PReSS Towards a Secure Smart Grid: Protection Recommendations against Smart Spoofing

Charith Wickramaarachchi[1], Rajgopal Kannan[1], Charalampos Chelmis[2], and Viktor K. Prasanna[1]

[1]University of Southern California, Los Angeles, USA
[2]University at Albany - SUNY, NY, USA
[1]{*cwickram,rajgopak,prasanna*}*@usc.edu*
[2]*cchelmis@albany.edu*

***Abstract*—Protecting the integrity of state estimates that inform the physical state of a power transmission network is vital for the safe operation. Existing methods to protect the critical state estimates in smart-grid against data spoofing attacks assume a static set of critical buses. Instead, we propose a generalized optimal protection scheme based on a prize-collecting Steiner tree formulation that captures the criticality of buses and protection cost. We argue that the criticality of buses can change over time, and present a set of optimal schemes for adaptive protection against data spoofing attacks in smart grids.**

Next, we note that such optimal schemes are computationally intractable and propose heuristics with polynomial time complexity. We evaluate the proposed protection schemes using simulations on publicly available transmission network datasets. Simulation results show that the proposed heuristics closely approximate the optimal results while being able to scale for large transmission networks.

I. Introduction

State estimation is a mission critical operation in a smart grid [1]. It is performed in an online manner in modern SCADA/EMS systems using the real-time sensor measurements of transmission network [1]. However, it has been recently shown that state estimation is vulnerable to data spoofing attacks (DSAs)[2]. Specifically, when an attacker has knowledge of the topology of a transmission network, he/she could formulate an attack by spoofing a carefully selected set of sensors. Such attacks are not detectable by the bad data detector (BDD) at SCADA/EMS systems. Invalid state estimates can cause severe socioeconomic impacts [3] [4].

Protection schemes to secure a critical subset of buses have been proposed [4], [5]. These protection schemes are designed with an underlying assumption of a static set of critical buses. It has been shown that attacking buses that have a high impact on a transmission network congestion can have a higher impact on power markets [3]. Congestion in different parts of the transmission network can change dramatically and frequently by the time of day, the day of the week and season [6]. Thus the criticality of buses may change over time. We argue that *protection schemes for a smart-grid should be adaptive* in order to capture these changes. Existing methods to protect the sensor measurements in a transmission network include, placing guards and surveillance instruments at sensor locations [4]. With the changes to the protection strategy, these resources will have to be relocated, activated, or deactivated.

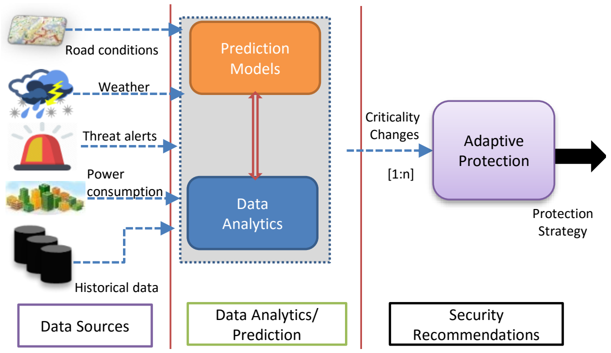

Fig. 1: *PReSS*: Framework for **p**rotection **re**commendations against **s**mart **s**poofing.

Capturing the above-mentioned scenarios, in this paper, we formulate generalized protection schemes based on prize-collecting Steiner tree problem (PCST). In particular, we propose *optimal adaptive protection schemes* that account for the cost of resource relocations, change in the criticality of buses and cost of protecting measurements to provide optimal protection recommendations. We note that proposed optimal adaptive protection schemes are NP-Hard and propose polynomial time heuristics that can scale for real scale smart-grid transmission networks. Using simulations on transmission network datasets, we experimentally evaluate the proposed protection schemes.

Fig. 1 illustrates a high-level framework that utilizes proposed protection schemes. It predicts the criticality changes in buses using data driven models such as [7]. Proposed protection schemes utilize these predictions to provide protection recommendations. The heuristics proposed in this paper are developed as a part of an ongoing project in USC Smart-Grid living lab [1].

