impractically long time to solve so many subproblems in each iteration.

To address this issue, we propose that a *Contingency Oracle* be used which identifies an unsurvivable contingency for the current unit commitment solution for a particular time period. The development of this oracle will be further explained in the next section 5, but let us for now assume that such an oracle exists. The Contingency Oracle provides a means for identifying violated constraints for the master problem even when there is a very large number of contingencies.

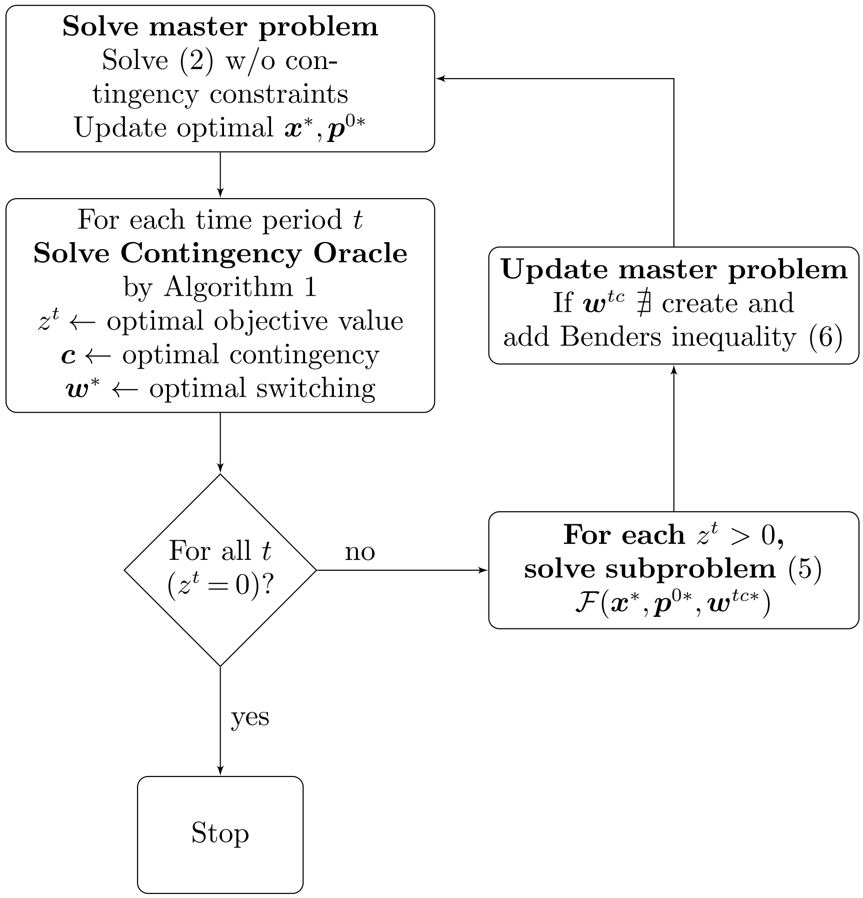

Figure 1 **Detailed Algorithm Overview**

The overall algorithm which incorporates the Contingency Oracle is illustrated in Figure 1. For every time period t, the unit commitment decisions $\boldsymbol{x}^t$ and 0-contingency economic dispatch decisions $\boldsymbol{p}^{t0}$ are passed to the Contingency Oracle. If, in all time periods, the Contingency Oracle identifies that all contingencies are survivable, the overall algorithm exits with the optimal set of unit commitment decisions $\boldsymbol{x}$ and 0-contingency economic dispatch decisions $\boldsymbol{p}^0$. If, for at least one time period, the Contingency Oracle identifies an unsurvivable contingency, this unsurvivable contingency is passed to a subproblem, along with the current master problem solution. The subproblem solution is then used to generate a feasibility cut to the master problem, and the procedure repeats.

Once an unsurvivable contingency has been identified for a particular time period, a feasibility cut is generated for the master problem by solving the dual of the following subproblem. The corresponding dual variables are denoted next to each constraint.

$$\mathcal{F}(\boldsymbol{x},\boldsymbol{p}^0,\boldsymbol{w}^{tc}) = \begin{cases} A\boldsymbol{f}^{ti} + G\boldsymbol{p}^{tc} \leq H\boldsymbol{c} + \boldsymbol{r}^t - B\text{diag}(\boldsymbol{1}-\boldsymbol{c})\boldsymbol{w}^{tc} & (\pi) \\ Y\boldsymbol{p}^{tc} \leq D\text{diag}(\boldsymbol{1}-\boldsymbol{c})\boldsymbol{x}^t & (\beta) \\ \boldsymbol{h}^\top \boldsymbol{f}^{tc} \leq \varepsilon_{|c|} b^t & (\gamma) \\ W\boldsymbol{p}^{tc} \leq V\boldsymbol{c} + \boldsymbol{s} + W\boldsymbol{p}^{t0} & (\rho) \\ \boldsymbol{p}^{tc} \geq \boldsymbol{0} \end{cases} \tag{5}$$

For the unsurvivable contingency c in time period t, the feasibility cut takes the following form, where $(\bar{\boldsymbol{\pi}}, \bar{\boldsymbol{\beta}}, \bar{\gamma}, \bar{\boldsymbol{\rho}})$ is the optimal dual subproblem solution.

$$\bar{\boldsymbol{\pi}}^\top(H\boldsymbol{c} + \boldsymbol{r}^t - B\text{diag}(\boldsymbol{1}-\boldsymbol{c})\boldsymbol{w}^{tc}) + \bar{\boldsymbol{\beta}}^\top (D\text{diag}(\boldsymbol{1}-\boldsymbol{c})\boldsymbol{x}^t) + \bar{\gamma}^\top \varepsilon_{|c|} b^t + \bar{\boldsymbol{\rho}}^\top (V\boldsymbol{c} + s + W\boldsymbol{p}^{t0}) \leq 0 \tag{6}$$

We recognize that initially, no feasibility cuts of the form (6) exist in the master problem, and all switching variables are unconstrained. Any first stage variables that are not contained in any constraints can effectively be ignored. As feasibility cuts of the form (6) are generated for the master problem, each of which contains a set of switching variables $\boldsymbol{w}^{tc}$, we suggest that the relevant vector of variables $\boldsymbol{w}^{tc}$ be added to the formulation. Thus, the number of switching variables effectively in the master problem grows gradually as cutting planes are generated for the master problem.

Using this procedure of dynamically generating switching variables for the master problem, we note that there is a choice to make when passing the master problem solution to the subproblem. Once an unsurvivable contingency c has been identified for time period t, two cases are possible:

1. At least one feasibility cut (6) for the given time period t and contingency c has already been added to the master problem. Thus, the vector of variables $\boldsymbol{w}^{tc}$ is contained in at least one constraint in the current master problem and therefore has been assigned a value in the solution to the master problem. The master problem solution $(\boldsymbol{x}^t, \boldsymbol{p}^{t0}, \boldsymbol{w}^{tc})$ should be passed to the subproblem (5).

