

# Corrective Line Switching With Security Constraints for the Base and Contingency Cases

Mingyang Li, *Student Member, IEEE*, Peter B. Luh, *Fellow, IEEE*, Laurent D. Michel, Qianchuan Zhao, *Senior Member, IEEE*, and Xiaochuan Luo, *Member, IEEE*

**Abstract**—Following a line outage, the fast corrective operations of transmission line switching might be used to regain  $N - 1$  security of the system without generation re-dispatch or load shedding. The problem to find feasible switching operations can be formulated as a constraint satisfaction problem (CSP). Feasibility checking, however, is difficult since changes in load flows caused by line switching operations are discontinuous, and many contingency cases need to be examined. In this paper, DC flow is considered for simplicity, and variables include binary line statuses and continuous phase angles. Security constraints for the base case,  $N - 1$ , and selected  $N - 2$  cases are formulated in a unified way by using a separate set of phase angles for each case. The problem is solved by using constraint programming (CP), and a tree search procedure is developed. Since it is time consuming to handle continuous variables in the tree search, only binary variables are branched on. Once reaching a leaf node where the topology is fixed, the constraints become linear DC flow feasibility conditions and are examined by solving a linear programming problem. Effectiveness of the method is demonstrated on IEEE 30-bus and 118-bus systems.

**Index Terms**—Constraint programming (CP), line outage, line overload, load flow feasibility,  $N - 1$  and  $N - 2$  contingency, transmission line switching.

## I. INTRODUCTION

ACCORDING to the  $N - 1$  operating criteria, the system should survive the failure of any one of its  $N$  components. Therefore, the system following a line outage should not have

Manuscript received August 11, 2010; revised November 29, 2010, March 05, 2011, May 02, 2011, and June 16, 2011; accepted July 29, 2011. Date of publication September 06, 2011; date of current version January 20, 2012. This work was supported in part by the NSFC under Grants 60574067, 60736027, 60721003, 61074034, and 61021063, by the NCET Program of China under Grant NCET-04-0094, by the National Key Lab for Power Systems in Tsinghua University, by the National 111 International Collaboration Project, and by the U.S. National Science Foundation grant ECS-0621936. Paper no. TPWRS-00653-2010.

M. Li and Q. Zhao are with the Center for Intelligent and Networked Systems (CFINS), Department of Automation, Tsinghua National Laboratory for Information Science and Technology (TNLIST), Tsinghua University, Beijing 100084, China (e-mail: li-my02@mails.tsinghua.edu.cn; zhaoqc@tsinghua.edu.cn).

P. B. Luh is with the Center for Intelligent and Networked Systems (CFINS), Department of Automation, Tsinghua National Laboratory for Information Science and Technology (TNLIST), Tsinghua University, Beijing 100084, China, and also with the Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT 06269-2157 USA (e-mail: Peter.Luh@uconn.edu).

L. D. Michel is with the Department of Computer Science and Engineering, University of Connecticut, Storrs, CT 06269-2155 USA (e-mail: ldm@engr.uconn.edu).

X. Luo is with ISO New England Inc., Holyoke, MA 01040 USA (e-mail: xluo@iso-ne.com).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TPWRS.2011.2164098

security violations in the new base case, but may be insecure in some of the contingency cases. In this situation, overloading of lines in these contingency cases might be relieved by transmission line switching operations without changing generation or load [1]–[5]. The idea is that by switching certain lines, part of the flows through the overloaded lines could be rerouted to other lines with spare transmission capacities. Compared with other corrective measures including generator re-dispatch and load shedding, the effects of line switching are fast in that load flows can be redistributed very quickly to relieve the overload. While the power electronics-based flexible AC transmission systems (FACTS) can also regulate load flows to perform corrective actions [6], line switching is economical because it requires no additional investment, only involves the relatively small costs in reducing lifetimes of circuit breakers. Such switching operations can be reversed afterwards if the line outage is temporary and is restored quickly, or a generator re-dispatch under the new system topology may be necessary at the next cycle of economic dispatch, which is typically run every 5 min or more [7]. Generator outages, however, cannot be corrected by line switching, because a generator re-dispatch is inevitable to address the power imbalance. Considering the increasing integration of intermittent renewable generation such as wind and solar energy, systems in the future will be faced with more uncertainties than today, and corrective switching operations will become more important in maintaining system security and reliability.

As required by NERC reliability standards [8], the system after a contingency should be corrected to be  $N - 1$  secure as soon as possible and within 30 min. Thus after line switching, load flow feasibility constraints need to be satisfied for both the base case and  $N - 1$  contingency cases. In certain critical areas, even  $N - 2$  security is required. Also, the switching actions should not lead to transient stability problems. Therefore, the problem to find a set of feasible line switching operations can be formulated as a constraint satisfaction problem (CSP), which is to find solutions of a set of decision variables satisfying the above security constraints. If no feasible line switching operation is available, other corrective measures such as generator rescheduling [9] should be considered, and if necessary, load shedding and controlled system islanding [10]–[14] may serve as the last resorts.

Since line switching operations lead to discrete changes of line statuses, the resulting changes in load flows are discontinuous and are difficult to analyze, especially when multiple lines are switched. Considering the large number of security constraints for  $N - 1$  and  $N - 2$  contingency cases, the task to find a set of feasible line switching operations is complicated.

As reviewed in Section II, although line switching operations have been studied by many, either as corrective or optimization measures, most of the existing methods consider only base case security constraints [1]–[5], [15]–[17]. A few results focusing on minimizing system operating costs considered  $N - 1$  security of switching actions, e.g., [18] and [19], with limitations such as examining only local effects of  $N - 1$  contingencies [18] and requiring long computation times [19].

In this paper, a new formulation and solution methodology are presented to search for corrective line switching operations that result in feasible load flows for both the base and contingency cases. For simplicity, steady state DC flow feasibility is considered as a necessary condition to screen out solutions to be further examined by using AC flow and transient stability analyses. We formulate the problem by using binary line status variables and continuous phase angle variables, as presented in Section III. Security constraints for the base case, the  $N - 1$ , and selected  $N - 2$  contingency cases are formulated in a unified way by using a separate set of phase angle variables for each case. This formulation can be used to analyze multiple line switching actions, and considers both switching in and out of lines. Although the formulation is nonlinear with multiplications of binary and continuous variables, a good feature is that once all binary variables are fixed, the constraints become linear DC flow feasibility conditions for a fixed topology.