II. Background and Related Work

Computing complex voltage phase angles at buses is a critical operation in smart grid state estimation [1]. While the voltage magnitudes can be directly measured at the buses, the phase angles are calculated indirectly using the power flow measurements of transmission lines [1]. These calculations are done at SCADA/EMS system that has a bad data detector

[1] http://energy.usc.edu/major-projects/

(BDD). BDD can identify random errors in the power flow measurements.

In [2] Liu et al, showed that attackers can formulate DSAs utilizing the topology information of the transmission network to bypass BDD undetected, resulting invalid phase angle estimates.

In [8] Bobba et al showed that no undetectable DSA can be formulated if the set of measurements that are protected makes the power system observable. The minimum number of measurements that needs to be protected in order to protect all buses is same as the number of buses as a result. This is expensive for large transmission networks. As a result, a graph based method to protect a given set of critical buses has been proposed in [5]. This method takes a critical set of buses and bus branch graph $G = (V, E)$ of the transmission network as input. It computes a set of measurements M to be protected in order to protect the state estimate associated with the critical set of buses. M is termed as the protection strategy to protect the critical set of buses.

G for a transmission network that consists of flow measurements and direct phase measurements (through phase measurement units (PMUs)) is constructed in which [5]: 1) Each state variable associated with a bus is represented by a vertex $v \in V$; 2) Each flow meter is represented by an edge $e = (u, v)$ where u and v are buses incident to the measured transmission line; 3) A new vertex: *reference vertex* (v_r) is introduced connecting each bus with a phase measurement to the *reference vertex* by adding new edges; 4) Each edge that represents a secured measurement has zero weight and each unsecured edge has a unit weight.

The minimum set of measurements that should be protected in order to protect a critical set of buses is given by the unprotected edges in minimum Steiner tree that connects v_r with vertices representing critical buses [5]. All the buses that are spanned by the resulting Steiner tree will be protected as a result [5].

Existing protection schemes [4], [5] target a static set of critical buses assuming that criticality of buses does not change over time. As described earlier [6], this assumption need not always be true. Our proposed protection schemes jointly optimize the measurement protection cost, the potential loss that can occur due to unprotected critical buses and cost of changing M to capture the change in criticality of buses.

Minimum prize-collecting Steiner tree is a well-known NP-hard problem that appears in the design of utility networks such at fiber optics. Some work has been done on approximate algorithms for minimum prize collecting Steiner tree problem [9], [10]. Our optimal protection schemes are formulated as a variant of prize-collecting Steiner tree problems and existing heuristics for classic prize-collecting Steiner tree can not be directly applied. None of the existing variants of PCST capture time-varying aspects in the objective.

III. Problem Formulation

We observe that the graph based protection scheme described above can be generalized to incorporate the cost of protecting each measurement and criticality of the buses using rooted prize-collecting Steiner tree problem. As mentioned in [3], an attacker can gain financial benefits (loss for utility) by attacking buses. The potential loss that can occur if a bus is attacked denotes its criticality. This is represented as the vertex ”prize” in G. Costs of protecting measurements are represented as edge weights of G. The objective of an optimal protection scheme is to minimize the potential loss that can occur due to unprotected buses and protection cost. Observation 1 summarizes the resulting protection scheme.

Observation 1:

The set of minimum cost measurements to protect in order to minimize the potential loss that can occur due to unprotected buses is given by a minimum rooted prize-collecting Steiner tree (PCST) of G. In which: 1) Each vertex has an associated prize representing its criticality; 2)Each edge has a weight representing the measurement protection cost of associated measurement; 3) Reference vertex v_r is the root.

Next, in order to capture the changes in criticality of buses, we denote it as a time varying value. π_i^t is the criticality of bus represented by vertex i at time t. As a result, potential loss that can occur due to unprotected buses at time t is $\sum_{i \in V} \pi_i^t y_i^t$. The decision variable $y_i^t = 1$ if bus represented by vertex i is protected at time t. Protection cost for the measurement represented by the edge e is denoted by c_e.