2. No constraints from the set (6) for the given time period t and contingency c have yet been added to the master problem. The vector of variables $\boldsymbol{w}^{tc}$ is not yet contained in any master problem constraints, and thus any binary vector is a feasible solution for $\boldsymbol{w}^{tc}$. The master problem solution $(\boldsymbol{x}^t, \boldsymbol{p}^{t0})$ is passed to the subproblem (5), and any arbitrary binary vector can be set for $\boldsymbol{w}^{tc}$.

5. Contingency Oracle

The purpose of the Contingency Oracle is to identify, for a particular size k, time period t with unit commitment decisions $\boldsymbol{x}^t$, and 0-contingency economic dispatch decisions $\boldsymbol{p}^{t0}$, a contingency for which the minimum loss-of-load exceeds the allowable threshold, even when the network operator has the opportunity to redispatch generators and switch lines out of service in response to the contingency. If such a contingency does not exist, the oracle provides a certificate that all contingencies of size k or smaller are survivable. If such an unsurvivable contingency is found, it can be used to generate a valid feasibility cut for the master problem.

It is worth noting that the switching decisions determined in the master problem need not be passed to the oracle. The overall goal is to determine unit commitment and 0-contingency dispatch decisions such that there is guaranteed to exist some feasible operating solution (possibly including switching decisions) in the event of any contingency of size k or smaller. It is not necessary to know what these solutions are, but simply that they exist. This distinction means that the oracle will identify fewer unsurvivable contingencies than would be identified in a traditional Benders' approach, which in turn results in fewer feasibility cuts and fewer switching variables added to the master problem.

To solve this Contingency Oracle, we could pose a bilevel program for each time period t that identifies a maximally-damaging contingency given the current first stage decisions $\boldsymbol{x}^t$ and $\boldsymbol{p}^{t0}$. This bilevel program can be thought of as an adversary's problem, where the adversary seeks to maximize the minimum loss-of-load. The adversary decides which elements of the system to destroy, knowing that the system operator has the opportunity to redispatch generators and switch lines out of service in order to minimize the loss-of-load.

Specifically, the upper level problem (i.e., the adversary) determines which generators and/or transmission lines to destroy for a given time period. The lower level problem (i.e., the system operator) determines how best to dispatch power and switch lines in response to this contingency event so as to minimize loss-of-load. The optimal solution to the bilevel program is a contingency which maximizes the minimum loss-of-load for the given time period.

Such bilevel programs with mixed binary lower level decisions are diffiult to solve, however (DeNegre and Ralphs (2009), Scaparra and Church (2008)). On the other hand, if transmission switching is ignored, then the lower level problem becomes an LP and the bilevel program can be reformulated as a relatively small single MIP. We therefore have developed an iterative constraint generation algorithm that uses this observation to our advantage.

Specifically, we observe that if a contingency can be survived when switching is not allowed, then clearly it can also be survived when switching is allowed. Thus, we first use the no-switching bilevel program formulated as a single MIP to initially identify candidate unsurvivable contingencies; this MIP is referred to as the Candidate Contingency Identifier (CCI).

Once such a candidate has been identified we then verify whether the contingency is also unsurvivable when switching is allowed. If so, we have found an unsurvivable contingency, which can then be used to generate a valid feasibility cut for the master problem. If not, a constraint is generated for CCI to rule out the candidate contingency and a new candidate is generated. This constraint generation procedure continues until either an unsurvivable contingency is identified or CCI certifies that no unsurvivable contingencies exist for the given $\boldsymbol{x}^t, \boldsymbol{p}^{t0}$.

5.1. Bilevel Program Without Lower Level Switching Decisions

The bilevel program for identifying contingencies that maximize the minimum loss-of-load above the allowable threshold when switching is *not* permitted is formulated as follows.

$$\max_{\boldsymbol{c}} \quad L(\boldsymbol{x}^t, \boldsymbol{p}^{t0}, \boldsymbol{c}) \tag{7a}$$

$$\text{s.t.} \quad \mathbf{1}^\top \boldsymbol{c} \leq k \tag{7b}$$

$L(\boldsymbol{x}^t, \boldsymbol{p}^{t0}, \boldsymbol{c})$ is the optimal objective value of the no-switching lower level problem, which minimizes the loss-of-load above the allowable threshold, given the inputs $\boldsymbol{x}^t, \boldsymbol{p}^{t0}, \boldsymbol{c}$. In this no-switching lower level problem, the operator has the option of redispatching generators in response to the contingency event $\boldsymbol{c}$, but does *not* have the option of switching transmission lines out of service.

This no-switching lower level problem is as follows. Note that the indices for contingency and time period have been omitted from the variables $\boldsymbol{f}$ and $\boldsymbol{p}$ for the sake of simplicity, but it should be understood that the lower level problem (8) is specific to a particular contingency-time period pair. The Contingency Oracle is called for a particular time period, and in that time period the upper level passes a contingency to the lower level problem.

$$L(\boldsymbol{x}^t, \boldsymbol{p}^{t0}, \boldsymbol{c}) = \min_{\boldsymbol{f}, \boldsymbol{p}} \quad \boldsymbol{h}^\top \boldsymbol{f} - \varepsilon_{|c|} b^t \tag{8a}$$

$$\text{s.t.} \quad A\boldsymbol{f} + G\boldsymbol{p} \leq H\boldsymbol{c} + \boldsymbol{r}^t \qquad (\pi) \tag{8b}$$

$$Y\boldsymbol{p} \leq (\mathbf{1} - \boldsymbol{c})^\top D\boldsymbol{x}^t \qquad (\beta) \tag{8c}$$

$$W\boldsymbol{p} \leq V\boldsymbol{c} + \boldsymbol{s} + W\boldsymbol{p}^{t0} \qquad (\rho) \tag{8d}$$

$$\boldsymbol{p} \geq \mathbf{0} \tag{8e}$$

Here we assume *relatively complete recourse*, that is problem (8) has a feasible solution for any $(\boldsymbol{x}^t, \boldsymbol{p}^{t0}, \boldsymbol{c})$. This assumption holds, for example, if the lower bound on committed generator output is equal to 0 for all generators because shedding all load is always a feasible solution (associated with setting all generation and power flows to zero). If there does not exist a feasible solution for any $(\boldsymbol{x}^t, \boldsymbol{p}^{t0}, \boldsymbol{c})$, the formulation can be modified to ensure feasibility by defining slack variables for loss-of-load and excess generation, and appropriately modifying the objective such that the non-negative slack variables are minimized. These additional slack variables are incorporated in our implementation but left out of the formulation for clarity of exposition.