Section IV presents our solution methodology based on constraint programming (CP) [20]. The key idea of CP is constraint propagation, which reduces domains of variables through the inferences of constraints. For problems with discrete variables, the basic procedure of CP is to propagate the constraints to examine their feasibility, and if necessary, the variables are sequentially branched on to form a tree search. It is, however, difficult to use the above procedure to solve continuous variables. This is because it is very time consuming to branch on continuous variables with infinite number of solutions, and the propagation techniques for continuous variables, such as interval arithmetic [20], have not been included in existing software packages. In our method, only binary variables are branched on, and once reaching a leaf node where all binary variables are fixed, the constraints become linear with only continuous variables owing to the feature of the formulation mentioned above. The feasibility of these linear constraints is then examined by solving a linear programming (LP) problem to overcome the difficulty in handling continuous variables. The method is able to obtain all feasible solutions of the problem, and after obtaining the first few solutions, AC flow analyses and transient simulations [21] can be run in parallel to further examine the solutions. For a quick corrective action, the method can terminate after finding a set of solutions that has passed the check of AC flow feasibility and transient stability.

Three examples are presented in Section V. A three-bus system illustrates the method in detail, and compares it with the alternative procedure of branching on both binary and continuous variables. The IEEE 30-bus example shows that the performance of the method can be improved by controlling the number of lines to be out, and by using heuristics in the sequence to branch on binary variables. Effectiveness of the method is also demonstrated on the IEEE 118-bus system.

## II. LITERATURE REVIEW

Transmission line switching has been studied for either corrective or optimization purposes. Several methods have been developed to search for corrective switching actions assuming fixed generation and load. The first type of methods discussed here is to model switching out a line by keeping the line in and adding a pair of power injections at its end buses, such that other power injections do not flow through the line [22]. Based on this model, a fast computation method of linear line outage distribution factors for DC flows [22] was developed in [1] to select corrective line and bus-bar switching actions, and a linear mixed integer programming (MIP) method was presented in [2] to minimize the cost of switching actions to relieve DC flow overload. An extension of the model, a power compensation model, was presented in [3], where a binary integer programming method was used to select switching actions to alleviate both line overload and voltage violations. The second type of methods is to compute network matrix modifications caused by switching actions. The changes of bus impedance matrix ( $Z$  matrix) were computed in [4] to select corrective line switching actions. Modifications of the bus admittance matrix were analyzed using a sparse inverse technique in [5], where switching actions of lines, bus-bars, and shunt elements were examined by fast decoupled load flow with a limited iteration count, and transient stability of the switching actions was analyzed using simulations. For all the above and many other related results, security constraints after line switching operations have been considered only for the base case, but not for the  $N - 1$  and  $N - 2$  contingency cases.

Line switching operations have also been used to reduce system operating costs in the context of optimal power flow (OPF). A network topology optimization scheme based on linear programming techniques was developed in [15], using a linear model to represent topology changes by injected currents. The method in [15] was further developed in [16] for loss minimization, and in [18] to include  $N - 1$  security constraints, assuming that a contingency affects only its surrounding area. Line switching and generator dispatch were combined to minimize generation costs in [17] using a linear MIP method, where binary line status variables were multiplied by a big positive number to linearize the DC flow feasibility constraints. The testing results in [17] on the IEEE 118-bus system indicated the high computational requirements for solving MIP. The formulation in [17] was extended in [19] to include  $N - 1$  security constraints by using a separate set of phase angles for each contingency case. Further extensions of [19] was presented in [23] to co-optimize unit commitment and transmission switching with  $N - 1$  reliability, and the problem was decomposed into unit commitment and switching sub-problems to improve the computational performance. Optimal system reconfiguration was studied in [24] considering transient stability constraints on short circuit currents and energy function values, and a decomposition method was used to separately solve the discrete topology variables and continuous state variables.

Applications of CP have not been reported for transmission line switching problems, but for the similar problem of reconfiguring radial distribution networks. The reconfiguration and

maintenance scheduling were studied together in [25], where the discrete variables of job starting times and switch statuses were solved by using constraint propagation, and the continuous variables of line currents were examined by using load flow analysis. The reconfiguration for loss reduction was studied in [26], where interval arithmetic [20] was implemented by the authors to solve continuous variables. Interval arithmetic is a technique to propagate linear or nonlinear constraints of continuous variables by making addition, subtraction, and multiplication operations on interval domains. Performance of the method in [26] was good for networks with up to 50 nodes, but the efficiency for larger networks still needed to be improved as acknowledged by the authors of [26].

### III. CONSTRAINT SATISFACTION PROBLEM FORMULATION

Consider a system following a certain line outage, with  $N$  buses,  $G$  generators, and  $L$  lines. Each bus, each line and each generator is assigned an integer as its index. For simplicity, only steady state DC flow is considered, and assume that the system has a single slack bus with index *slack*. The set of lines is denoted by  $E$ , from the word "edge" in graph theory. Each line is assigned a reference direction, and is also denoted by an ordered pair of the starting and ending buses  $(i, j)$ . The "in" and "out" statuses of each line is represented by, respectively, 1 and 0. The notations are as follows: the total load at bus  $n$  ( $1 \leq n \leq N$ ) is  $d_n$  MW; generator  $g$  ( $1 \leq g \leq G$ ) is connected to bus  $I(g)$ , and the output is  $p_g$  MW; the reactance of line  $(i, j)$  is  $x_{i,j}$ ; the current status of line  $(i, j)$ ,  $s_{i,j}$ , is given, and the status after switching is a binary variable  $z_{i,j}$ .

