

DIGITALLY ANNEALED SOLUTION FOR THE VERTEX COVER PROBLEM WITH APPLICATION IN CYBER SECURITY

Mohammad Javad-Kalbasi*, Keivan Dabiri†, Shahrokh Valaee† and Ali Sheikholeslami†

*Isfahan University of Technology, m.javadkalbasi@ec.iut.ac.ir

†University of Toronto, {keivan, valaee, ali}@ece.utoronto.ca

Abstract—Cyber attacks on the power systems can mislead the control center to produce incorrect state and topology estimate. State and topology attacks can have harmful impacts on the operation of a power system. The problem of placing secure phasor measurement units (PMUs) to detect these attacks has been studied in the literature. Specifically, it has been shown that placing secure PMUs to disable undetectable state and topology attacks can enhance the security of the system against cyber attacks. Placing secure PMUs is indeed the minimum vertex cover problem. Since the cost of deploying PMUs is high, it is important to place the secure PMUs efficiently in order to maximize the ability of detecting cyber attacks while reducing the costs. In this paper, we use Digital Annealer to solve the vertex cover problem. Digital Annealer is a hardware architecture for solving combinatorial optimization problems. We have performed numerous numerical experiments and noticed that our approach has an enhanced level of optimality compared to other well-known alternatives in the literature.

Index Terms—Power system cyber security, State and topology attacks, Phasor measurement unit, Vertex cover problem, Digital Annealer

I. INTRODUCTION

Smart grid operations are based on communications among participating entities. The reliance on communications increases the potential of cyber attacks. *State and topology attacks* are two important kinds of cyber attacks in which an adversary wants to mislead the control center by changing part of meter and network data [1], [2]. The ability to perturb state or topology estimates enables the adversary to interfere real-time grid operations. In [3]- [5], it was shown that the state and topology attacks have significant effect on real-time pricing.

Modern power systems use bad data detectors to detect inconsistency between network data and meter data. For a successful attack, the adversary needs to modify the network and meter data in order to make them consistent target state and topology. Undetectable state and topology attacks might be feasible when the adversary has the power to modify a large amount of data [1], [2]. Secure phasor measurement units (PMUs) can be used to detect the state and topology attacks on the power grid. Since additional security measures are implemented on the PMUs, PMU data are not subject to corruption by man-in-the-middle-attacks [6]. Since the cost of deploying PMUs is high, it is important to place the secure

PMUs efficiently in order to maximize the ability of detecting cyber attacks while reducing the costs.

The first feasible cyber attack on power system state estimation (*state attack*) was introduced in [1]. In [1], it was shown that an adversary can perturb the state estimate without being detected by modifying the meter data. In [7], [8], a connection was proposed between the feasibility of undetectable state attacks and system observability. Specifically, the placement of secure PMUs to defend against the state attacks was considered in [9], [10]. A kind of cyber attack perturbing the topology estimate (*topology attack*) was studied in [2], [5]. In [2], a countermeasure was proposed to make the undetectable topology attacks infeasible.

The protection of a grid against both state and topology attacks was first studied in [11]. The authors presented a necessary and sufficient graph-theoretical condition for the existence of an undetectable state or topology attack. Especially, it was shown that no undetectable attack exists if and only if the buses with secure PMUs form a vertex cover of the grid topology. In other words, finding an optimal countermeasure is equivalent to finding a minimum vertex cover of the grid topology.

Consider an undirected graph $G = (V, E)$. A vertex cover of G is a set of vertices such that each edge of the graph is incident to at least one vertex of the set. Finding a minimum vertex cover is a typical example of an NP-hard optimization problem. The NP completeness of minimum vertex cover problem was proved in [12]. There are well known algorithms such as Maximum Degree Greedy (MDG) and 2-approximation algorithm for finding an approximate solution for the problem [13], [14]. Similar approximation algorithms have been developed with an approximation ratio of $2 - \frac{\ln(\ln|V|)}{2\ln|V|}$ in [15], [16].