A bilevel program with a linear lower level problem is traditionally solved by incorporating the upper level variables and constraints into the dual of the lower level problem. The upper level problem and the dual of the lower level problem have aligned objectives, and thus the bilevel program can be solved as a single optimization problem. The single optimization problem for solving the no-switching bilevel program is as follows, where the binary contingency variable $\boldsymbol{c}$ is added to the dual of the no-switching lower level problem.

$$
\begin{aligned}
\max_{\boldsymbol{c},\boldsymbol{\pi},\boldsymbol{\beta},\boldsymbol{\rho}} \quad & (H\boldsymbol{c}+\boldsymbol{r}^t)^\top\boldsymbol{\pi} \ +(D\mathrm{diag}(\mathbf{1}-\boldsymbol{c})\boldsymbol{x}^t)^\top\boldsymbol{\beta}+(V\boldsymbol{c}+\boldsymbol{s}+W\boldsymbol{p}^{t0})^\top\boldsymbol{\rho}-\varepsilon_\ell b^t && \text{(9a)}\\
\text{s.t.} \quad & \mathbf{1}^\top\boldsymbol{c}=\ell && \text{(9b)}\\
& \boldsymbol{\pi}^\top A=\boldsymbol{h}^\top \qquad (f) && \text{(9c)}\\
& \boldsymbol{\pi}^\top G+\boldsymbol{\beta}^\top Y+\boldsymbol{\rho}^\top W\le\mathbf{0}^\top \qquad (p) && \text{(9d)}\\
& \boldsymbol{\pi},\boldsymbol{\beta},\boldsymbol{\rho}\le\mathbf{0}, \quad \boldsymbol{c} \text{ binary} && \text{(9e)}
\end{aligned}
$$

Note that in constraint (9b), the size of the contingency is fixed to be of size ℓ. If the size of the contingency were not fixed, and constraint (9b) were replaced with the requirement that the contingency be of size less than or equal to k ($\mathbf{1}^\top\boldsymbol{c}\le k$), the size of the contingency would not be known *a priori*, and it would be unclear what load-shedding threshold value ε should be used in the objective (9a). However, because k is typically a small value, such as 1, 2 or 3, we can use a procedure of fixing the size of the contingency ℓ to progressively larger values, up to the value k. For example, we first restrict the size of contingency to be 1, and if no unsurvivable contingency of size 1 is found, then we change this constraint to consider contingencies of size 2, and so on, until an unsurvivable contingency is found, or it has been verified that there do not exist any contingencies of size k or smaller that are unsurvivable.

In its current form the objective (9a) contains bilinear terms, as the binary variables in $\boldsymbol{c}$ are multiplied by the continuous dual variables $\boldsymbol{\pi}$, $\boldsymbol{\beta}$, and $\boldsymbol{\rho}$. However, the objective can be linearized by standard methods (see equations (10) in Chen et al. (2015)), which involve replacing these bilinear terms with auxiliary variables and adding appropriate constraints to enforce a relationship that the auxiliary variables take on the same value as the original bilinear terms. The formulation (9) in its linearized form will be referred to as the CCI.

5.2. Contingency Oracle Solution Routine

In order to identify a contingency that is unsurvivable given the unit commitment decisions $\boldsymbol{x}^t$ and 0-contingency economic dispatch decisions $\boldsymbol{p}^{t0}$, we propose an iterative constraint generation algorithm. This basic algorithm is illustrated in Figure 2.

For a particular contingency size ℓ, CCI is first solved to identify an initial unsurvivable contingency candidate for the current unit commitment decisions $\boldsymbol{x}^t$ and 0-contingency economic dispatch decisions $\boldsymbol{p}^{t0}$. The unsurvivability of this contingency is checked by solving the Unsurvivability Authenticator (UA), the feasibility problem defined in (3). If UA is feasible, a constraint is added to CCI to make the current contingency solution infeasible. The procedure repeats until a contingency is identified that is unsurvivable, even for the optimal switching configuration, or a certification that no unsurvivable contingency of size ℓ exists is returned. This certification is obtained if the optimal objective value of CCI is greater than 0. If this certification is returned for all $\ell = 1, \ldots, k$, all contingencies of size k or smaller are survivable for the current first stage solution for the given time period. If, for all time periods, the Contingency Oracle verifies that no unsurvivable contingencies exist, the overall algorithm terminates with the optimal set of unit commitment and 0-contingency economic dispatch decisions.

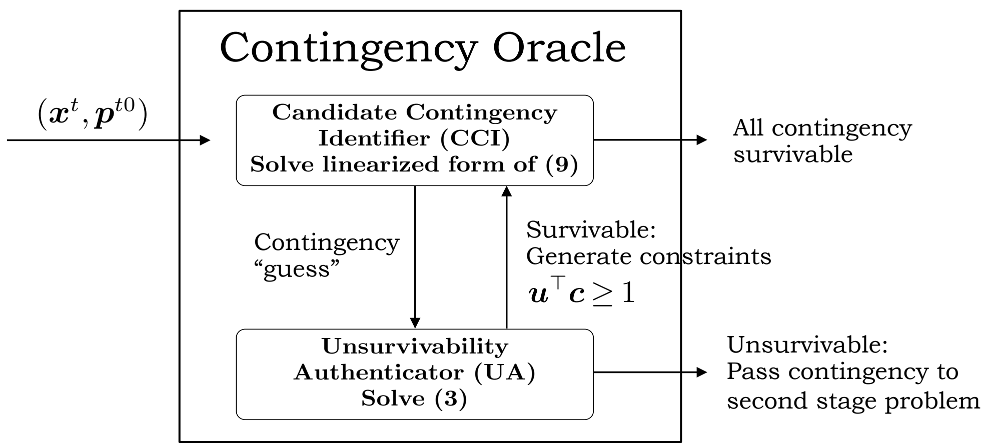

Figure 2 **Contingency Oracle Solution Approach**

When UA verifies that a particular contingency $\boldsymbol{c}$ is survivable, one valid inequality that could be added to CCI to make the current contingency solution $\boldsymbol{c}$ infeasible is to require that at least one element that is not included in the contingency $\boldsymbol{c}$ must be destructed. However, a tighter constraint is one that utilizes information about which elements were used in the feasible UA solution. Consider that for a survivable contingency $\boldsymbol{c}$ for time period t, the UA solution indicates a feasible set of edge flows and generator outputs. Let the vector of binary parameters $\boldsymbol{u}$ indicate which lines have nonzero flows and which generators have nonzero power outputs in the feasible solution to UA. In the next iteration, if none of the lines that had nonzero flows and none of the generators that had nonzero power outputs are destructed, then the same solution to UA will be feasible. Thus, in order to identify an unsurvivable contingency, at least one line with nonzero flow or one generator with nonzero output must be destroyed, which is expressed in the following constraint:

$$\boldsymbol{u}^\top \boldsymbol{c} \geq 1 \tag{10}$$

Thus we add constraint (10) to CCI based on the solution of UA to rule out survivable contingencies.