Changes in criticality of buses will require changes in M. However, a change in M incur a cost. This is a cost associated with relocating (moving/activating/deactivating) measurement protection resources due to changes in the protection strategy. Let M^t, M^{t-1} denote protection strategies at time t, $t-1$ respectively and E_{M^t}, $E_{M^{t-1}}$ the edges that represent them in G. We denote this by adding a relocation cost r_e for each edge $e \in (E_{M^t} \setminus E_{M^{t-1}}) \cup (E_{M^{t-1}} \setminus E_{M^t})$. As a result, relocation cost for a measurement represented by edge e at time t is $r_e \Delta^t(e)$ where $\Delta^t(e) = \mid x_e^t - x_e^{t-1} \mid$. The decision variable $x_e^t = 1$ if the measurement e is protected at time t.

An optimal adaptive protection scheme takes a set of criticality predictions for buses as input to provide a protection strategy/security recommendation that will minimize measurement protection cost (C), the potential loss that can occur due to unprotected buses (P) and relocation cost (R). We formalize two such optimal adaptive protection schemes below. C, P, R are defined separately for each protection scheme.

Optimal adaptive protection schemes are defined below as variant prize-collecting Steiner tree problems in which $\delta(S)$ is the *cut set* for the set $S \subset V$. Recommended protection strategy is given by the set of measurements represented by the edges in the resulting Steiner trees. Buses represented by vertices in these trees will be protected as a result.

Minimum protection cost tree for local risk predictions (MPT-Local):

MPT-Local protection scheme provide a protection recommendation (M_t) based on criticality predictions for each bus for the next time interval starting at time t. As a result, C, P, R are given by $\sum_{e \in E} c_e x_e^t$, $\sum_{i \in V} \Pi_i^t (1 - y_i^t)$ and $\sum_{e \in E} r_e \Delta^t(e)$

respectively. The ILP formulation for the variant PCST problem that minimize the overall cost is as follows:

$$\begin{aligned}
&\text{min :}\\
&\textstyle\sum_{e\in E} c_e x_e^t + \sum_{i\in V} \pi_i^t(1-y_i^t) + \sum_{e\in E} r_e \Delta^t(e)\\
&\text{subject to:}\\
&\textstyle\sum_{e\in\delta(S)} x_e^t \geq y_i^t, \forall S \subset V \setminus \{v_r\}, S \neq \emptyset, \forall i \in S\\
&x_e^t \in \{0,1\},\ \forall e \in E\\
&y_i^t \in \{0,1\},\ \forall i \in V
\end{aligned} \tag{1}$$

Minimum protection cost trees for a time window of risk predictions (MPT-Window):

MPT-Window protection scheme provide protection recommendations for a time window. It takes criticality predictions for the buses in time window as the input. Protection strategy for each time interval is selected based on these predictions. C, P, R for a time window with n time intervals is given by $\sum_{j\in[0,n),e\in E} c_e x_e^{t+j}$, $\sum_{j\in[0,n),i\in V} \pi_i^{t+j}(1-y_i^{t+j})$ and $\sum_{j\in[0,n),e\in E} r_e\Delta^{t+j}(e)$ respectively for a time window of size n. The ILP formulation that minimize the overall cost for the time window is as follows:

$$\begin{aligned}
&\text{min :}\\
&\textstyle\sum_{j\in[0,n),e\in E} c_e x_e^{t+j} + \sum_{j\in[0,n),i\in V} \pi_i^{t+j}(1-y_i^{t+j}) +\\
&\textstyle\sum_{j\in[1,n),e\in E} r_e\Delta^{t+j}(e)\\
&\text{subject to:}\\
&\forall j \in [1,n) :\\
&\textstyle\sum_{e\in\delta(S)} x_e^{t+j} \geq y_i^{t+j}, \forall S \subset V \setminus \{v_r\}, S \neq \emptyset, \forall i \in S\\
&x_e^{t+j} \in \{0,1\}\ \forall e \in E\\
&y_i^{t+j} \in \{0,1\}\ \forall i \in V
\end{aligned} \tag{2}$$

Above stated objective functions are LP objective functions since we are minimizing a sum of absolute values.