In [17], a heuristic algorithm (VSA) is proposed for the minimum vertex cover problem, which is based on finding a vertex $i \in V$ with the maximum support value $s(i)$ that is defined as $s(i) \triangleq \sum_{j:\{i,j\} \in E} deg(j)$. This algorithm provides near optimum solutions for most of the well known graphs. Also, two modifications of this algorithm (MVSA) and (AVSA) which provide better or equal results have been presented in [18], [19], respectively.

A greedy algorithm (NOVCA-I) based on the fact that the vertices incident to minimum degree nodes are candidate for the minimum vertex cover with high probability, was presented in [20]. Moreover, a stochastic local search algorithm (COVER) for the vertex cover problem was proposed in [21].

In [22], the genetic algorithm was used to solve the vertex cover problem. Recently, the problem of vertex cover has been solved using Ising model on a neuromorphic processor [23]. Although time and space efficiency decreases by a constant factor, the size of vertex cover obtained by this processor is close to the 2-approximation algorithm.

In this paper, we introduce a digital annealing based method to place secure PMUs to alleviate cyber attacks on power systems. Towards this end, we derive an Ising model formulation for the vertex cover problem in the system topology and minimize its associated Ising energy using Digital Annealer. To evaluate the performance of Digital Annealer, we compare the quality of solution and execution time of Digital Annealer with well-known NOVCA-I algorithm using 67 benchmark graphs and 6400 randomly generated graphs.

II. PROBLEM STATEMENT

The control center obtains the meter data $Z \in \mathbb{R}^m$ and network data $S \in \{0, 1\}^l$ from the measurement devices in the grid. The meter data consists of bus injection, line flow and PMU measurements, and the network data consists of l breaker status measurements. Each $S \in \{0, 1\}^l$ states a system topology which is represented by an undirected graph (ν, ε) . ν is the set of buses, and ε is the set of connected transmission lines. Formally, for every transmission line between the buses i and j , $\{i, j\} \in \varepsilon$ if and only if the line is connected. The set of all lines (both connected and disconnected) is denoted by ε_0 .

For the meter data, we have

$$Z = HV$$

where $V \in \mathbb{R}^n$ is the unknown state vector consisting of voltage phases of the buses, and $H \in \mathbb{R}^{m \times n}$ is the measurement matrix which depends on the system topology (ν, ε) .

The adversary modifies the remote terminal data (Z, S) and sends the modified version (\hat{Z}, \hat{S}) to the control center. The adversary may modify all entries of Z and S except those coming from secure PMUs.

The following theorem states a necessary and sufficient graph-theoretical condition for feasibility of undetectable state and topology attacks.

Theorem 1: [11] Placing the minimum number of secure PMUs to disable undetectable state and topology attacks can be achieved by placing secure PMUs on the minimum vertex cover of (ν, ε_0) .

III. PROPOSED HIGH SPEED SOLUTION

In this section, we propose a digitally annealed high speed solution to solve the vertex cover problem in the grid topology (ν, ε_0) . First, we introduce Digital Annealer in more details.

A. Digital Annealer

Digital Annealer [24] is a massively parallel hardware architecture for solving combinatorial optimization problems. This architecture performs a Markov Chain Monte Carlo (MCMC) search to minimize the Ising energy

$$E(x_1, \dots, x_N) \triangleq - \sum_{\{i, j\}} w_{ij} x_i x_j - \sum_i b_i x_i + c \quad (1)$$

where $x_i \in \{0, 1\}$ is a binary variable, N is the total number of bits, w_{ij} is the connection weight between the i 'th and j 'th bits, b_i is the bias term and c is a constant value.