The Contingency Oracle takes in the unit commitment and 0-contingency dispatch decisions from the master problem $(\boldsymbol{x}, \boldsymbol{p}^0)$ and returns an unsurvivable contingency $\boldsymbol{c}$. The algorithm for solving the Contingency Oracle for a particular time period t, as illustrated in Figure 2, is summarized in Algorithm 1.

Algorithm 1: Contingency Oracle Algorithm

```
Initialization: ℓ = 1
while ℓ ≤ k do
    Solve CCI with contingency size ℓ let c and z_1 be optimal contingency and objective;
    if z_1 = 0 then
        ℓ ← ℓ + 1;
    else
        solve UA(x^t, p^{t0}, c) ande let z_2 be the objective value
        if z_2 = 0 then
            add u^T c ≥ 1 to CCI
        else
            return c;
        end
    end
end
```

6. Implementation Details

As described in section 4.2, the overall algorithm proceeds by iteratively solving the master problem, identifying unsurvivable contingencies, and solving the subproblem to generate feasibility cuts for the master problem. In this section, we discuss two algorithmic design decisions that have a significant impact on runtime, and we describe the implementation that we have empirically found to work well.

6.1. Identifying Unsurvivable Contingencies

Given the current master problem solution, the Contingency Oracle can identify an unsurvivable contingency for a particular time period, if one exists. However, the Contingency Oracle routine described in section 5.2 is an iterative procedure that involves generating constraints for CCI. The constraints generated for CCI are relatively weak, and so it is not uncommon for the routine to require many iterations, especially towards the end of the algorithm, when there do not exist many unsurvivable contingencies.

Rather than immediately calling the Contingency Oracle to identify an unsurvivable contingency, we suggest that a list of contingencies previously identified as unsurvivable first be checked. For many contingencies, multiple feasibility cuts must be added to the master problem before survivability is achieved. Contingencies that have been previously been identified as unsurvivable are thus good candidates for unsurvivability in future iterations. For a given master problem solution, we suggest checking all time periods. For each time period, we first check whether any contingencies in the list are unsurvivable. The Contingency Oracle is only called if all contingencies in the list are survivable for the time period *and* an unsurvivable contingency has not yet been identified for the current master problem solution. This routine reduces the frequency with which the Contingency Oracle is called while still ensuring that feasibility cuts are generated for the master problem in every iteration.

In solving RTS96 system for $k = 2$ the runtime with this improvement (along with ordered time periods described next) is 8.5 hrs. Given average CO runtime of 747 seconds (standard deviation 329 seconds) and a total of 278 iterations, the runtime if the CO was solved to identify a violated contingency at each iteration, instead of using the contingency list, would be approximately 278×747 seconds ≈ 600 hours. This represents an almost two order of magnitude increase in runtime. We suspect that this improvement will be even more significant for larger instances, as solving the CO is typically the bottleneck for larger systems and larger contingency size k.

6.2. Ordered Time Periods

We also suggest that the time periods be ordered by decreasing total load. The time periods are checked in their ranked order. It is more likely that an unsurvivable contingency will exist for peak load time periods, so by checking these time periods early in the iteration, unsurvivable contingencies are identified sooner. A new unsurvivable contingency is immediately added to the list of contingencies. When the longer list of contingencies is checked for subsequent time periods, there is greater likelihood of generating a feasibility cut, an increase in runtime of almost two orders of magnitude.

7. Computational Results

Our computational results were performed on a computer with 4GB RAM and a 2.3 GHz processor, using CPLEX v12.4. Computational tests were done with the IEEE24 and RTS-96 test systems, which are available online (Grigg et al. (1999)). In our test instances we modified the original network in the same way as described in Hedman et al. (2010), with the intent of slightly increasing congestion. See Hedman et al. (2010) for details.

The characteristics of these networks are summarized in Table 1. Note that "# Conting. $k = 1$" is the number of contingencies of size 1, which is the total number of transmission lines (i.e., arcs) and generators. Additionally, "# Conting. $k=2$", is the number of contingencies of size 2 (total number of transmission lines and generators choose 2) plus the number of contingencies of size 1, because setting $k=2$ means protecting all contingencies of size 2 or smaller.

System	# Nodes	# Arcs	# Generators	# Loads	# Conting. $k=1$	# Conting. $k=2$
IEEE24	24	37	32	17	69	2,415
RTS-96	73	117	96	51	213	22,791

Table 1 IEEE24 and RTS-96 System Characteristics

7.1. Run Times

To perform our computational experiments, we needed to pick a value for ε_k, the fraction of the allowable loss-of-load for contingencies of size k. In practice, $\varepsilon_0 = \varepsilon_1 = 0$. But there is not an established value for ε_k for $k > 1$. To obtain the most meaningful results, we sought the tightest values of ε_k, where the system is operating the closest to its limits. We refer to the smallest ε_k value that yields a feasible N-k unit commitment solution as the *critical* ε_k. For $k=1$ for the IEEE24 and RTS-96 systems, we initially set $\varepsilon_1 = 0$ and run our algorithm. If the N-k unit commitment problem was infeasible, we increased ε_k by increments of 0.01 until a feasible N-k secure unit commitment solution was obtained. For $k=2$, we held ε_1 at its critical value, and followed the same procedure to identify the critical ε_2. The critical ε_k values are for IEEE24 and RTS-96 for $k=1$ and $k=2$ are shown in Table 2.

System	Critical ε_1	Critical ε_2
IEEE24, w/ switching	0	0.09
IEEE24, w/o switching	0.01	0.1
RTS-96, w/ switching	0	0.04
RTS-96, w/o switching	0	0.04

Table 2 Critical ε_k Values

We tried computing the critical ε_k for $k=3$ for IEEE24, but no feasible N-k secure unit commitment solution could be obtained for any value of ε_3, and so we did not perform computational experiments with $k=3$ for these test instances. N-3 security may make sense for larger systems where 3 components is a small fraction of the total number of components, but it does not make sense for these test instances.

Using these critical ε_k values, we obtained the run time results for IEEE24 and RTS96 test systems for $k=1$ and $k=2$ shown in Table 3.

