2011 50th IEEE Conference on Decision and Control and
European Control Conference (CDC-ECC)
Orlando, FL, USA, December 12-15, 2011

Electric Power Network Security Analysis via Minimum Cut Relaxation

Kin Cheong Sou, Henrik Sandberg and Karl Henrik Johansson

Abstract— **In this paper an efficient computation scheme for analyzing the security of power transmission networks is presented. In order to strategically allocate protection devices in the network, the problem of finding the sparsest stealthy false data attack to the state estimator is studied. While the attack search problem is traditionally solved as a generic constrained cardinality minimization problem, this paper exploits the problem structure intrinsic to the power network application to obtain a polynomial time approximate algorithm based on a minimum cut relaxation. Experiment results from realistic test cases are presented to demonstrate the efficiency and accuracy of the proposed algorithm.**

I. INTRODUCTION

A. *SCADA Security and Power Transmission*

A modern society relies critically on the proper operation of the electric power distribution system, which is supervised and control through the *Supervisory Control And Data Acquisition* (SCADA) systems. These systems are highly dependent on control algorithms but also on computerized and networked information. The resilience of power system on this infrastructure, makes it more susceptible not only to operational errors but also to external attacks.

SCADA systems measure data through remote devices all over the grid and gathers them at a control center through communication channels. There computer processing takes place and control commands are sent back to the system. The vulnerabilities that are introduced could be exploited by malicious attackers. Many reports concerning the vulnerabilities due to cyber attacks [1], [2] are given, but also real incidents (e.g. [3]) affirm the importance of this issue.

B. *State Estimation of Power Systems*

State estimators (such as power flow estimators) in power systems are currently used to, for example, detect faulty equipment and to route power flows. The estimators are currently located in control centers. Large numbers of measurements are sent to the centers over unencrypted communication channels, which are susceptible to false-data attacks. Therefore the security of the estimator becomes an important issue [4]–[8].

In this paper the ***linearized*** power network state estimation problem is considered. The network can be modeled as a graph with n nodes (i.e., buses) and m_a directed arcs (i.e., transmission lines) on which power flows. The flow can be negative, meaning that the actual direction of the flow is opposite to the direction of the arc. The (arc-to-node) incidence matrix $A \in \mathbb{R}^{n\times m_a}$ describes the topology:

$$\forall j = 1,\ldots,m_a \quad A(i,j) = \begin{cases} 1 & \text{if arc } j \text{ starts at node } i \\ -1 & \text{if arc } j \text{ ends at node } i \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

The states of the power network are the phase angles at the nodes, grouped into a vector $\delta \in \mathbb{R}^n$. The measurements, denoted as z, contain three parts: (a) arc power flows, (b) their negative copies, and (c) external power injections into nodes. The arc power flows are related to δ by the linearized formula $DA^T\delta$, where $D \in \mathbb{R}^{m_a\times m_a}$ is a diagonal matrix whose diagonal entries are the reciprocal of the reactance of the arcs [9]. By the flow conservation law, the external power injections into nodes are also related to the states δ as $\mathscr{L}\delta = ADA^T\delta$, where $\mathscr{L}$ is the weighted Laplacian of the graph. To summarize, the states and measurements of a power network are related by

$$z = H\delta, \quad \text{where} \quad H \triangleq \begin{bmatrix} DA^T \\ -DA^T \\ ADA^T \end{bmatrix}_{m\times n} \tag{2}$$

with $m = n + 2m_a$. Hence, H has more rows than columns by construction. Note that for the convenience of exposition in this paper, the H matrix defined in (2) is a row permuted version of the standard H matrix defined in the IEEE benchmarks. Finally, it is assumed in this paper that D is positive definite (i.e., positive, inductive reactance). This is not always true, even though it is uncommon to encounter negative reactance. Out of the 19 test cases from MATPOWER [10], there is one case (300-bus case) having one arc with negative reactance.

To obtain an estimate of the states δ using the measurements z, the relation in (2) can be inverted, for instance, by solving a least squares problem. To detect whether the measurements z are reliable, a bad data detection (BDD) test is typically performed [9]. If the residual $\left\|(I - H(H^TH)^{+}H^T)z\right\|$ is too big, then an alarm is triggered.