We note that MPT-Local and MPT-Window are NP-Hard by considering a special case in which there is no relocation cost. With no relocation cost, MPT-Local and MPT-Window become instances of standard PCST problem making them at least as hard as PCST. PCST is a well known NP-Hard problem.

IV. Graph Heuristics

Due to intractable computation complexity, above-mentioned optimal protection schemes do not scale for real world transmission networks. Attackers can exploit the long computation time windows to attack unprotected critical buses. In order to address this limitation, we propose heuristic algorithms with polynomial time complexity that can scale for real-scale transmission networks.

The proposed heuristics for MPT-Local and MPT-Window uses the Algorithm 1 as the base algorithm for computing rooted prize-collecting Steiner tree. Algorithm 1 first creates a spanning tree in a greedy manner and then prune the edges that improve the overall objective.

The heuristic for MPT-Local computes the Steiner tree for each time interval using Algorithm 1. It uses graph G' as the input to Algorithm 1. G' is created from G by recomputing the edge weights. The edge weights are recomputed by adding a penalty to the measurements that are not protected in the previous time interval based on equations 3 and 4.

$$c'_e = \begin{cases} c_e, & \text{if } e \in E_{M_{t-1}}. \\ r_e + m_e + c_e, & \text{otherwise.} \end{cases} \tag{3}$$

$$m_e = max\{r_{e'} | \,\forall e' \in E_{M_{t-1}} \setminus \{e\}\} \tag{4}$$

Where c'_e denotes new weight of edge e in G'.

Similarly, the heuristic for MPT-Window computes a Steiner tree for each time interval in the time window $(t, t+n)$ using the Algorithm 1. G' is created from G for each time interval in the window by recomputing edge weights and vertex criticality values. The edge weights are computed based on the equation 3 and 4 and the vertex criticality values are computed using the equation 5.

$$\pi'^{t+i}_v = \frac{\sum_{j=i}^{j=n} \frac{\pi_v^{t+j}}{j-i+1}}{\sum_{j=1}^{j=n-i+1}(j^{-1})},\ \forall v \in V \setminus \{v_r\} \tag{5}$$

Algorithm 1 Heuristic Algorithm for Price Collecting Steiner Tree (H-PCST)

```
1: procedure H-PCST($G$)
2:     $T_V \leftarrow \{v_r\}$, $T_E \leftarrow \{\}$
3:     $cost[v_r] \leftarrow 0$
4:     for all $v \in V \setminus \{v_r\}$ do
5:         $cost[v] \leftarrow \infty$
6:     while $T_V$ not contain all $v \in V_{D_{t+1}}$ do
7:         find $e \leftarrow \max_{e\in E}\{\pi_j - c_e \mid e = (i,j) \in \delta(T_V)\}$
8:         $T_V \leftarrow T_V \cup \{j\}$
9:         $T_E = T_E \cup \{e\}$
10:        for all $e = (j,k) \in E$ do
11:                ▷ update edge costs $\pi_k - c_e$
12:            if $cost[k] < \pi_k - c_e$ then
13:                $cost[k] \leftarrow \pi_k - c_e$
14:                    ▷ call Increase-Key on $(k, cost[k])$
15:                    ▷ set predecessor of $k$ as $j$
16:    $Q \leftarrow$ CREATEMINHEAP($T_V \setminus \{v_r\}$)
17:            ▷ out-degree as the key
18:    while $Q$ not empty do
19:        $j \leftarrow$ GETMIN(Q)
20:        $val \leftarrow$ GETKEY(j)
21:        if val = INF then
22:            break
23:        if ISLEAF($v$) then
24:            $e = (i,j)$            ▷ $i$ is the parent of $j$
25:            if $\pi_j < c_e$ then
26:                $T_V \leftarrow T_V \setminus \{j\}$
27:                $T_E = T_E \setminus \{e\}$
28:                REMOVE(Q,v)
29:            else
30:                INCREASEKEY(Q,v,INF)
31:        else
32:            break
33:    return $(T_V, T_E)$
```