For the above system, the "base case" refers to the case immediately after the line outage, and a set of  $N-1$  and  $N-2$  contingency cases are considered. The set of all cases is denoted by  $C$ , which contains an integer index  $c$  for each case:  $c = 0$  represents the base case, and  $c > 0$  represents a contingency case. The  $N-1$  contingency cases include outages of any line or generator, while according to NERC reliability standards [8], exclude the outage of any radial line, which is the only line connected to a bus. This is because the bus will become isolated from the rest of the system in case the radial line is out. After a generator outage, the lost power needs to be temporarily provided by other generators, and a generator re-dispatch may then be needed if the outage cannot be restored quickly. For simplicity, all lost power is assumed to be picked up by the system slack bus by following [27] and many commercial software packages, e.g., PSS/E and PowerWorld. The  $N-2$  cases include simultaneous outages of two lines, or one line and one generator. Such cases are usually considered only for certain critical areas with relatively dense connections, characterized by a given line set  $E_2 \subset E$  and a given generators set  $G_2 \subset \{1, \dots, G\}$ . In each case  $c \in C$ , the sets of line(s) and generator(s) being out are denoted by, respectively,  $O_l(c)$  and  $O_g(c)$ . The phase angles and line ratings in each case are denoted by using a superscript  $c$ , and the emergency ratings in contingency cases are usually higher than the long-term ratings in the base case.

Based on the above descriptions, the line switching problem can be stated as a constraint satisfaction problem as follows. After a line outage, assuming that all generation  $p_g$  ( $1 \leq g \leq G$ ) and load  $d_n$  ( $1 \leq n \leq N$ ) are given and fixed, the task is to

find a set of line statuses  $z_{i,j}$  ( $(i, j) \in E$ ) and phase angles  $\theta_n^c$  ( $1 \leq n \leq N, c \in C$ ), such that the resulting load flows do not exceed the line ratings in all cases  $c$ .

The basis for mathematically formulating the above problem is the DC flow [22]. Suppose the phase angles at buses  $i$  and  $j$  are, respectively,  $\theta_i$  and  $\theta_j$ , then the flow of line  $(i, j)$ ,  $f_{i,j}$ , is

$$f_{i,j} = \frac{z_{i,j}}{x_{i,j}} (\theta_i - \theta_j). \quad (1)$$

It is clear from (1) that  $f_{i,j}$  is zero if line  $(i, j)$  is out. Similar to [19], the load flow feasibility constraints are established among the line statuses and the phase angles for both the base case and contingency cases. The lines and generators that are out in contingency cases are excluded using the sets  $O_l(c)$  and  $O_g(c)$  ( $c \in C$ ), so that the constraints for all cases are formulated in a unified way, as presented below.

First, power should be balanced at each bus, i.e., the net power injection at a bus equals to the algebraic sum of the flows going in and out of the bus. For a bus except the slack bus, the generation excludes the outputs of the generators being out in a contingency case, i.e.,

$$\sum_{\substack{I(g)=i \\ \text{and } g \notin O_g(c)}} p_g - d_i = \sum_{\substack{(i,j) \in E \\ (i,j) \notin O_l(c)}} \frac{z_{i,j}}{x_{i,j}} (\theta_i^c - \theta_j^c) + \sum_{\substack{(j,i) \in E \\ (j,i) \notin O_l(c)}} \frac{z_{j,i}}{x_{j,i}} (\theta_j^c - \theta_i^c), \\ i = 1 \sim N, \quad i \neq \text{slack}, \quad c \in C. \quad (2a)$$

The constraint for the slack bus is formulated differently, in that the generation includes the lost power of any generator being out in a contingency case, i.e.,

$$\sum_{\substack{I(g)=i \\ \text{or } g \in O_g(c)}} p_g - d_i = \sum_{\substack{(i,j) \in E \\ (i,j) \notin O_l(c)}} \frac{z_{i,j}}{x_{i,j}} (\theta_i^c - \theta_j^c) + \sum_{\substack{(j,i) \in E \\ (j,i) \notin O_l(c)}} \frac{z_{j,i}}{x_{j,i}} (\theta_j^c - \theta_i^c), \\ i = \text{slack}, \quad c \neq C. \quad (2b)$$

Second, the load flows through all lines, except those being out in contingency cases, should not exceed the line ratings:

$$-F_{i,j}^c \leq \frac{z_{i,j}}{x_{i,j}} (\theta_i^c - \theta_j^c) \leq F_{i,j}^c, \quad (i, j) \in E - O_l(c), \quad c \in C. \quad (3)$$

Third, following the DC flow assumptions [22], the phase angle difference between the starting and ending buses for any line being "in" should be within a given threshold  $\varepsilon$ :

$$-\varepsilon \leq z_{i,j} (\theta_i^c - \theta_j^c) \leq \varepsilon, \quad (i, j) \in E - O_l(c), \quad c \in C. \quad (4)$$

Also, to facilitate the solution methodology to be presented in Section IV, the phase angle at each bus is assumed to be within its lower and upper bounds, while no reference bus with a zero phase angle is specified:

$$\theta_{n,\min} \leq \theta_n^c \leq \theta_{n,\max}, \quad n = 1 \sim N, \quad c \in C. \quad (5)$$

Finally, to control the computational requirements of our method, at most  $M$  ( $1 \leq M \leq L$ ) lines are allowed to be out, i.e.,

$$\sum_{(i,j) \in E} (1 - z_{i,j}) \leq M. \quad (6)$$

Since it may not be necessary to consider switching actions for all lines, a line cannot be switched if it does not belong to a given switching candidate set  $E_C \subset E$ :

$$z_{i,j} = s_{i,j}, \quad (i,j) \in E - E_C. \quad (7)$$

Putting (2a), (2b), and (3)–(7) together, the formulation of a constraint satisfaction problem has been obtained.

Because of the multiplications of binary and continuous variables in the basic (1), constraints (2a), (2b), (3), and (4) in the above formulation are nonlinear. However, if all binary line statuses are fixed, the constraints are degenerated to the DC flow feasibility conditions for a given topology, and are linear in terms of the continuous phase angles. This feature enables an efficient solution procedure for the problem, as presented next.

#### IV. SOLUTION METHODOLOGY

In this section, the problem formulated in Section III is solved. The basic procedure of CP is reviewed in Section IV-A, and the difficulty in solving continuous variables is discussed. Section IV-B then presents our solution methodology based on CP, and the above difficulty is addressed by exploring the feature of the formulation mentioned at the end of Section III.

##### A. Basic Procedure of Constraint Programming

Constraint programming (CP) [20] is a systematic approach to solve constraint satisfaction problems. The major technique of CP is constraint propagation, which is a form of inference among constraints to reduce domains of variables. The inferences are based on the necessary conditions characterized by network consistency properties, such as arc consistency, which eliminates the values from the domain of a variable if they do not constitute feasible solutions with any values in the domains of other variables. Such inferences make CP especially efficient for problems with a large number of constraints, while complexity of many other techniques, such as those for solving MIP, usually grows with the increase of constraints. After constraint propagation, if any domain becomes empty, the problem is proved to be infeasible; otherwise, a search algorithm is used to find feasible solutions within the reduced domains.