Test Case	# Iterations	Run Time	Percentage of Run Time Spent in Last Iteration
IEEE24, $k = 1$, $\varepsilon_1 = 0$	60	4 min	10%
IEEE24, $k = 2$, $\varepsilon_1 = 0$, $\varepsilon_2 = 0.09$	99	18 min	7%
RTS96, $k = 1$, $\varepsilon_1 = 0$	41	52 min	12%
RTS96, $k = 2$, $\varepsilon_1 = 0$, $\varepsilon_2 = 0.04$	92	8.5 hrs	23%

Table 3 Run Times

We implemented the algorithm serially. However, one of the advantages of the proposed algorithm is that it could be easily parallelized. We will discuss the runtimes of our serial implementation, and project how a parallel implementation could performed.

For the the IEEE24 network with $k = 1$, the algorithm converged in 60 iterations, and the run time was a little under 4 minutes, of which about 10% of that time was spent solving the last iteration, taking about 23 seconds. The last iteration is the slowest because the Contingency Oracle must be called serially for each time period, to verify that no unsurvivable contingencies exist. The earlier 59 iterations average about 3 seconds each. In a parallelized implementation, if there were 24 processors, each of the 24 Contingency Oracle instances in the last iteration could be solved simultaneously, such that solving the Contingency Oracles in the last iteration would take about 1.5 seconds instead of 23 seconds. Additionally, the algorithm would not require 60 iterations to converge. In the current implementation, the Contingency Oracle is only called if an unsurvivable contingency cannot be identified with the contingency list, and once an unsurvivable contingency is identified for this iteration, the Contingency Oracle is not called again. Thus, only a few constraints are added to the master problem in each iteration. However, in a parallelized implementation, the Contingency Oracle could be solved for multiple time periods in parallel in each iteration, potentially generating many more constraints for the master problem per iteration, and reducing the number of iterations necessary for convergence.

It is interesting to note that as k increased or as the network size increased, the number of iterations required did not dramatically increase. The main effect of increasing k and the network size is that the Contingency Oracle takes longer to solve. For example, the longest Contingency Oracle run time for RTS-96 with $k = 1$ is 35 seconds. The longest Contingency Oracle run time for RTS-96 with $k = 2$ is almost 10 minutes. The individual Contingency Oracle run times would not be reduced in a parallel implementation, but given that the bottleneck, the Contingency Oracle, could be parallelized, we would expect a substantial improvement in the overall runtime.

7.2. Critical ε Analysis

As previously mentioned, the minimum values of ε_k for which there exists a feasible N-k secure unit commitment solution for IEEE24 and RTS-96 for $k = 1$ and $k = 2$ are shown in Table 2. These critical ε_k values are a measure of how reliable the given power system is, and so it is interesting to analyze these ε_k values.

For IEEE24, when $k = 1$ and switching is used, it is possible to not shed any load for all contingencies, i.e., the critical $\varepsilon_1 = 0$ when switching is employed. When switching is not used in response to a contingency, it is necessary to shed 1% of the total load in the worst-case contingency in order for a feasible unit commitment solution to exist. The minimum loss-of-load that must be allowed for contingencies of size 2 in order for there to exist a feasible unit commitment solution is 9%, when switching is allowed, but increases to 10% if switching is not employed.

For the RTS-96 network, the critical ε_k values are the same with and without switching: for $k = 1$, the critical $\varepsilon_1 = 0$, and for $k = 2$, the critical $\varepsilon_2 = 0.04$. We believe the fact that switching reduces the critical ε_k values in the IEEE24 system and not in the RTS-96 system demonstrates that switching is most valuable in dense systems. The RTS-96 system is constructed of three zones, where there is significant interconnection among the buses within a zone, but only minimal connection between different zones, whereas the IEEE24 system is equivalent to one of the zones in the RTS-96 system. The IEEE24 network is more dense overall and thus is more constrained, and switching is more likely to increase survivability in the worst-case contingency.

7.3. Scaled-Load Analysis

We analyzed how the value of switching changed as the system load levels varied. We defined the load levels used in Section 7.1 as the 100% baseline, and then scaled the load at each node and time period for the IEEE24 system from 85% to 102%. We observed that as the load increased, the difference between the cost of the optimal solution when transmission switching is allowed, and the cost of the optimal solution when transmission switching is not allowed increases. Essentially, in a more congested system, switching is more valuable. We observed this effect in the IEEE24 system both when $k = 1$, shown in Figure 3, and when $k = 2$, shown in Figure 4.

In these computational experiments, the ε_k values were set equal to the critical ε_k values with switching. Both when $k = 1$ and when $k = 2$, the maximum load scaling at which there exists a feasible unit commitment solution without switching is 98%. For $k = 1$, there exists a feasible unit commitment solution with switching up to 102% scaled load, and for $k = 2$ there exists a feasible unit commitment solution with switching up to 100% scaled load. At 98% scaled load, when there exists a feasible unit commitment solution both with and without switching, the optimal objective value of the unit commitment solution and 0-contingency dispatch is about 18% cheaper when switching is used, for $k = 1$, and about 13% cheaper with switching for $k = 2$.

In Figure 5, the cost curves for IEEE24 from Figures 3 and 4 are overlaid on each other. In this plot, it can be seen that the cost curves for $k = 1$ without switching and $k = 2$ with switching