C. *Sparsest Stealthy False Data Attacks*

This paper considers the hypothetical scenario where the measurements of a power network are susceptible to additive false data attacks. The measurement vector that the state estimator receives is of the form $z + \Delta z$, where $z = H\delta$ is the vector of true power flows and Δz is the measurement attack vector. If an attack vector Δz is such that it is impossible to satisfy $\Delta z = H\Delta\delta$ for any $\Delta\delta$, then the BDD alarm would

The authors are with the ACCESS Linnaeus Center and the Automatic Control Lab, the School of Electrical Engineering, (KTH) Royal Institute of Technology, Sweden. {sou,hsan,kallej}@kth.se

This work is supported by the European Commission through the VIKING project, the Swedish Foundation for Strategic Research (SSF) and the Knut and Alice Wallenberg Foundation.

978-1-61284-799-3/11/$26.00 ©2011 IEEE

be triggered and the attack would fail. To avoid BDD alarm trigger, [4] introduces the notion of stealthy false data attack, amounting to a change of decision variables of the form

$$\Delta z = H\Delta\delta \quad \text{for some} \quad \Delta\delta \tag{3}$$

To make the attack vector Δz stealthy, attacks might be required on multiple measurements even if the attacker is interested in attacking only one measurement (e.g. an arc). Since each attack on a measurement entails some risk, the attacker would be interested in finding the sparsest stealthy attack. From the viewpoint of a defender, the question of the sparsest stealthy attack is also important because it identifies the measurements which are vulnerable to attack. Then the analysis result can be used to strategically place protective equipment to its best effect (e.g. [7]). This paper considers the following sparsest stealthy false data attack problem of a single measurement $k \in \{1, 2, \ldots, m\}$, initially studied in [6].

$$\begin{array}{rcll} \alpha_k & \triangleq & \underset{\Delta\delta}{\min} & \|H\Delta\delta\|_0 \\ & & \text{subject to} & H(k,:)\Delta\delta = 1 \end{array} \tag{4}$$

where $\|\cdot\|_0$ denotes the cardinality of a vector, and $H(k,:)$ follows the "MATLAB convention" denoting the k^{th} row of H. It is assumed that there is no column of A which is all-zero. This implies that H does not have any all-zero row (i.e., empty measurement). Hence, the problem in (4) is always feasible. The minimum objective value in (4), denoted as α_k, is defined in [6] as the ***security index*** of measurement k. α_k provides the absolute lower bound on the number of measurements that need to be compromised in a stealthly attack on measurement k. Hence, the knowledge of the security indices of the measurements allows the power network operator to pinpoint the location where protective mechanisms such as encryptions and PMUs should be placed. For the rest of the paper, if $k \in [1, 2m_b]$, then the corresponding α_k is referred to as arc measurement security index. On the other hand, if $k \in [2m_b + 1, m]$, then the corresponding α_k is called injection measurement security index.

D. Previous Works

The security index problem in (4) is NP-hard because it is a cardinality minimization problem with a nontrivial constraint. [4] reports the use of matching pursuit for obtaining suboptimal solutions to (4). [6] proposes a simple and efficient security index upper bounding scheme – among all columns of H with a nonzero entry in the k^{th} row, the sparsest column is chosen as a suboptimal solution $H\Delta\delta$ to (4). [6] also reports using LASSO [11] to suboptimally solve (4). To solve (4) to optimality, a mixed integer linear program (MILP) can be set up and solved using, for instance, the branch-and-bound algorithm (e.g. [12]). Alternatively, [7] proposes an enumerative algorithm for (4).

The mentioned previous works are "generic" in the sense that (4) is solved without taking into account the specific structure of the H matrix. The proposed scheme, however, is restricted to the case where H has the form in (2). More recently there are also results focusing on the structure of H [13], [14]. However, the considered problems in these work are different from the problem in this paper. This paper shows that the optimal solution to (4) is structured, as summarized in Proposition 1 in Section II. By exploiting the structure of the optimal solution, efficient computation schemes for (4) can be derived in Section III. Section III also shows that in the special case where the H matrix does not have the ADA^T block (i.e., no injection measurement), then (4) can be solved exactly in polynomial time. Finally, numerical experiments in Section IV demonstrate that the proposed schemes outperform the previous works.