For problems with discrete variables, the basic search algorithm is the backtracking search, which is, most often, a depth-first traversal of a tree. Nodes of the tree represent partial assignments of variables, and branches represent alternative choices to be examined. At the root node, one variable is selected and its domain is divided into several subsets. A group of new nodes are generated, each with the domain of the selected variable restricted to one of the subsets, while domains of other variable do not change. The search then goes to one of the new nodes and performs constraint propagation. If the new node is proved infeasible, it is pruned and the search backtracks to explore other nodes; otherwise, the new node is further branched on. If the search reaches a leaf node where the domain of each variable has a single value, a feasible solution is obtained. This search procedure is guaranteed to find a solution if one exists, and can prove the infeasibility of a problem if no solution exists.

The above procedure is relatively efficient for discrete variables with finite domains. However, solving continuous variables with interval domains is more challenging, because their

domains contain an infinite number of values and are time consuming to be branched on. In addition, although techniques are available to efficiently propagate constraints with continuous variables, e.g., interval arithmetic [20], such techniques have not been implemented in existing software packages that we know of. Thus the domains of continuous variables cannot be efficiently reduced to prune the search tree, and the search usually needs to go very deep and is time consuming. Since the problem formulated in Section III has both binary and continuous variables, it is difficult to solve the problem by using the basic CP procedure.

##### B. Our Solution Methodology

As mentioned in Section III, a good feature of the problem formulation is that the constraints become linear of continuous variables if all binary variables are fixed. Therefore in our method, only binary variables are branched on in a tree search. Once reaching a leaf node, feasibility of the linear constraints is examined by using LP methods to overcome the difficulty in solving continuous variables. The method is described below.

At the root node, all line status variables have possible values of 0 or 1, and all phase angle variables are within their lower and upper bounds as given by (5). A search tree for the binary variables is formed following the procedure reviewed in Section IV-A. In the search, continuous variables are not branched on, and their given domains are used in the constraint feasibility check at each node. If a leaf node is reached where all binary variables are fixed, the constraints are linear in terms of continuous variables, and the feasibility is checked by LP techniques. The basic method is to execute the first phase of many two-phase LP methods [28] to examine the feasibility of the LP problem and to obtain a set of feasible solutions if one exists. If the LP problem is infeasible, the leaf node is pruned and the search backtracks; otherwise, solutions of the LP problem and the values of binary variables at the corresponding leaf node constitute a set of feasible solutions of the problem. After obtaining the first few solutions, the search continues for more solutions, while AC flow analyses and transient simulations can run in parallel to further examine the solutions. If multiple feasible solutions are available, they can be ranked by using an optimization objective such as loss minimization.

Performance of the above method depends on two key factors. One is the number of line status variables to be branched on, and this is controlled by specifying the switching candidate set  $E_C$ . Considering  $N - 1$  security, a line cannot be switched out if it does not satisfy the following *topological necessary condition*: both of its end buses are either de-energized (with no net power injection) or connected to at least three lines. This is because if an energized bus is connected to only two lines and one of them is switched out, the bus will be isolated from the rest of the system if the other line is out in a contingency case. Then the system-wide power balance can no longer be maintained since generation is assumed to be fixed. Although the above condition is embedded in the  $N - 1$  constraints and can be automatically handled in constraint propagation, the search will become more efficient if the lines violating this condition are manually removed from the switching candidate set. To further control the number of candidates, a line is preferred if its status change

leads to large changes in the flows of overloaded lines. This can be estimated by using line outage distribution factors [22], or LODFs, which indicate how the outage of a line changes the flows of other lines. A heuristic way is thus to rank all lines by their LODFs with respect to the overloaded lines, and use the lines at the top of the list as switching candidates.

The second key factor of search performance is the sequence to branch on those selected line status variables. To obtain the first feasible solution as soon as possible, one heuristic way is to give priority to the lines with large LODFs with respect to the overloaded lines, as mentioned above. Another way is to first examine the switching of the lines close to the overloaded lines. The distance between two lines,  $l_1$  from bus  $i_1$  to  $j_1$  and  $l_2$  from bus  $i_2$  to  $j_2$ , can be defined as

$$d(l_1, l_2) = \min \{sp(i_1, i_2), sp(i_1, j_2), sp(j_1, i_2), sp(j_1, j_2)\} \quad (8)$$

where  $sp(i, j)$  is the length of the shortest path between buses  $i$  and  $j$ , i.e., the minimum number of lines to constitute a path from  $i$  to  $j$ . At each tree node, the search should examine the case of switching the corresponding line prior to that of keeping the line in its current status, e.g., value 0 of the line status variable is examined prior to value 1 if the line is currently in. Effectiveness of the above heuristics will be analyzed in Example 2 and 3 in Section V.

Although solving the LP problem at each leaf node is usually fast, considering the combinatorial growth of the search tree with respect to the number of binary variables, the number of LP problems to be solved in the search procedure could be large. However, since generation is fixed, the set of constraints for the base case and each contingency case contains a different set of phase angles. The LP problem at each leaf node can then be solved separately as a set of LP sub-problems, and the infeasibility of any sub-problem indicates the infeasibility of the leaf node. In our method, the base case LP sub-problem is first solved at each leaf node, so that the leaf node can be proved infeasible very quickly if it does not satisfy the base case security constraints. If the base case sub-problem is feasible, all the constraints of contingency cases are then solved as a single LP problem. In Example 3 of Section V, this procedure is shown to be more efficient than that of solving an LP sub-problem for each contingency case, because the latter needs to call and release the LP solver for each case. Another option to improve efficiency is to solve the LP sub-problems in a multi-processor environment, where each processor solves a few sub-problems, and the leaf node is proved infeasible and discarded once any processor finds an infeasible sub-problem.