The hardware implements 1024 bits with full connectivity among all bits, with 16-bit weights. Higher weight resolution will provide a higher accuracy in the search. Furthermore, full connectivity enables us to deal with a wider range of problems. In the conventional MCMC algorithm, at each iteration a new state is generated from the current state $X = (x_1, x_2, \dots, x_N)$ by flipping one of the N bits, where $(x_i^{new} = 1 - x_i^{old})$. First, one of the N bits is picked randomly, and a trial is performed by calculating the energy change, and evaluating the acceptance probability of that flip. However, in Digital Annealer at each iteration, a trial is performed on all of the bits (up to 1024) in parallel and subsequently, one bit is picked for flipping. Therefore, significant speedup is achieved in MCMC search. This process is described in detail as follows:

- 1) The energy change is calculated for all bits as

$$\Delta E_i(X) = -(1 - 2x_i) \left(\sum_j w_{ij} x_j + b_i \right).$$

- 2) The acceptance probability

$$P(X^{old} \rightarrow X^{new}) = \min\{1, e^{-\beta \Delta E_i(X)}\}$$

is calculated for each bit using Metropolis criterion and is accepted if it is greater than a random number (generated uniformly between 0 and 1).

- 3) One of the accepted bits in step 2 is flipped, and the associated properties are updated.

Digital Annealer finishes the search after certain number of MCMC iterations which must be specified prior to the execution. During the search, Digital Annealer keeps track of the best state found so far. After it reaches the maximum number of iterations, it outputs the minimum energy (most optimal solution) and the state vector associated with that energy found during the entire MCMC search. Here, the execution time only depends on the maximum iteration number not the iteration number in which the minimum is reached.

B. Vertex Cover as Ising Model

In [25], it has been shown that there are Ising formulations for all of Karp's 21 NP-complete problems. In this section, we focus on finding the Ising model representation of the vertex cover problem. First, the problem of interest is reformed as a constrained integer program and then we use the penalty method to convert it to an unconstrained minimization problem.

Consider the grid topology (ν, ε_0) . By assigning the binary variable $x_i \in \{0, 1\}$ to the bus $i \in \nu$, the minimum vertex cover problem can be formulated as the following *integer program*

$$\begin{cases} \min \sum_{i=1}^{|\nu|} x_i \\ x_i \in \{0, 1\}, & i = 1, \dots, |\nu| \\ (1 - x_i)(1 - x_j) = 0, & \text{for every } \{i, j\} \in \varepsilon_0. \end{cases}$$

Using penalty method, this problem can alternatively be solved by minimizing the following cost function

$$E(x_1, \dots, x_{|\nu|}) = \sum_{i=1}^{|\nu|} x_i + \sum_{\{i,j\} \in \varepsilon_0} \lambda(1 - x_i)(1 - x_j)$$

in which λ ($\lambda \gg 1$) is the penalty coefficient¹. It is not hard to see that

$$\begin{aligned} E(x_1, \dots, x_{|\nu|}) &= \lambda |\varepsilon_0| - \sum_{\{i,j\} \in \varepsilon_0} \lambda(x_i + x_j) \\ &+ \sum_{i=1}^{|\nu|} x_i + \sum_{\{i,j\} \in \varepsilon_0} \lambda x_i x_j \\ &\stackrel{(a)}{=} - \sum_{\{i,j\} \in \varepsilon_0} -\lambda x_i x_j \\ &- \sum_{i=1}^{|\nu|} (\lambda \deg(i) - 1)x_i + \lambda |\varepsilon_0| \end{aligned} \quad (2)$$

where (a) follows from the fact that

$$\sum_{\{i,j\} \in \varepsilon_0} (x_i + x_j) = \sum_{i=1}^{|\nu|} \deg(i)x_i.$$

As it can be seen, the derived cost function in (2) is in the form of Ising energy (1) where $N = |\nu|$, $w_{ij} = -\lambda$ for each edge $\{i, j\} \in \varepsilon_0$, $b_i = \lambda \deg(i) - 1$ and $c = \lambda |\varepsilon_0|$.