II. Node Partitioning Formulation of the Security Index Computation Problem

A. Binary Characterization of Optimal Solutions

From the point of view of achieving the smallest value of $\|H\Delta\delta\|_0$ in (4), an all-zero vector would be the best choice. However, this choice violates the constraint $H(k,:)\Delta\delta = 1$. The next logical guess would be an n-vector whose entries are either 0 or β, for some $\beta \neq 0$. This turns out to be the structure of at least one optimal solution to (4), due to the specific setup in this paper. The following statement formally verifies the intuition.

Proposition 1: Let H in (4) have the structure in (2), and assume that D is positive definite. Then there is at least one optimal solution of (4) of the form $\beta\Delta\delta_b$ where $\Delta\delta_b$ is an n-vector whose entries are either 0 or 1, and β is a scalar such that $\beta H(k,:)\Delta\delta_b = 1$.

Proof: See [15]. ■

The above statement implies that, without loss of generality, the problem in (4) can be restricted to a node partitioning problem assigning the entries of $\Delta\delta$ to be either 0 or 1. This in turn simplifies the counting of injection attacks (i.e., part of the objective function value in (4)).

Corollary 1: For any 0-1 partition of the nodes specified by $\Delta\delta_b$ in Proposition 1, a node i is subjected to an injection attack if and only if its node value $\Delta\delta_b(i)$ is greater than that of at least one of its neighbors.

Proof: See [15]. ■

For instance, the above statement rule out the arc flow balance situation depicted in Fig. 1.

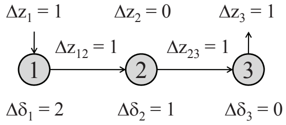

Fig. 1. Arc flows are balanced at node 2 even if both of its incident arcs have nonzero flows. This situation cannot happen if the node values can only be either 0 or 1.

B. Node Partitioning Formulations for Security Index

Proposition 1 provides an equivalent but simplified graph interpretation of (4) for the security index of an arc.

- Let (i, j) (from i to j) be the targeted arc. Assign node value at i to be 1 and node value at j to be 0.

- Assign value 0 or 1 to the rest of the nodes to minimize cost, to be described in the next step.
- Cost is the number of arcs with different start/end node values, plus the number of nodes incident to such arcs.

For reference, the above problem will be referred to as the ***minimum cost node partitioning problem*** for the rest of this paper.

Corollary 1 also provides an equivalent but simplified graph interpretation of (4) in the case of injection attack.

- Let node i be subjected to an injection attack. For each arc incident to i, calculate the arc measurement security index (with the node value at i set to 1).
- The minimum of the above indices is the security index of the injection measurement at node i.

III. Efficient Security Index Computation Based on Node Partitioning

A. *Minimum Cut Problem on a Weighted Directed Graph*

The rest of this paper will make use of the solving of a standard graph optimization problem called minimum cut problem (MIN-CUT) [12], which is now briefly reviewed. Consider a directed graph defined by its node set $\mathscr{N}$ and arc set $\mathscr{A}$. The arcs are weighted with w_{ij} for all $(i,j) \in \mathscr{A}$. Let s and t be two different nodes called source and sink respectively. An "$s-t$ cut" is defined as a partition of the nodes into two disjoint sets S and $\mathscr{N} \setminus S$ such that $s \in S$ and $t \in \mathscr{N} \setminus S$. For any cut, the associated "cut capacity" is defined as

$$C(S) \triangleq \sum_{\{(i,j) \in \mathscr{A} \mid i \in S, j \notin S\}} w_{ij}$$

The problem of $s-t$ MIN-CUT (or MIN-CUT for short) is the problem of finding the $s-t$ cut with the minimum cut capacity.

$$\begin{array}{lll} \text{(MIN-CUT)} & \underset{S}{\text{minimize}} & C(S) \\ & \text{subject to} & S \text{ and } \mathscr{N} \setminus S \text{ is an } s-t \text{ cut} \end{array} \tag{5}$$

The MIN-CUT problem above can be solved in polynomial time using algorithms algorithms such as [16] or [17] via the max-flow/min-cut theorem [12]. In addition, the set of all optimal partitions of a MIN-CUT problem can be efficiently characterized and enumerated. This is due to the results of [18], [19].