For a practical system, it is usually difficult to estimate the performance of our method, because it depends on many factors including system topology, line capacities, and the set of contingency cases. In theory, a constraint satisfaction problem with binary and continuous variables is NP-hard in general [20]. For our method, the computational requirements include two major parts: expanding the tree and solving the LP problems at leaf nodes. In tree expansion, the number of leaf nodes depends on the number of lines allowed to be out, i.e.,  $M$  in (6), and the size of the switching candidate set  $E_C$  in (7). As an example, suppose all lines are currently in, one line is allowed to

be switched out ( $M = 1$ ), and  $K$  lines are switching candidates, then the number of leaf nodes is at most  $K + 1$ , corresponding to the  $K + 1$  possible topologies. At leaf nodes, although an LP problem cannot necessarily be solved within polynomial time [28], the time performance is usually efficient enough for practical use. Moreover, using the heuristics mentioned above to decide the branching sequence of binary variables, the first few feasible solutions could be obtained after visiting only a few leaf nodes. Computational requirements of the method will be analyzed by using numerical examples in Section V. In practice, our method can be integrated with the Energy Management System (EMS) as part of the online sequence: once the "topology processor" finds a line outage, the "state estimator" runs, then our method can be executed to search for corrective switching actions.

## V. NUMERICAL EXAMPLES

Our method is implemented using Comet version 2.1 [29] on a personal computer with an Intel Xeon 1.60-GHz CPU and 8 GB of memory. Comet is a hybrid optimization system with the capabilities of both CP and LP. The CP solver accepts both discrete and continuous decision variables, and can solve both variables using constraint propagation based on consistency techniques, though the implementation for continuous variables is preliminary. In many other software packages, e.g., ILOG CP, continuous decision variables cannot even be accepted.

In this section, three numerical examples are provided to analyze the performance of our method presented in Section IV. A three-bus example is used to illustrate the search procedure. The IEEE 30-bus example analyzes the search performance with respect to the number of lines allowed to be out, and the heuristics in the branching sequence of binary variables. The method is also demonstrated on the IEEE 118-bus system. For all examples, the lower and upper bounds of phase angles in (4) are  $-0.79$  and  $0.79$  radians, or  $-45$  and  $45$  degrees, implying that the differences between any two phase angles cannot exceed 90 degrees; and the threshold of phase angle differences between the ends of lines in (5) is  $0.52$  radians, or 30 degrees. The above phase angle requirements can usually be met by a system in its steady state. Detailed data and numerical results are provided on our website [30].

*Example 1:* This example first illustrates the search procedure of our method, then compares it with the alternative procedure of branching on both binary and continuous variables. Assume that a certain line outage results in a three-bus system shown in Fig. 1. The system has two generators at buses 1 and 3, and bus 1 is the slack bus. The generation and load at each bus are given in Table I. The lines are numbered in the following sequence: (1, 2), (1, 3), and (2, 3). For all these lines, the current statuses are "in," the reactances are 1 per unit with 100 MVA base, and the long-term and emergency ratings are, respectively, 30 MW and 36 MW. The contingency cases are given in Table II, where  $N - 2$  cases are considered for line 2, line 3, and generator 2. In the following search procedures, all three lines are switching candidates, the maximum number of lines to be out is three, and the sequence to branch on binary variables is set to be the sequence of lines.

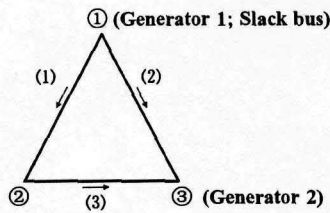


Fig. 1. Three-bus system. The reference direction of each line is indicated by an arrow. Bus indices in circles; line indices in parentheses.

TABLE I  
GENERATION AND LOAD LEVELS FOR THE THREE-BUS SYSTEM  
("I" INDICATES "NONE")

Bus No.	1	2	3
Generation Level (MW)	32	0	3
Load Level (MW)	2	30	3

TABLE II  
LIST OF CONTINGENCY CASES FOR THE THREE-BUS SYSTEM  
("I" INDICATES "NONE")

Contingency case No.	1	2	3	4	5	6	7
Line(s) being out	1	2	3	/	2, 3	2	3
Generator(s) being out	/	/	/	2	/	2	2

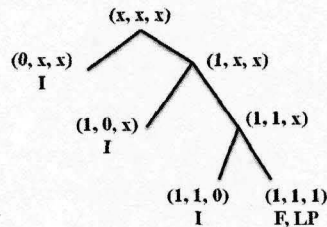


Fig. 2. Search tree for the three-bus system. "F" indicates a feasible solution, "I" indicates an infeasible node that is pruned, and "LP" indicates a feasibility checking by using LP methods.

First, our search procedure is illustrated. The search tree is shown in Fig. 2, where "x" indicates an unfixed binary variable. In the tree, the node  $(0, x, x)$  is proved infeasible using constraint propagation, with only one binary variable fixed. This is because by switching out line 1, or  $(1, 2)$ , bus 1 will be isolated from buses 2 and 3 in case line  $(1, 3)$  is out in a contingency. Then since generation is assumed to be fixed at all buses, there will not be sufficient power to supply the load at bus 2. For a similar reason, the nodes  $(1, 0, x)$  and  $(1, 1, 0)$  are also infeasible. Therefore, the only LP problem is solved at the leaf node  $(1, 1, 1)$ , which turns out to be feasible. This LP problem has the constraints for the base case and all the seven contingency cases listed in Table II. The constraints for each case contain different phase angle variables, and can actually be viewed as separate LP sub-problems. For example, the power balance constraints, (2a) and (2b), for the base case are

$$\begin{aligned} (\theta_1^0 - \theta_2^0) + (\theta_1^0 - \theta_3^0) &= 30, \\ (\theta_2^0 - \theta_1^0) + (\theta_2^0 - \theta_3^0) &= -30, \\ (\theta_3^0 - \theta_1^0) + (\theta_3^0 - \theta_2^0) &= 0 \end{aligned} \quad (9)$$

TABLE III  
GENERATOR DISPATCH FOR IEEE 30-BUS SYSTEM

Generator	1	2	3	4	5	6
Bus	1	2	5	8	11	13
Generation level (MW)	93.4	52	48	55	18	17

and for contingency case 1 with line  $(1, 2)$  out, the power balance constraints are

$$\begin{aligned} \theta_1^1 - \theta_3^1 &= 30, \\ \theta_2^1 - \theta_3^1 &= -30, \\ (\theta_3^1 - \theta_1^1) + (\theta_3^1 - \theta_2^1) &= 0. \end{aligned} \quad (10)$$

The LP problem at the leaf node  $(1, 1, 1)$  can thus be decomposed into eight sub-problems, each corresponding to one case with one set of phase angles. These sub-problems can be solved separately, but the resulting procedure requires repeatedly calling and releasing LP solvers and may not be efficient, as will be shown in Example 3.