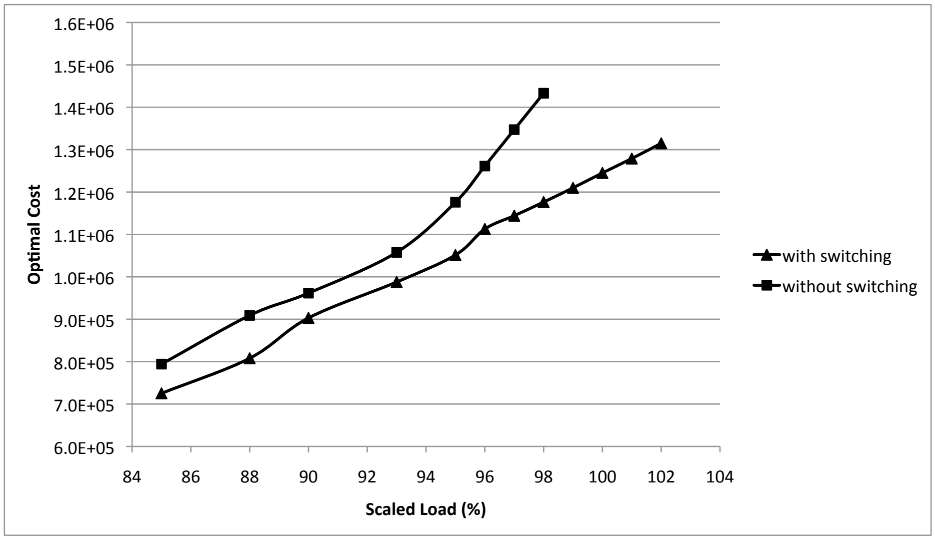

Figure 3 **IEEE24, k=1, Optimal Cost at Different Load Levels**

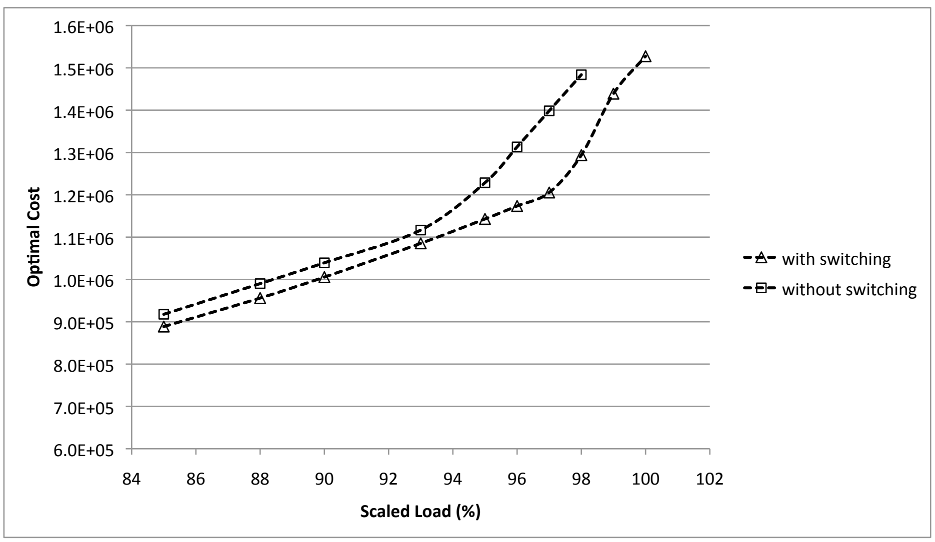

Figure 4 **IEEE24, k=2, Optimal Cost at Different Load Levels**

intersect at the scaled load level of 94%. This indicates that for demand levels above 94%, the decision to use transmission switching can allow the operator to achieve a higher level of reliability (N-2 security instead of N-1 security) at a lower cost. More generally a plot of this nature may help operators evaluate the cost of different levels of reliability, and determine the value of switching in their system.

We note that with switching, the optimal cost of the unit commitment and 0-contingency dispatch for IEEE24 with 100% scaled load and $\varepsilon_1 = 0$ is \$1.25 million. When k is increased to 2, with $\varepsilon_1 = 0$ and $\varepsilon_2 = 0.09$, the optimal cost increases to \$1.53 million, an increase of 22.7%. For the RTS96 network, the optimal cost of the unit commitment and 0-contingency dispatch when $k = 1$ with 100% scaled load and $\varepsilon_1 = 0$ is \$2.98 million. When k is increased to 2, with $\varepsilon_1 = 0$ and $\varepsilon_2 = 0.09$,

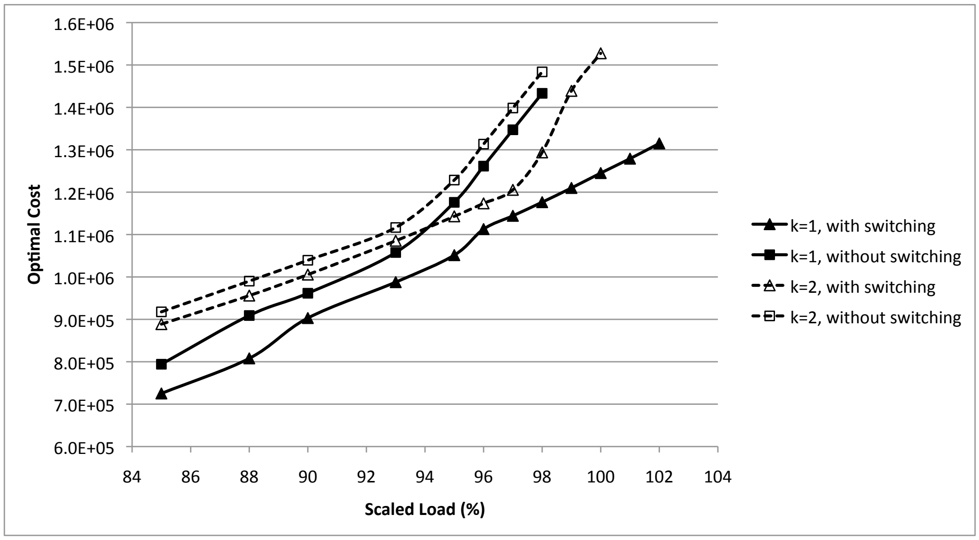

Figure 5 **IEEE24, k=1 & k=2, Optimal UC Cost at Different Load Levels**

the optimal cost increases to \$3.05 million, an increase of 2.5%. The difference between the optimal cost at $k = 1$ and $k = 2$ obviously is heavily dependent on the particular system costs. However, how much cost will increase when k is increased is difficult to predict without the use of tools like the algorithm presented here. The generators costs and characteristics used for the IEEE24 and RTS96 networks were quite similar, and yet the cost increase seen when increasing k from 1 to 2 was quite different.

7.4. Line Removal Analysis

Another interesting observation concerns which lines are frequently switched out in the optimal switching solution for various contingencies. Given the optimal unit commitment solution returned by the cutting plane algorithm, the optimal switching problem was solved for each contingency-time period pair, determining the generator dispatch and switching decisions that minimize the total loss-of-load given the available components. There appear to be multiple optimal switching configurations for many of the contingency-time period pairs. However, among these different optimal solutions, there is a pattern; a small subset of lines are switched out in the optimal solution a significant percentage of the time while most other lines are hardly ever switched out. Most optimal switching solutions for the IEEE24 network with $k = 1$ have 1-3 lines switched out in the optimal solution, and these lines generally belong to this subset of candidate switchable lines.

One might conclude that a line that is frequently switched out in the optimal solution should be permanently switched out. We tested this hypothesis by individually removing the six most frequently switched out lines, and computing the optimal unit commitment solution for each. In most cases, the objective value was nearly the same, within 1%. However, for two instances, the optimal unit commitment cost was 15% and 20% worse. In these cases, there was a line that was important for dispatching generators efficiently under normal operating conditions, but under contingency conditions it is useful to remove this line to minimize the loss-of-load. This result highlights the value of switching dynamically; the presence of a line can be valuable under one set of conditions, while the absence of that same line is valuable under a different set of conditions.