B. *Security Index Upper Bounding via MIN-CUT Relaxation*

The minimum cost node partitioning problem in Section II-B is, unfortunately, still a combinatorial optimization problem. In response, this paper proposes the following polynomial-time approximate algorithm, based on solving a MIN-CUT problem described in (5).

Security index of an arc measurement – upper bounding via MIN-CUT relaxation

Step 1

Define a directed graph $\mathscr{G}$ as follows. Whenever (i,j) is an arc of the original power network graph, both (i,j) and (j,i) will be arcs of the new graph $\mathscr{G}$. Let the weights on all arcs in $\mathscr{G}$ be 2.

Step 2

Denote (s,t) as the targeted arc of which the security index is considered. Let s and t be, respectively, the source and sink nodes in $\mathscr{G}$ defined in step 1.

Step 3

Solve the $s-t$ MIN-CUT problem on $\mathscr{G}$. Let $\Delta\delta_{mc}$ be the optimal MIN-CUT partition, $\|H\Delta\delta_{mc}\|_0$ is an upper bound of the security index of the arc (s,t).

Step 4

Enumerate all optimal MIN-CUT partitions for the best security index upper bound.

In step 1 of the above procedure, the opposite direction arcs are added to account for the fact that the power flows on a power network can be positive or negative, while the flows in the standard MIN-CUT setting are always nonnegative. The arc weights are chosen to be 2 to account for the fact that whenever $\Delta\delta(i) \neq \Delta\delta(j)$, both arc (i,j) and (j,i) will require an arc attack (cf. (2)). In step 2, the designation of s and t guarantees that the MIN-CUT partition in step 3 is feasible to the original minimum cost node partitioning problem.

Modulus the specific details of step 1 and step 2, the MIN-CUT problem in step 3 is a relaxed version of the minimum cost node partitioning problem in Section II-B, ***without taking into account the cost with the injection measurements***. Hence, a security index upper bound can be obtained by solving the MIN-CUT problem. Note that the number of injection attacks is monotonically (though not necessarily linearly) non-decreasing with the number of arc attacks. Hence, minimizing only the number of arc attacks (as in MIN-CUT) tends to make the number of injection attacks small. It will be demonstrated in Section IV that, with the enumeration in Step 4, the proposed approximate algorithm is highly accurate and efficient. However, at this moment the rigorous justification for the accuracy is still unknown. In fact, in the worst case even if with the enumeration in Step 4, the proposed upper bound can still be strict. This is illustrated in Fig. 2.

Finally, notice that the MIN-CUT upper bounding procedure can be used to find upper bounds for injection security index as well. This is done simply by following the injection measurement security index formulation in Section II-B (i.e., the last two bullets), with the MIN-CUT bounding replacing the actual arc security index computation.

C. *Enforcing Encryption Protection*

To protect the power network from malicious attacks, [5], [7] considered the placement of encryption devices at nodes. It is assumed that once encrypted, none of the arcs incident to the protected nodes can be attacked. Denote $\mathscr{P}$ as the set of arcs incident to the protected nodes (i.e., the corresponding attack vector components should be zero), then the protected version of the security index problem in (4) can be written

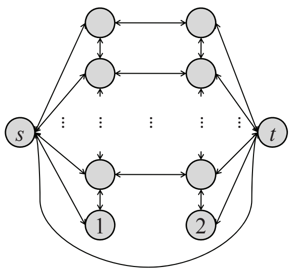

Fig. 2. A counterexample showing that the optimal MIN-CUT partition is not the optimal solution to the minimum cost node partitioning problem in Section II-B. s and t are the source and sink (i.e., the two nodes of the attacked arc). All arcs in the graph have weight 2. Because there is no arc connecting 1 and 2, the only MIN-CUT partition separating s and t cuts through all the arcs in the middle connecting the unlabeled nodes. However, when the number of unlabeled nodes in the middle columns is sufficiently large (i.e., larger than 3), the minimum cost partition either cuts through the arcs incident to s or those incident to t.

as

$$\begin{aligned} \underset{\Delta\delta}{\text{minimize}} \quad & \|H\Delta\delta\|_0 \\ \text{subject to} \quad & H(k,:)\Delta\delta = 1 \\ & H(j,:)\Delta\delta = 0, \quad \forall j \in \mathscr{P} \end{aligned} \tag{6}$$

It can be verified that (6) is also a node partitioning problem, as summarized by the following statement.