IV. NUMERICAL EXPERIMENTS

In this section, we investigate the performance of Digital Annealer for solving the vertex cover problem. We only report the comparison results between Digital Annealer and NOVCA-I due to space limitation. All the procedures of NOVCA-I have been coded in Python language. The experiments were carried out on an Intel(R) Xeon(R) E5-1650 v4 @ 3.60GHz CPU. We evaluated the performance of Digital Annealer using 67 benchmark graphs and 6400 randomly generated graphs.

A. Results for DIMACS Benchmark Graphs

We tested the performance of Digital Annealer using DIMACS [26] benchmark graphs. For DIMACS benchmark graphs, the optimal solutions are known. For benchmarking purposes Digital Annealer is stopped when it reaches the energy associated with that solution. The resulting execution

¹ $\lambda \gg 1$ is required to never violate the constraints.

TABLE I: Results for DIMACS Benchmark Graphs

| instance | $ \nu $ | OPT | DA | T_1 (ms) | NOV | T_2 (ms) |
|----------------|---------|-----|-----|------------|-----|------------|
| brock200-1 | 200 | 179 | 179 | 179 | 181 | 10 |
| brock200-2 | 200 | 188 | 188 | 316 | 191 | 3 |
| brock200-3 | 200 | 185 | 185 | 660 | 187 | 5 |
| brock200-4 | 200 | 183 | 183 | 706 | 186 | 6 |
| brock400-1 | 400 | 373 | 373 | 1159 | 378 | 16 |
| brock400-2 | 400 | 371 | 371 | 935 | 378 | 17 |
| brock400-3 | 400 | 369 | 369 | 401 | 377 | 17 |
| brock400-4 | 400 | 367 | 367 | 498 | 378 | 16 |
| brock800-1 | 800 | 777 | 777 | 3845 | 782 | 30 |
| brock800-2 | 800 | 776 | 776 | 5092 | 783 | 28 |
| brock800-3 | 800 | 775 | 775 | 952 | 783 | 30 |
| brock800-4 | 800 | 774 | 774 | 4198 | 782 | 31 |
| C125.9 | 125 | 91 | 91 | 18 | 92 | 12 |
| C250.9 | 250 | 206 | 206 | 96 | 211 | 24 |
| C500.9 | 500 | 443 | 443 | 463 | 449 | 54 |
| C1000.9 | 1000 | 932 | 932 | 498 | 942 | 128 |
| c-fat200-1 | 200 | 188 | 188 | 3 | 188 | 2 |
| c-fat200-2 | 200 | 176 | 176 | 490 | 176 | 4 |
| c-fat200-5 | 200 | 142 | 142 | 284 | 142 | 18 |
| c-fat500-1 | 500 | 486 | 486 | 3 | 486 | 5 |
| c-fat500-2 | 500 | 474 | 474 | 10 | 474 | 7 |
| c-fat500-5 | 500 | 436 | 436 | 186 | 436 | 23 |
| c-fat500-10 | 500 | 374 | 374 | 332 | 374 | 67 |
| gen200-p0.9-44 | 200 | 156 | 156 | 258 | 163 | 20 |
| gen200-p0.9-55 | 200 | 145 | 145 | 76 | 162 | 20 |
| gen400-p0.9-55 | 400 | 345 | 345 | 427 | 351 | 41 |
| gen400-p0.9-65 | 400 | 335 | 335 | 50 | 355 | 42 |
| gen400-p0.9-75 | 400 | 325 | 325 | 28 | 353 | 48 |
| hamming6-2 | 64 | 32 | 32 | 6 | 32 | 8 |
| hamming6-4 | 64 | 60 | 60 | 4 | 60 | 1 |
| hamming8-2 | 256 | 128 | 128 | 60 | 128 | 84 |
| hamming8-4 | 256 | 240 | 240 | 4 | 240 | 6 |
| hamming10-2 | 1024 | 512 | 512 | 129 | 512 | 1832 |
| hamming10-4 | 1024 | 984 | 984 | 216 | 988 | 75 |

time will represent the upper-bound of Digital Annealers performance on these graphs. The numerical results are given in Tables I and II. DA and NOV represent the size of cover obtained by Digital Annealer and NOVCA-I, respectively. Moreover, T_1 and T_2 state the execution time of Digital Annealer and NOVCA-I, respectively. We noticed that Digital Annealer can find the optimal solution for all of the tested benchmark graphs after tuning the annealing parameters.