Given that such a small number of lines were switched out when the number of lines that could be switched out was not limited, we also tried explicitly limiting the number of lines that could be simultaneously switched out. For IEEE24 with $k=1$, limiting the number of lines switched out to 1 had a very small effect: the optimal objective value increased from \$1.3970M without a line limit to \$1.3973M, an increase of only 0.02%. This increase was just above the optimality gap threshold for the solver of 0.01%, so it was a observable difference, but a small one. The runtime did decrease, from 517 seconds without the limit to 441 seconds, a decrease of 15%. Limiting the number of lines that could be switched out reduced the feasible region and made the problem slightly easier to solve, without having much impact on the optimal objective value. It appears that most of the value in switching comes from the ability to switch 1 line out.

7.5. Future work: Extension to $k=3$ and higher

As previously mentioned, the IEEE24 system is too small for there to be a feasible unit commitment solution with $k=3$ for any loss-of-load threshold ε_3. For the RTS-96 test system, a feasible unit commitment solution is possible with $k=3$ and $\varepsilon_3=0.2$, but the instance takes days to run. Thus, enhancements must be made to our proposed algorithm in order to solve larger problem instances with $k\geq 3$ in a reasonable amount of time. Here we highlight some potential approaches that may improve scalability of the proposed algorithmic framework.

As is widely know, Benders decomposition suffers from slow convergence for certain classes of mixed integer linear programs, such as the unit commitment problem Wu and Shahidehpour (2010). The Benders cut strengthening technique proposed by Magnanti and Wong (1981) is based on selecting judiciously from alternate optima of subproblem (3) to obtain strong or pareto optimal cuts. The “strong” cut is obtained by solving another linear program, constrained to have the same objective value as the optimal solution of (3) but whose objective function is designed to maximized the coefficients of the resulting Benders inequality.

In Zeng and Zhao (2013) and Zhao and Zeng (2012), a column-and-constraint generation framework for two-stage robust optimization problems is proposed in which the worst-case extreme point and the associated recourse dispatch constraints are added to the master problem in each iteration. This column-and-cut approach is based on the premise that for many robust optimization problems only a small number of scenarios needed to be evaluate to ensure optimality and/or feasibility. This approach was shown to be extremely efficient for small and moderate size systems. For solving larger systems and larger contingency sizes (e.g. $k \geq 3$) a hybrid approach that combines the column-and-constraint generation framework of Zeng and Zhao (2013) with the Benders' cut augmentation procedure of Magnanti and Wong (1981) may greatly improve scalability.

Finally, in state unit commitment implementations, Benders decomposition is typically implemented as a cutting plane algorithm. Although easy to implement, cutting plane algorithms are computationally challenging because at each iteration of the algorithm an integer master problem such as (2) must be solved to optimality to select a candidate unit commitment vector $\boldsymbol{x}$. A branch-and-cut algorithm Mitchell (2002) avoids this drawback by incorporating violated inequalities directly within the branch-and-bound tree, thus avoiding the need to repeatily explore the branch-and-bound tree defined by the binary unit commitment variables $\boldsymbol{x}$. This has the potential to significantly decrease the overall runtime of the master problem solves, which is the bottleneck subroutine after the CO.

8. Conclusion

We have presented models and algorithms for solving the N-k secure unit commitment problem when switching is allowed as a recovery action. We first presented a natural two-stage decomposition of the problem with mixed-integer variables in both stages. We then offered a novel reformulation where the second stage integer switching decisions are moved to the first stage. The resulting two-stage formulation has a linear second stage and, using a procedure for dynamically generating first stage switching variables, can be solved via a cutting plane algorithm inspired by Benders' decomposition.

We formulate a Contingency Oracle, an optimization problem which identifies an unsurvivable contingency for the current unit commitment and 0-contingency dispatch decisions. In each iteration of the overall algorithm, all contingencies do not have to be explicitly considered because the Contingency Oracle is used to identify unsurvivable contingencies, for which feasibility cuts can be generated for the first stage. We demonstrate that the effective number of variables in the first stage is modest, as the switching decisions are added to the first stage only when a feasibility cut is added for the corresponding contingency-time period pair. Thus, this approach may be used when the number of contingencies is extremely large.

We have also presented several implementation details which have a significant impact on the total runtime. In particular, maintaining a list of contingencies that have been unsurvivable in any previous iteration can be used to quickly identify unsurvivable contingencies in the current iteration.

We have shown computational results for the IEEE24 and RTS-96 systems, when $k = 1$ and $k = 2$. Our results indicate that transmission switching is significantly valuable in reducing the cost of an

N-2 unit commitment solution. Additionally, our results indicate that the ability to dynamically switch lines in and out as needed has significant value, as opposed to statically removing a line. Further, these results suggest that this algorithmic framework could be implemented in parallel, and, with the described enhancements, may be used to solve problems of larger size, and larger values of k.

In addition to exploring parallelization and improved scalability, in our future work, we would like to extend this model to include uncertainty in the forecasted demand and/or available renewable generation. Additionally, it may be interesting to consider a different type of uncertainty set which defines the set of all contingencies that must be protected against to include more nuanced information about which components are more or less prone to failure.

Acknowledgments

This work was supported in part by an NSF Graduate Student Fellowship.

Sandia National Laboratories' Laboratory-Directed Research funded portions of this work. Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

References

Ben-Tal, A., A. Goryashko, E. Guslitzer, A. Nemirovski. 2004. Adjustable robust solutions of uncertain linear programs. *Math. Programming* **99** 351–376.

Bienstock, D., A. Verma. 2010. The $N-k$ problem in power grids: New models, formulations, and numerical experiments. *SIAM J. on Optim.* **20** 2352–2380.

Chen, R. L.-Y., A. Cohn, N. Fan, A. Pinar. 2014. Contingency-risk informed power system design. *IEEE Trans. on Power Systems* **29** 2087–2096.

Chen, R. L.-Y., N. Fan, A. Pinar, J.-P. Watson. 2015. Contingency-constrained unit commitment with post-contingency corrective recourse. *Annals of Operations Research* 1–27.

Delgadillo, A., J.M. Arroyo, N. Alguacil. 2010. Analysis of electric grid interdiction with line switching. *IEEE Trans. on Power Systems* **25** 633–641.

DeNegre, ST, T.K. Ralphs. 2009. A branch-and-cut algorithm for integer bilevel linear programs. *Operations Research and Cyber-Infrastructure*. Springer, 65–78.

Fan, N., H. Xu, F. Pan, P.M. Pardalos. 2011. Economic analysis of the $N-k$ power grid contingency selection and evaluation by graph algorithms and interdiction methods. *Energy Systems* 1–12.

Fisher, E.B., R.P. O'Neill, M.C. Ferris. 2008. Optimal transmission switching. *IEEE Trans. on Power Systems* **23** 1346–1355.