Complexity Analysis:

Assuming an adjacency list is used to store the graph and a min-heap is used to find minimum cost edges in $\delta(T_V)$ (Line 7 of Algorithm 1), we can observe that line 11-15 of the algorithm will be executed $O(|E| + |V|)$ times. In each step, updates to the min-heap in order to update the new edge costs require $O(\log |V|)$ operations. Therefore the total number of

Dataset	# of Buses	# of Lines
IEEE 9	9	9
IEEE 14	14	20
IEEE 57	57	80
IEEE 118	118	186
IEEE 300	300	409
EU 1494	1494	2322

TABLE I: Sizes of different power system test cases.

operations required to create a spanning tree from line 11-15 of the algorithm is $O(|E|\log|V|)$.

Next, line 16 of the algorithm will require $O(|V|\log|V|)$ operations to build a min heap. Since $|T_E| = |V| - 1$, line 18-32 will require $O(|V|\log|V|)$ operations for the leaf pruning process.

Therefore, the total number of operations Algorithm 1 require is $O(|E|\log|V|)$. Algorithm 1 is executed for each time interval in a time window. In each time interval edge weights and vertex criticality values are recalculated using equations 3 and 5 respectively. Assuming $|TW|$ is the number of time intervals in the time window, it will require $O(|E||V||TW|)$ and $O(|V||TW|^2)$ operations for these calculations based on 3 and 5 respectively. As a result, heuristics for MPT-Local and MPT-Window will require $O(|E||V||TW| + |TW||E|\log|V|)$ and $O(|V||TW|^2 + |E||V||TW| + |TW||E|\log|V|)$ operations respectively for a time window with $|TW|$ intervals.

V. Simulation Results and Discussion

In this section, we present our simulation results of above mentioned protection schemes. ILP formulations were implemented using Gurobi solver [2] and heuristic algorithms were implemented using Java 1.8. All experiments were conducted on an Intel Core i5 3.2Ghz machine with 16GB memory.

We used a set of publicly available IEEE power system[3] and European transmission network [4] test cases in our simulations. Details of the six datasets we used in simulations are summarized in Table I. We generated bus branch graphs following steps mentioned in the section II. We considered three configurations for each transmission network in which 25%, 50% and 75% buses have direct phase measurements (PMUs). We conducted the experiments on all possible configurations. Only a few representative results are presented due to space restrictions.

Evaluation:

We first compare the performance of optimal protection schemes based on the protection cost for different time windows. This is achieved by comparing the protection cost given by optimal protection schemes in 1 and 2 on IEEE 9 bus test case for a time window of size 3 and 6. The reason we had to use this smaller test case was due to the intractable complexity of optimal formulations. It took 3 days to compute an optimal protection strategy based on 2 for time window size 9 on a single IEEE 9 bus test case where the proposed heuristics could compute results under a minute on all transmission network datasets (It took less than 1 millisecond to compute the protection strategy for for 9-bus test case for MPT-Local and MPT-Window for window size 9 on average. Computation on EU 1494 test case for window size 1152 took 47 and 54 seconds for MPT-Local and MPT-Window respectively). In order to report statistically significant results, we generated 1000 different test cases by assigning different bus criticality values and measurement protection costs. Values were generated from a uniform random distribution between 0 and 100. We report the distribution of relative increase in protection cost when MPT-Local is used compared to MPT-Window (R_C) for different time windows given by:

%PMU	25%		50%		75%	
	3	6	3	6	3	6
Mean	5.72	10.73	7.65	13.41	8.86	17.04
SD	5.63	8.64	7.33	10.58	6.88	11.24
SKW	1.15	0.93	1.28	1.17	0.90	0.91

TABLE II: R_C for time windows 3 and 6. SD=standard deviation, SKW= skewness.