Proposition 2: Let H in (6) have the structure in (2), and assume that D is positive definite. Then there is at least one optimal solution of (6) of the form $\beta\Delta\delta_b$ where $\Delta\delta_b$ is an n-vector whose entries are either 0 or 1, and β is a scalar such that $\beta H(k,:)\Delta\delta_b = 1$.

Proof: See [15]. ■

The minimum cost node partitioning problem in Section II-B can be modified to allow the protection constraint. The protected arcs should cost "infinity" to cut In practice, any number larger than the number of rows of H in (2) suffices. With this modification, the targeted arc can be attacked if and only if the corresponding security index is less than infinity. Similar conclusion can be made with the injection attack case, stating that the injection measurement can be attacked if and only if its security index is less than infinity. Finally, note that the proposed MIN-CUT based upper bounding scheme in Section III-B also detects infeasible instances. This is summarized in the following statement.

Proposition 3: For the security index problem with protection in (6), the proposed MIN-CUT based upper bounding scheme in Section III-B (with protected arcs weight being infinity) finds an infinity upper bound if and only if the corresponding measurement cannot be attacked.

Proof: See [15]. ■

Reference [7] also considers other types of protection schemes, namely protecting arc and/or injection measurements. While the proposed framework can handle the scenario with protected arcs (as explained before), the situation with protected injection can create a difficulty for the proposed computation scheme. In particular, consider Fig. 1 in which the injection measurement to node 2 is protected (i.e., no (perturbed) external injection). Modulus a constant offset of the node values, the only feasible solution in this example requires that all three node values are different. This contradicts Proposition 2.

Finally, it is noted that the "protection constraint" in (6) might arise from a situation unrelated to encryption protection. There exist some "pseudo-measurements" in the power network that are known to have some fixed values, irrespective of the operation condition. Hence, the corresponding components of the attack vector must be zero. This is the same requirement as the protection constraint in (6).

D. Exact MIN-CUT Formulation in the Injection-free Case

In a restricted case where the H matrix in (2) does not contain any injection measurements, the MIN-CUT upper bounding scheme in Section III-B becomes ***exact***. The following statement provides the rationale.

Proposition 4: Denote $H_{arc} \triangleq \begin{bmatrix} DA^T \\ -DA^T \end{bmatrix}$ as the H matrix in the injection-free case. Consider the following restricted security index computation problem

$$\begin{aligned} \underset{\Delta\delta}{\text{minimize}} \quad & \|H_{arc}\Delta\delta\|_0 \\ \text{subject to} \quad & H_{arc}(k,:)\Delta\delta = 1 \end{aligned} \tag{7}$$

There is at least one optimal solution to (7) that is of the form $\beta\Delta\delta_b$, where the entries of $\Delta\delta_b$ is either 0 or 1, and β is chosen such that $\beta H_{arc}(k,:)\Delta\delta_b = 1$. Consequently, optimization problem (7) is also a node partitioning problem. Because of the lack of injection cost in the objective function in (7), the security index upper bounding procedure in Section III-B becomes exact.

Proof: See [15]. ■

Contrary to Proposition 1, Proposition 4 does not require that reactance matrix D to be positive definite. Proposition 4 also has a "protected version" in which some arcs are protected. This version is not explicitly stated in here.

E. Security Index with Partial Measurements

While not being the main focus of this paper, the security index problem with a H matrix containing only part of the measurements can also be approximately addressed by the proposed MIN-CUT based scheme in Section III-B. In step 1 of proposed scheme, the weights of the unmeasured arcs can be set to 0. Hence, the MIN-CUT solving in step 3 would not take into consideration whether the unmeasured arcs are attacked or not. However, the above is only an approximate scheme, since the cost associated with the injection is ignored.