Next, the alternative procedure to branch on both binary and continuous variables is tested to compare with our method. The search first branches on binary variables until they are fixed. At each resulting node, the continuous variable with the largest interval domain is branched on by equally dividing the interval into two. In the software Comet, the stopping criterion of branching on a continuous variable is that the length of its interval domain is no more than  $10^{-7}$ . This procedure is not efficient, and terminates with a tree of depth 426 with the same solutions as those of our method. The solution time of this procedure is 0.062 s, while our method takes only 0.015 s to terminate. Therefore in our method, the continuous phase angles are solved by using LP techniques.

*Example 2:* This example is to analyze the performance of our method with respect to the number of lines to be out, and the heuristics in the branching sequence of binary variables. Consider the IEEE 30-bus system with 41 lines and six generators from [31], where the system topology, line reactances, and loads are given. Assume that line  $(6, 10)$  fails, and in the base case, all the other lines are "in." Bus 1 is the slack bus, and the generation dispatch is from [13] with the generation at the slack bus adjusted for power balance, as given in Table III. The long-term line ratings are based on [13] with certain ratings increased to create spare capacities, the corresponding emergency ratings are 20% higher, and all ratings are given in [30]. The system has three radial lines, lines 13, 16, and 34, and their outages are not considered as  $N - 1$  cases. The  $N - 2$  cases are considered for lines and generators selected from the area containing buses 2, 3, 4, 6, 12, 13, and 15 with relatively dense connections:  $E_2 = \{3, 4, 6, 7, 15, 18\}$ , and  $G_2 = \{6\}$ .

Load flow analysis shows that the above system is secure in the base case, but in the  $N - 1$  contingency case with line  $(9, 10)$  out, line  $(16, 17)$  will be overloaded. Our method is used to search for line switching operations to relieve this overload. Except line  $(6, 10)$  in failure, all other lines are switching candidates, and the sequence to branch on the binary variables is based on LODFs. Table IV shows the results with different values of  $M$ , i.e., the number of lines allowed to be out, to compare the corresponding performances. With  $M = 1$ , the feasible

TABLE IV  
RESULTS FOR IEEE 30-BUS SYSTEM (CPU TIMES IN SECONDS)

Max No. of open lines $M$	1	2	3	10
CPU time for 1 <sup>st</sup> solution	2.5	7.7	22.9	38.5
Total No. of solutions	1	2	2	2
Switching actions of each solution	{15 out}	{15 out}		
		{15 out, 18 out}		
Total CPU time	8.3	29.3	63.7	93.9

TABLE V  
RESULTS FOR IEEE 30-BUS SYSTEM WITHOUT USING THE HEURISTIC IN BRANCHING SEQUENCE (CPU TIMES IN SECONDS)

Max No. of open lines $M$	1	2	3	10
CPU time for 1 <sup>st</sup> solution	4.1	16.6	38.8	53.9
Total CPU time	8.2	27.4	52.2	66.9

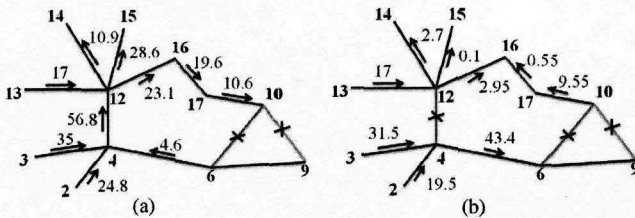


Fig. 3. Relevant part of the 30-bus system after the outage of line (6, 10), with line (9, 10) out in a contingency. (a) Before switching out line (4, 12). (b) After switching out line (4, 12). Bus indices in bold; line flows in MW.

operation is to switch out line 15, or (4, 12). With  $M = 2$ , another solution is obtained, which is to switch out both lines 15 and 18. However, there are no more feasible solutions with  $M$  increased to 10. This demonstrates that it is reasonable to reduce the search space by using a small  $M$  in constraint (6), in that allowing more lines to be out is not likely to result in more feasible solutions.

To analyze the impacts of the branching sequence of binary variables based on LODFs, we rerun the method with the sequence set to be the increasing order of line indices, and the results are shown in Table V. Although the total CPU times are smaller than those in Table IV because of the savings in the sorting procedure before branching on each variable, the times for the first feasible solution become much longer. This demonstrates that the heuristics in the branching sequence of binary variables can effectively improve the performance of the method in obtaining the first feasible solution.

To illustrate the switching of line (4, 12), the relevant part of the system is shown in Fig. 3, where the blue line (6, 10) is in failure and the pink line (9, 10) is out in an  $N - 1$  contingency. In Fig. 3(a) before switching out line (4, 12), its flow of 56.8 MW goes into bus 12, and part of this flow goes to line (12, 16) and the red line (16, 17) which is overloaded. In Fig. 3(b), the orange line (4, 12) is switched out and its large flow into bus 12 is eliminated; hence, line (16, 17) in green is no longer overloaded. Also, the flow on line (4, 6) changes direction, so that the original flow on line (4, 12) is redirected to other parts of the system.

*Example 3:* This example is to demonstrate the effectiveness of our method on the IEEE 118-bus system. The system has 186

TABLE VI  
N-2 CASES AND RADIAL LINES FOR IEEE 118-BUS SYSTEM

$E_2$	165, 166, 168, 169
$G_2$	18 (Generator 18 is at bus 103)
Radial lines	7, 9, 184, 183, 113, 133, 134, 176, 177

TABLE VII  
RESULTS FOR IEEE 118-BUS SYSTEM (CPU TIMES IN SECONDS)

Max No. of open lines $M$	1	2
CPU time for 1 <sup>st</sup> solution	44.8	285.1
Total No. of solutions	1	5
Switching actions of each solution	{21 in, 51 out}	{21 in, 51 out, 105 out} {21 in, 51 out, 109 out} {21 in, 51 out} {21 in, 66 out, 102 out} {21 in, 67 out, 102 out}
Total CPU time	416.8	2636.3

lines and 54 generators, and the topology, line reactances, generation, and load levels are given in [31]. Assume that line 45 fails, and in the base case, all the other lines are "in," except that line 21 is assumed to be "out" to demonstrate that our method is able to handle switching in of lines. Bus 69 is the slack bus. The long-term line ratings are based on [14] with certain ratings increased to create spare capacities, the corresponding emergency ratings are 20% higher, and all ratings are given in [30]. The  $N - 2$  contingency cases are considered within the area containing buses 103 to 106 with relatively dense connections, and the sets  $E_2$  and  $G_2$  as well as the radial line indices are given in Table VI.