Method	%PMU	25%		50%		75%	
	Window size	3	6	3	6	3	6
MPT-Local	Mean	1.09	1.08	1.03	1.05	1.15	1.14
	SD	0.11	0.11	0.08	0.09	0.13	0.15
	SKW	1.14	0.90	0.94	0.62	0.80	1.11
MPT-Window	Mean	1.11	1.15	1.09	1.17	1.18	1.24
	SD	0.09	0.12	0.07	0.12	0.11	0.14
	SKW	1.23	1.39	1.33	1.25	0.83	1.10

TABLE III: Approximation ratios for simulations on IEEE-9 bus test case with PMUs at 25%, 50% and 75% of the buses. SD=standard deviation, SKW= skewness.

$$R_C = \frac{(\Gamma_{MPT-Local} - \Gamma_{MPT-Window})}{\Gamma_{MPT-Window}} \tag{6}$$

$\Gamma_{MPT-Local}$ and $\Gamma_{MPT-Window}$ are the overall protection cost for a given time window for MPT-Local and MPT-Window respectively. $\Gamma_{MPT-Local}$ for a time window is the sum of the overall protection cost given by MPT-Local for each time interval.

As shown in Table II we could consistently observe a rapid increase in protection cost for MPT-Local compared to MPT-Window with the increasing time window size.

Next, we evaluated the performance of the heuristics by comparing the protection cost given by heuristics to the optimal protection costs of MPT-Local and MPT-Window. We used the same 1000 test cases generated from the IEEE 9 bus network for these evaluations. The distributions of approximation ratios for different test cases are reported in Table III. The approximation ratio is the ratio between the protection cost given by the heuristics and the optimal values. Results show that the proposed heuristics approximate the optimal results very closely in many cases.

Next, we compared the performance of heuristics for MPT-Local and MPT-Window on large transmission networks. Each experiment with the same configuration (Number of PMUs, transmission network and time window size) was conducted 50 times on different test cases by assigning different bus criticality values and measurement protection costs. Each reported data point represent a mean value of the results.

In order to study the impact of the size of the time window on the performance of the heuristics, we compared the performance on increasing time window sizes. In each time interval, criticality of buses was assigned from a uniform random distribution between 0 and 100. As shown in Fig.2 heuristic

[2] http://www.gurobi.com/

[3] http://amfarid.scripts.mit.edu/Datasets/SPG-Data/index.php

[4] http://wiki.openmod-initiative.org/wiki/Transmission_network_datasets

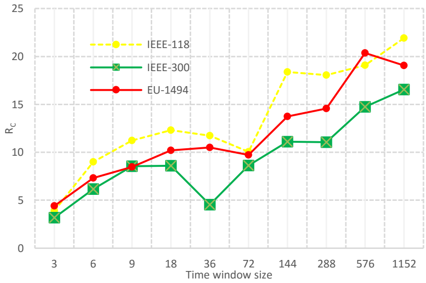

Fig. 2: R_C with increasing time window size.

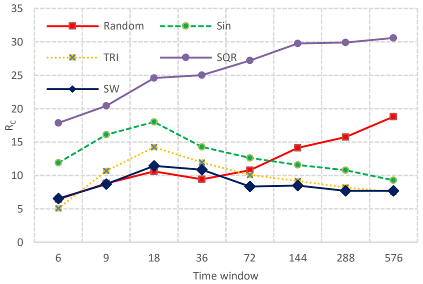

Fig. 3: R_C for various bus criticality variations on EU-1494. Sin = sin wave, SQR = square wave, TRI= triangle wave, SW= sawtooth wave.

for MPT-Window performed better compared to MPT-Local with increasing time windows size.

In order to further understand the impact of changes in the criticality of buses to this behavior, We conducted experiments in which criticality of buses oscillate between 0 and 100 based on different wave functions. We used sin [11], $square$ [12], $triangle$[13] and $sawtooth$[14] wave functions for this evaluation. We assigned wavelength (λ) of the waves to be same as the time window size. Each bus was assigned an initial value randomly between 0 and 100 (initial phase shift). Fig. 3 shows the results for EU-1494 transmission network dataset with PMUs at 25% of the buses. While heuristic for MPT-Window outperformed MPT-Local in all cases, we observed that the performance of MPT-Local improved with increasing time window size when criticality of buses change based on $sine$, $triangle$ and $sawtooth$ wave functions.