IV. Numerical Experiment

In this section various methods to (optimally or sub-optimally) compute the security indices will be evaluated. To simplify the exposition, the explanation is given in the unprotected case only. However, experiment results in the protected case will also be covered. All computations are performed on a laptop with an Intel Core i5 2.53GHz CPU and 4GB of memory. The power network benchmarks are all obtained from [10]. The methods which are compared in detail include

MILP

Exactly solving a MILP formulation of (4) using well-established solvers such as CPLEX and Gurobi. See [15] for the details of this formulation. The security indices obtained by MILP are used as the reference for accuracy comparison.

Matching Pursuit

Suboptimally solving (4) using matching pursuit. The algorithm `greed_mp.m`, implementing the work from [20], is from Dr. Thomas Blumensath's website.

LASSO

Suboptimally solving (4) using LASSO. This means that the cardinality objective function in (4) is replaced with the vector 1-norm. The problem can be formulated as a convex linear program (see, for example, [12] or [21] for detail). The linear program solver is using CPLEX.

MIN-CUT

Security index upper bounding using the proposed MIN-CUT based scheme in Section III-B.

In order to study the effect of the enumeration in step 4 in the proposed upper bounding scheme, the proposed scheme is carried out with three variants, each requiring more computation time than the previous. These three variants are denoted MIN-CUT-1, MIN-CUT-2, and MIN-CUT-all described as follows. In MIN-CUT-1, step 4 is ignored. In MIN-CUT-2, step 4 is partially executed. Instead of enumerating all optimal partitions of the MIN-CUT problem, only two partitions with the minimum and maximum source sets are considered. See [15] for the details of MIN-CUT-2. Finally, MIN-CUT-all is the proposed scheme, with step 4 fully executed. All MIN-CUT problems in this paper are solved using the MAX-FLOW solver from MatlabBGL [22], which uses the routines from Boost Graph Library [23].

The first test case is the IEEE 14-bus example, which includes 14 nodes and 20 arcs. The security indices of all except the first 20 redundant ones (cf. (2)) are either exactly or approximately computed using the methods described above. Denote α_k as the security index of measurement k (computed using MILP), and $\hat{\alpha}_k$ as the inexact index computed by matching pursuit, LASSO, or the proposed MIN-CUT based schemes. Define the relative error for a security index as $\frac{\hat{\alpha}_k-\alpha_k}{\alpha_k}$. Then define the ***average relative error*** as the average of the relative errors over all security indices, except the last redundant ones. Table I shows the average relative error and the computation time. Table I suggests that the quality of the security index approximation by matching pursuit and LASSO is relatively poor, while the proposed MIN-CUT schemes are very accurate and time-efficient. Since the security index α_k is the minimum number of attacks that need to be carried out in order to compromise measurement k. The erroneously large indices estimated by matching pursuit and LASSO can lead to a false sense of security.

TABLE I

AVERAGE RELATIVE ERROR AND COMPUTATION TIME IN THE IEEE 14 BUS EXAMPLE

Method	Ave. rel. error (%)	Time (sec)
MILP	0	1.147
Matching pursuit	77.52	0.8946
LASSO	35.00	0.5842
MIN-CUT-1	0	0.007879
MIN-CUT-2	0	0.01213
MIN-CUT-all	0	0.02017

The next test case is the IEEE 118-bus example. In this example, 10 different copies of the original IEEE 118-bus network are created by adding protection (see Section III-C) to different sets of nodes. The protection is uniformly placed (in terms of node labels) in the copies. The percentages of protected nodes increase in the copies from 0% to 50%. Fig. 3 shows the average relative errors (as defined in the previous example) due to matching pursuit and LASSO. Matching pursuit spends a total of 54 minutes to compute the indices (probably having a convergence issue; MILP takes about 50 minutes), and LASSO spends about 95 seconds.

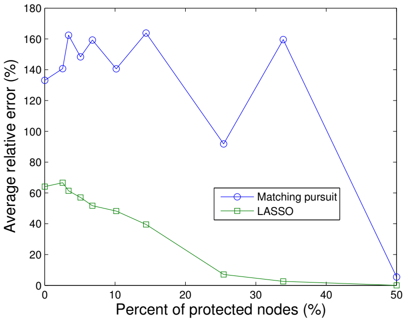

Fig. 3. Average relative errors (%) due to matching pursuit and LASSO. The relatively small errors for the cases with a lot of protection (e.g. 50%) is due to the fact that there are many measurements that cannot be attacked, and those attackable admits very simple attack vectors.