Gade, D., S. Küçükyavuz, S. Sen. 2014. Decomposition algorithms with parametric gomory cuts for two-stage stochastic integer programs. *Math. Program., Ser. A* **144** 36–64.

Grigg, Cliff, Peter Wong, Paul Albrecht, Ron Allan, Murty Bhavaraju, Roy Billinton, Quan Chen, Clement Fong, Suheil Haddad, Sastry Kuruganty, et al. 1999. The IEEE reliability test system-1996. *IEEE Trans. on Power Systems* **14** 1010–1020. http://www.ee.washington.edu/research/pstca/.

Hedman, K.W., M.C. Ferris, R.P. O'Neill, E.B. Fisher, S.S. Oren. 2010. Co-optimization of generation unit commitment and transmission switching with N-1 reliability. *IEEE Trans. on Power Systems* **25** 1052–1063.

Hedman, K.W., R.P. O'Neill, E.B. Fisher, S.S. Oren. 2008. Optimal transmission switching: sensitivity analysis and extensions. *IEEE Trans. on Power Systems* **23** 1469–1479.

Hedman, K.W., R.P. O'Neill, E.B. Fisher, S.S. Oren. 2009. Optimal transmission switching with contingency analysis. *IEEE Trans. on Power Systems* **24** 1577–1586.

Hedman, K.W., S.S. Oren, R.P. O'Neill. 2011a. Optimal transmission switching: economic efficiency and market implications. *J. of Regulatory Econom.* **40** 111–140.

Hedman, K.W., S.S. Oren, R.P. O'Neill. 2011b. A review of transmission switching and network topology optimization. *IEEE PES General Meeting*. IEEE, 1–7.

Khodaei, A., M. Shahidehpour, S. Kamalinia. 2010. Transmission switching in expansion planning. *IEEE Trans. on Power Systems* **25** 1722–1733.

Laporte, G., F. Louveaux. 1993. The integer *l*-shaped method for stochastic integer programs with complete recourse. *Oper. Res. Lett.* **13** 133–142.

Li, M., P.B. Luh, L.D. Michel, Q. Zhao, X. Luo. 2012. Corrective line switching with security constraints for the base and contingency cases. *IEEE Trans. on Power Systems* **27** 125–133.

Liscouski, B, W Elliot. 2004. Final report on the august 14, 2003 blackout in the united states and canada: Causes and recommendations. *A report to US Department of Energy* **40** 4.

Lubin, M., K. Martin, C. Petra, Sandikçi. 2013. On parallelizing dual decomposition in stochastic integer programming. *Oper. Res. Lett.* **41** 252–258.

Magnanti, T. L., R. T. Wong. 1981. Accelerating benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research* **29** 464–484.

Mitchell, J. E. 2002. *Handbook of applied optimization*, chap. Branch-and-cut algorithms for combinatorial optimization problems. 65–77.

NERC, North American Electric Reliability Corporation. 2011. Standard TPL-001-01, system performance under normal conditions. http://www.nerc.com/docs/standards/sar/Project_2006-02_TPL-001-1.pdf.

Ntaimo, L. 2013. Fenchel decomposition for stochastic mixed-integer programming. *J. of Global Optim.* **55** 141–163.

O'Neill, R.P., K.W. Hedman, E.A. Krall, A. Papavasiliou, S.S. Oren. 2010. Economic analysis of the N-1 reliable unit commitment and transmission switching problem using duality concepts. *Energy Systems* **1** 165–195.

Pinar, A., J. Meza, V. Donde, B. Lesieutre. 2010. Optimization strategies for the vulnerability analysis of the electric power grid. *SIAM J. on Optim.* **20** 1786–1810.

PPD-21, Presidential Policy Directive. 2013. Critical infrastructure security and resilience. `http://www.whitehouse.gov/the-press-office/2013/02/12/presidential-policy-directive-critical-infrastructure-security-and-resil`.

Read, E Grant. 2010. Co-optimization of energy and ancillary service markets. *Handbook of Power Systems I*. Springer, 307–327.

Salmeron, J., K. Wood, R. Baldick. 2004. Analysis of electric grid security under terrorist threat. *IEEE Trans. on Power Systems* **19** 905–912.

Salmeron, J., K. Wood, R. Baldick. 2009. Worst-case interdiction analysis of large-scale electric power grids. *IEEE Trans. on Power Systems* **24** 96–104.

Scaparra, M.P., R.L. Church. 2008. A bilevel mixed-integer program for critical infrastructure protection planning. *Comput. & Oper. Res.* **35** 1905–1923.

Sen, S., H.D. Sherali. 2006. Decomposition with branch-and-cut approaches for two-stage stochastic mixed-integer programming. *Math. Programming* **106** 203–223.

Sherali, H.D., B.M.P. Fraticelli. 2002. A modification of benders' decomposition algorithm for discrete subproblems: An approach for stochastic programs with integer recourse. *J. of Global Optim.* **22** 319–342.

Srivastava, S. C., A. Velayutham, A. S. Bakshi. 2012. Report of the enquiry committee on grid disturbance in northern region on 30th July 2012 and in northern, eastern & north-eastern region on 31st July, 2012, Ministry of Power, Government of India.

Street, A., F. Oliveira, J.M. Arroyo. 2011. Contingency-constrained unit commitment with $n-K$ security criterion: A robust optimization approach. *IEEE Trans. on Power Systems* **26** 1581–1590.

Wang, Q., J-P Watson, Y. Guan. 2012. Two-stage robust optimization for $N-k$ network and contingency-constrained unit commitment. Available at `http://www.optimization-online.org/DB_FILE/2012/01/3332.pdf`.

Watson, Jean-Paul, David L Woodruff. 2011. Progressive hedging innovations for a class of stochastic mixed-integer resource allocation problems. *Computational Management Sci.* **8** 355–370.

Wu, L., M. Shahidehpour. 2010. Accelerating the benders decomposition for network-constrained unit commitment problems. *Energy Systems* **1** 339–376.

Yuan, Y., S. Sen. 2009. Enhanced cut generation methods for decomposition-based branch and cut for two-stage stochastic mixed-integer programs. *INFORMS J. on Comput.* **21** 480–487.

Zeng, B., L. Zhao. 2013. Solving two-stage robust optimization problems using a column-and-constraint generation method. *Operations Research Letters* **41** 457–461.

Zhao, L., B. Zeng. 2011. An exact algorithm for power grid interdiction problem with line switching. available in *optimization-online*, University of South Florida.

Zhao, L., B. Zeng. 2012. Robust unit commitment problem with demand response and wind energy. *IEEE PES General Meeting*. IEEE, 1–8.

Zhu, J. 2009. *Optimization of power system operation*, vol. 49. Wiley-IEEE Press.