Load flow analysis shows that the above system is not  $N - 1$  secure, because line 46 is overloaded when line 50 is out in a contingency. To test our method, a set of 15 candidates is used, constituting 8% of the total 186 lines. We select the 14 lines with the largest LODFs with respect to line 46 among all lines satisfying the topological necessary condition introduced in Section IV-B, and line 21 which is currently out but can be possibly switched in. The resulting candidate lines are 21, 30, 51, 53, 57, 62, 63, 66, 67, 68, 96, 102, 105, 109, and 110. The branching sequence of binary variables is based on the distances of lines from line 46 as defined by (8). Table VII shows the results of our method. With  $M = 1$ , i.e., allowing one line to be out, the only feasible solution is to switch in line 21 and switch out line 51. The time for obtaining this solution is 44.8 s, including 17.1 s to reach the first leaf node and 27.7 s to solve the LP problem, which turns out to be feasible. With  $M = 2$ , there are five feasible solutions. Three of them include the above switching actions of line 21 and 51, while the other two are alternative options without switching out line 51. It is interesting to note from system data that lines 66 and 67 are both from bus 42 to 49; thus, switching out either 66 or 67 actually changes the reactance between buses 42 and 49 instead of changing system topology, and this leads to a redistribution of load flow to relieve the overload.

Two other cases in running the method are analyzed for the  $M = 1$  case. First, if the branching sequence is based on LODFs, the time for obtaining the first feasible solution is 124.3 s, not as fast as the above result of 44.8 s. Second, for the

LP problem at the first leaf node, the procedure to separately solve an LP sub-problem for each case takes 211.6 s in all. This is not as efficient as the result of 27.7 s by using the procedure in our method as described in Section IV-B.

## VI. CONCLUSIONS

In this paper, DC flow feasibility constraints with line status and phase angle variables are formulated in a unified way for the base case, the  $N - 1$  and selected  $N - 2$  contingency cases. In our method based on constraint programming, binary line status variables are branched on in a tree search, and at each leaf node, the constraints become linear of continuous phase angles and are examined by linear programming methods. Testing results on systems with up to 118 buses demonstrate the effectiveness of the method.

Potential extensions of our method include using certain criteria to rank the feasible solutions, and implementing the method in a multi-processor environment. In addition, the issues of AC flow feasibility and transient stability of switching actions are important future research topics.

## ACKNOWLEDGMENT

The authors would like to thank Prof. X. Guan and Dr. Q. Jia of Tsinghua University and Dr. E. Litvinov, Dr. T. Zheng, and Dr. F. Zhao of ISO-New England for contributing their valuable suggestions.