The average relative errors (%) due to MIN-CUT-1 are 1.104, 0.8978, 0.9370, 0.7997, 0.7897, 0.6174, 0.4308, 0.1600, 0.1023, 0 for the 10 cases, respectively. On the other hand, MIN-CUT-2 and MIN-CUT-all do not incur any error. Regarding computation time, the three MIN-CUT schemes take 0.8711 second, 1.287 seconds, and 2.007 seconds respectively (MILP takes about 50 minutes). This example again demonstrates the efficiency and accuracy advantages of the proposed scheme.

The last test case is a 2383-bus example (`case2383wp` from [10]). Out of the 8175 measurements, the last 5279 security indices are meaningful. In this example, none of the measurements are protected. The proposed MIN-CUT schemes, as well as the simple upper bounding scheme from [6] are used to compute all 5279 security indices. Only 14 selected measurements are chosen to compute the true security indices using MILP (due to time limitation). Figure 4 shows the comparison for these 14 selected indices. For all security indices, MIN-CUT-2 and MIN-CUT-all provide

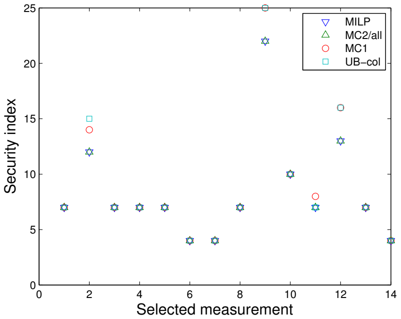

Fig. 4. Security indices (and bounds) for 14 selected measurements in the IEEE 2383-bus example. The MIN-CUT-1 exhibits some error, while the other two MIN-CUT schemes do not incur any. The method from [6] (UB-col) also leads to some error.

the same bounds (and exact at least for the 14 selected cases in Fig. 4). The security index bounds found by the simplest MIN-CUT-1 and the method from [6] are no smaller than those of the other two MIN-CUT schemes, and sometimes strictly larger. The "average relative error" (with respect to MIN-CUT-all scheme) of the MIN-CUT-1 scheme is 1.433%, while the analogous quantity for [6] is 6.889%.

In terms of computation time, MILP takes a total of 1307 seconds to solve the security indices for the 14 measurements (minimum time case is 7.978 seconds and maximum case is 665.8 seconds). On average, MILP will require 5.7 days to compute all 5279 meaningful indices. On the other hand, the MIN-CUT bounding schemes take 12.92 seconds, 19.36 seconds and 31.41 seconds respectively. The scheme from [6] is very efficient, requiring only 0.8543 second. Finally, it is worth noting that CPLEX can stop the algorithm for finding security index as soon as a feasible solution is found. Using this feature of CPLEX, another 14 approximate security indices can be obtained for the selected measurements. The computation takes 129.9 seconds (for 14 indices only). The average relative error for these 14 approximate indices is 31.43%. This example further demonstrates the time-efficiency of the proposed method versus MILP.

V. Conclusion

This paper analyzes the security of electric power network through the security index introduced in [6]. By exploiting its structure, the security index problem in (4) can be reduced to a minimum cost node partitioning problem. While this problem is still combinatorial, its cost structure inspires a highly efficient and accurate approximate algorithm via MIN-CUT relaxation. The proposed approximate algorithm can be modified in several ways, depending on the application. For instance, the situation with encryption protection of buses can be handled (cf. (6)), and the proposed algorithm becomes exact in the special case where the H matrix does not contain any injection measurement (cf. (7)). However, the presented results are by no means complete. For instance, the rigorous investigation of the accuracy of the approximate algorithm is desirable. In addition, the capability to handle injection measurement protection without protecting all incident arcs is highly relevant. The removal of the positive reactance assumption is also worth investigation.

Acknowledgement

The authors would like to thank Andŕe Teixeira for discussions, Yao Liu for sharing the code in [4], and Prof. György Dán for sharing the code in [7].