This is due to the fact that λ is set to the time window size. With increasing λ, the changes in criticality were more gradual on these waves, this made MPT-Local perform better compared to the cases with sharp criticality changes.

Next, to understand the impact of transmission network size to the performance of heuristics we evaluated the heuristics performance on transmission networks with increasing size. As shown in Fig. 4 R_C was not impacted significantly by the transmission network size.

VI. Conclusion

In this paper, we proposed two optimal adaptive protection schemes, MPT-Local and MPT-Window against smart grid DSAs. We noted that the proposed optimal protection schemes are NP-Hard and proposed approximate solutions with polynomial time complexity.

Our evaluations showed that, given a set of predictions for a time window on criticality changes of buses, MPT-Window always provided the most cost effective protection strategies in terms of overall cost for the time window. We observed that MPT-Local performs best when the criticality of buses change gradually over time. Our heuristics scaled for large transmission networks and closely approximated the optimal solutions. The worst-case approximation ratio observed was ~ 1.25.

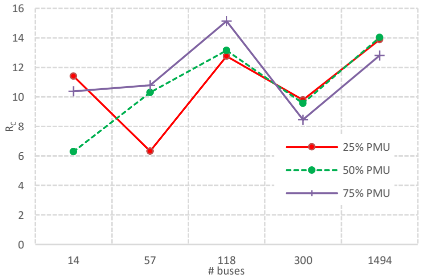

Fig. 4: R_C with increasing number of buses.

In scenarios where transmission network spans a large geographical area or the criticality of the buses change frequently, it may become hard to relocate the measurement protection resources such as guards. The relocation cost values for such relocations can be increased in order to capture such difficult relocations. Cumulative criticality values (ex: sum, average, etc.,) over a period of time can be assigned to buses in order to reduce the frequency of changes.

References

[1] A. Abur and A. G. Exposito, *Power system state estimation: theory and implementation*. CRC press, 2004.

[2] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Transactions on Information and System Security (TISSEC)*, vol. 14, no. 1, p. 13, 2011.

[3] M. Esmalifalak, Z. Han, and L. Song, "Effect of stealthy bad data injection on network congestion in market based power system," in *2012 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2012, pp. 2468–2472.

[4] S. Bi and Y. J. Zhang, "Graphical methods for defense against false-data injection attacks on power system state estimation," *Smart Grid, IEEE Transactions on*, vol. 5, no. 3, pp. 1216–1227, 2014.

[5] C. Wickramaarachchi, S. R. Kuppannagari, R. Kannan, and V. K. Prasanna, "Improved protection scheme for data attack on strategic buses in the smart grid," in *4th IEEE Conference on Technologies for Sustainability (SusTech)*. IEEE, 2016, pp. 96–101.

[6] "The national electric transmission congestion study," September 2015.

[7] M. Misyrlis, C. Chelmis, R. Kannan, and V. K. Prasanna, "Sparse causal temporal modeling to inform power system defense," *Procedia Computer Science*, vol. 62, 2016.

[8] R. B. Bobba, K. M. Rogers, Q. Wang, H. Khurana, K. Nahrstedt, and T. J. Overbye, "Detecting false data injection attacks on dc state estimation," in *the First Workshop on Secure Control Systems10*, 2010, pp. 1–9.

[9] A. Archer, M. Bateni, M. Hajiaghayi, and H. Karloff, "Improved approximation algorithms for prize-collecting steiner tree and tsp," *SIAM Journal on Computing*, vol. 40, no. 2, pp. 309–332, 2011.

[10] M. Akhmedov, I. Kwee, and R. Montemanni, "A fast heuristic for the prize-collecting steiner tree problem," *Lecture Notes in Management Science*, vol. 6, pp. 207–216, 2014.

[11] "Sin wave function," http://mathworld.wolfram.com/Sine.html.

[12] "Square wave function," http://mathworld.wolfram.com/SquareWave.html.

[13] "Triangle wave function," http://mathworld.wolfram.com/TriangleWave.html.

[14] "Sawtooth wave function," http://mathworld.wolfram.com/SawtoothWave.html.