## REFERENCES

- [1] A. A. Maci, B. F. Wollenberg, and M. H. Hesse, "Corrective control of power system flows by line and bus-bar switching," *IEEE Trans. Power Syst.*, vol. 1, no. 3, pp. 258–264, Aug. 1986.
- [2] E. Lobato, F. Echavarren, L. Rouco, M. I. Navarrete, R. Casanova, and G. Lopez, "A mixed-integer LP based network topology optimization algorithm for overload alleviation," in *Proc. 2003 IEEE Bologna Power Tech Conf.*, Jun. 2003, vol. 2.
- [3] W. Shao and V. Vittal, "BIP based OPF for line and bus-bar switching to relieve overloads and voltage violations," in *Proc. 2006 IEEE PES Power System Conf. Expo.*, Oct. 2006.
- [4] E. B. Makram, D. P. Thornton, and H. E. Brown, "Selection of lines to be switched to eliminate overloaded lines using a Z-matrix method," *IEEE Trans. Power Syst.*, vol. 4, no. 2, pp. 653–661, May 1989.
- [5] W. Shao and V. Vittal, "Corrective switching algorithm for relieving overloads and voltage violations," *IEEE Trans. Power Syst.*, vol. 20, no. 4, pp. 1877–1885, Nov. 2005.
- [6] W. Shao and V. Vittal, "LP-based OPF for corrective FACTS control to relieve overloads and voltage violations," *IEEE Trans. Power Syst.*, vol. 21, no. 4, pp. 1832–1839, Nov. 2006.
- [7] NERC Reliability Functional Model Technical Document, ver. 3, p. 30. [Online]. Available: [http://www.nerc.com/files/Functional\\_Model\\_Technical\\_Document\\_V3\\_for\\_OC\\_and\\_PC\\_approval\\_06Dec06.pdf](http://www.nerc.com/files/Functional_Model_Technical_Document_V3_for_OC_and_PC_approval_06Dec06.pdf).
- [8] U.S. Federal Energy Regulatory Commission, Mandatory Reliability Standards for the Bulk-Power System, Order No. 693, 72 FR 16516, Apr. 4, 2007.
- [9] R. Wang and R. H. Lasseter, "Re-dispatching generation to increase power system security margin and support low voltage bus," *IEEE Trans. Power Syst.*, vol. 15, no. 2, pp. 496–501, May 2000.
- [10] O. E. Moya, "A spinning reserve, load shedding and economic dispatch solution by Bender's decomposition," *IEEE Trans. Power Syst.*, vol. 20, no. 1, pp. 1805–1810, Feb. 2005.
- [11] H. You, V. Vittal, and Z. Yang, "Self-healing in power systems: An approach using islanding and rate of frequency decline-based load shedding," *IEEE Trans. Power Syst.*, vol. 18, no. 1, pp. 174–181, Feb. 2003.
- [12] H. You, V. Vittal, and X. Wang, "Slow coherency-based islanding," *IEEE Trans. Power Syst.*, vol. 19, no. 1, pp. 483–491, Feb. 2004.
- [13] Q. Zhao, K. Sun, D. Zheng, J. Ma, and Q. Lu, "A study of system splitting strategies for islanding operations of power system: A two-phase method based on OBDDs," *IEEE Trans. Power Syst.*, vol. 18, no. 4, pp. 1556–1565, Nov. 2003.
- [14] K. Sun, D. Zheng, and Q. Lu, "Splitting strategies for islanding operation of large-scale power systems using OBDD based methods," *IEEE Trans. Power Syst.*, vol. 18, no. 2, pp. 912–923, May 2003.
- [15] B. Bacher and H. Glavitsch, "Network topology optimization with security constraints," *IEEE Trans. Power Syst.*, vol. 1, no. 4, pp. 103–111, Nov. 1986.
- [16] R. Bacher and H. Glavitsch, "Loss reduction by network switching," *IEEE Trans. Power Syst.*, vol. 3, no. 2, pp. 447–454, May 1998.
- [17] E. B. Fisher, R. P. O'Neill, and M. C. Ferris, "Optimal transmission switching," *IEEE Trans. Power Syst.*, vol. 23, no. 3, pp. 1346–1355, Aug. 2008.
- [18] G. Schnyder and H. Glavitsch, "Security enhancement using an optimal switching power flow," *IEEE Trans. Power Syst.*, vol. 5, no. 2, pp. 674–681, May 1990.
- [19] K. W. Hedman, R. P. O'Neill, E. B. Fisher, and S. S. Oren, "Optimal transmission switching with contingency analysis," *IEEE Trans. Power Syst.*, vol. 24, no. 3, pp. 1577–1586, Aug. 2009.
- [20] F. Rossi, P. Van Beek, and T. Walsh, *Handbook of Constraint Programming*. New York: Elsevier, 2006.
- [21] P. Kundur, *Power System Stability and Control*. New York: McGraw-Hill, 1994.
- [22] A. J. Wood and B. F. Wollenberg, *Power Generation Operation and Control*, 2nd ed. New York: Wiley, 2003.
- [23] K. W. Hedman, M. C. Ferris, R. O'Neill, E. B. Fisher, and S. S. Oren, "Co-optimization of generation unit commitment and transmission switching with N-1 reliability," *IEEE Trans. Power Syst.*, vol. 25, no. 2, pp. 1052–1063, May 2010.
- [24] L. Chen, H. Tozyo, Y. Tada, H. Okamoto, and R. Tanabe, "Reconfiguration of transmission systems with transient stability constraints," in *Proc. IEEE/PES Winter Meeting 2000*, Singapore, Jan. 2000, pp. 1320–1324.
- [25] T. Creemers, L. R. Giral, J. Riera, C. Ferrarons, J. Roca, and X. Corbella, "Constraint-based maintenance scheduling on an electric power-distribution network," in *Proc. 3rd Int. Conf. Practical Applications of Prolog*, Paris, France, Apr. 1995, pp. 135–144.
- [26] J. R. Diaz, G. Gutierrez, C. A. Olarte, and C. Rueda, "Using constraint programming for reconfiguration of electrical power distribution networks," *Lecture Notes in Computer Science*, vol. 3389, pp. 263–276, 2005.
- [27] P. A. Ruiz and P. W. Sauer, "Voltage and reactive power estimation for contingency analysis using sensitivities," *IEEE Trans. Power Syst.*, vol. 22, no. 2, pp. 639–647, May 2007.
- [28] D. G. Luenberger, *Linear and Nonlinear Programming*, 2nd ed. New York: Springer, 2003.
- [29] Dynadec Corporation, The Comet Hybrid Optimization Platform. [Online]. Available: <http://dynadec.com/support/downloads/>.
- [30] Center for Intelligent and Networked Systems, Tsinghua University, Power System Testing Data and Results. [Online]. Available: <http://cfins.au.tsinghua.edu.cn/files/powersys/>.
- [31] University of Washington, Power Systems Test Case Archive. [Online]. Available: <http://www.ee.washington.edu/research/pstca/>.



**Mingyang Li** (S'08) received the B.E. degree in automatic control in 2006 from Tsinghua University, Beijing, China. He is currently pursuing the Ph.D. degree in the Center for Intelligent and Networked Systems (CFINS), Department of Automation, Tsinghua University.

His research interests include modeling and optimization of complex networked systems, and power system security.





**Peter B. Luh** (S'77-M'80-SM'91-F'95) received the Ph.D. degree in applied mathematics from Harvard University, Cambridge, MA, in 1980.

He has been with the University of Connecticut, Storrs, since then. Currently he is the SNET Professor of Communications and Information Technologies. His interests include design of auction methods for electricity markets; electricity load and price forecasting with demand management; control and optimization of boilers and energy systems; optimized resource management and coordination

for sustainable, green, and safe buildings; planning, scheduling, and coordination of design, manufacturing, and service activities; decision-making under uncertain, distributed, or antagonistic environments; and mathematical optimization for large-scale problems.

Dr. Luh is Vice President of Publication Activities for the IEEE Robotics and Automation Society, the founding Editor-in-Chief of the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING (2003-2008), and was the Editor-in-Chief of the IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION (1999-2003).



**Qianchuan Zhao** (M'06-SM'08) received the B.E. degree in automatic control in 1992, the B.S. degree in applied mathematics in 1992, and the Ph.D. degree in control theory and its applications in 1996, all from Tsinghua University, Beijing, China.

He is currently a Professor and Associate Director of the Center for Intelligent and Networked Systems, Department of Automation, Tsinghua University.

**Xiaochuan Luo** (M'97) received the B.S. and M.S. degrees from the Shanghai Jiao Tong University, Shanghai, China, in 1992 and 1995, respectively, and the Ph.D. degree from the Texas A&M University, College Station, TX, in 2000.

His major interests include power system planning and operations, power system reliability, and visualization of bulk power system. He is currently a Principal Analyst with ISO New England Inc., Holyoke, MA.



**Laurent D. Michel** received the Ph.D. degree in computer science from Brown University, Providence, RI, in 1999.

He is currently holding an Associate Professor position in the Computer Science and Engineering Department at the University of Connecticut, Storrs. He specializes in combinatorial optimization with a particular emphasis on constraint programming. He has co-authored two monographs and more than 60 papers.

Dr. Michel sits on the Editorial Board of *Constraints*, *Mathematical Programming Computation*, and *Constraint Letters*.