References

[1] S. Spoonamore and R. Krutz, "Smart Grid and Cyber Challenges: National Security Risks and Concerns of Smart Grid," 2009.

[2] S. M. Amin, "Smart Grid: Opportunities and Challenges Toward a Stronger and Smarter Grid," *MIT Energy Conference Accelerating Change in Global Energy, Cambridge, Massachusetts*, 2009.

[3] *Forbes, Congress Alarmed at Cyber-Vulnerability of Power Grid, available at http://www.forbes.com/2008/05/22/cyberwar-breach-government-tech-security_cx_ag_0521cyber.html*.

[4] Y. Liu, M. Reiter, and P. Ning, "False data injection attacks against state estimation in electric power grids," in *16th ACM Conference on Computer and Communication Security*, New York, NY, USA, 2009, pp. 21–32.

[5] R. Bobba, K. Rogers, Q. Wang, H. Khurana, K. Nahrstedt, and T. Overbye, "Detecting false data injection attacks on dc state estimation," in *the First Workshop on Secure Control Systems, CPSWEEK 2010*, 2010.

[6] H. Sandberg, A. Teixeira and K.H. Johansson, "On security indices for state estimators in power networks," in *First Workshop on Secure Control Systems, CPSWEEK 2010*, 2010.

[7] G. Dan and H. Sandberg, "Stealth attacks and protection schemes for state estimators in power systems," in *IEEE SmartGridComm*, 2010.

[8] A. Teixeira, S. Amin, H. Sandberg, K. H. Johansson, and S. Sastry, "Cyber security analysis of state estimators in electric power systems," in *Proceedings IEEE Conference on Decision and Control*, dec 2010.

[9] A. Abur and A. Expósito, *Power System State Estimation*. Marcel Dekker, Inc, 2004.

[10] R. Zimmerman, C. Murillo-Sánchez, and R. Thomas, "MATPOWER Steady-State Operations, Planning and Analysis Tools for Power Systems Research and Education," *IEEE Transacations on Power Systems*, vol. 26, no. 1, pp. 12–19, 2011.

[11] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Royal. Statist. Soc B*, vol. 58, no. 1, pp. 267–288, 1996.

[12] J. Tsitsiklis and D. Bertsimas, *Introduction to Linear Optimization*. Athena Scientific, 1997.

[13] O. Kosut, L. Jia, R. Thomas, and L. Tong, "Malicious data attacks on smart grid state estimation: Attack strategies and countermeasures," in *IEEE SmartGridComm*, 2010.

[14] A. Giani, E. Bitar, M. McQueen, P. Khargonekar, and K. Poolla, "Smart grid data integrity attacks: Characterizations and countermeasures," in *IEEE SmartGridComm*, 2011, to appear.

[15] K.C. Sou and H. Sandberg and K.H. Johansson, "Electric power network security analysis via minimum cut relaxation," Reglerteknik, Royal Institute of Technology, Tech. Rep., available online from `https://eeweb01.ee.kth.se/upload/publications/reports/2011/IR-EE-RT_2011_089.pdf`.

[16] M. Stoer and F. Wagner, "A simple min-cut algorithm," *J. ACM*, vol. 44, pp. 585–591, July 1997. [Online]. Available: http://doi.acm.org/10.1145/263867.263872

[17] L. Ford and D. Fulkerson, "Maximal flow through a network," *Canadian Journal of Mathematics*, vol. 8, pp. 399–404, 1956.

[18] J.-C. Picard and M. Queyranne, "On the structure of all minimum cuts in a network and applications," in *Combinatorial Optimization II*, ser. Mathematical Programming Studies, 1980, vol. 13, pp. 8–16.

[19] L. Schrage and K. R. Baker, "Dynamic programming solution of sequencing problems with precedence constraints," *Operations Research*, vol. 26, no. 3, pp. pp. 444–449, 1978.

[20] S. Mallat and Z. Zhang, "Matching pursuit with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, pp. 3397–3415, 1993.

[21] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[22] D. Gleich, "Contents matlab bgl v4.0," 2006.

[23] *The boost graph library: user guide and reference manual*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2002.