B. Results for Random Graphs

We also studied the performance of Digital Annealer on Erdős Rényi random graphs [27] with $|\nu| = 64k$ ($k = 1, 2, \dots, 16$) vertices for which the probability of an edge between any pair of nodes (p) was set to four different values 0.1, 0.2, 0.3 and 0.4. We generated 100 random instances for each pair of $(|\nu|, p)$ and solved the vertex cover problem using Digital Annealer and NOVCA-I algorithm. The average size of vertex cover obtained by Digital Annealer is given in Table III for different values of $|\nu|$ and p . Fig. 1 shows the average improvement (reduction in the size of cover set) over NOVCA-I. As it can be seen, Digital Annealer finds better solutions in comparison to NOVCA-I in all sizes and probabilities. This figure indicates that Digital Annealer can yield better solutions when p decreases. Also, we compared the execution time of Digital Annealer with NOVCA-I in Fig. 2.

TABLE II: Results for DIMACS Benchmark Graphs

| instance | $ \nu $ | OPT | DA | T_1 (ms) | NOV | T_2 (ms) |
|---------------|---------|-----|-----|------------|-----|------------|
| johnson8-2-4 | 28 | 24 | 24 | 4 | 24 | 1 |
| johnson8-4-4 | 70 | 56 | 56 | 4 | 56 | 3 |
| johnson16-2-4 | 120 | 112 | 112 | 4 | 112 | 2 |
| johnson32-2-4 | 496 | 480 | 480 | 4 | 480 | 21 |
| keller4 | 171 | 160 | 160 | 6 | 164 | 4 |
| keller5 | 776 | 749 | 749 | 120 | 761 | 37 |
| p-hat300-1 | 300 | 292 | 292 | 13 | 293 | 4 |
| p-hat300-2 | 300 | 275 | 275 | 13 | 275 | 12 |
| p-hat300-3 | 300 | 264 | 264 | 71 | 266 | 19 |
| p-hat500-1 | 500 | 491 | 491 | 7 | 492 | 7 |
| p-hat500-2 | 500 | 464 | 464 | 53 | 466 | 25 |
| p-hat500-3 | 500 | 450 | 450 | 257 | 454 | 40 |
| p-hat700-1 | 700 | 689 | 689 | 50 | 692 | 9 |
| p-hat700-2 | 700 | 656 | 656 | 20 | 657 | 44 |
| p-hat700-3 | 700 | 638 | 638 | 151 | 641 | 76 |
| p-hat1000-1 | 1000 | 900 | 900 | 9 | 991 | 17 |
| p-hat1000-2 | 1000 | 954 | 954 | 44 | 955 | 62 |
| p-hat1000-3 | 1000 | 932 | 932 | 201 | 939 | 110 |
| san200-0.7.1 | 200 | 170 | 170 | 256 | 183 | 8 |
| san200-0.7.2 | 200 | 182 | 182 | 378 | 186 | 8 |
| san200-0.9.1 | 200 | 130 | 130 | 229 | 153 | 31 |
| san200-0.9.2 | 200 | 140 | 140 | 310 | 160 | 23 |
| san200-0.9.3 | 200 | 156 | 156 | 356 | 167 | 17 |
| san400-0.5.1 | 400 | 387 | 387 | 476 | 391 | 8 |
| san400-0.7.1 | 400 | 360 | 360 | 439 | 379 | 19 |
| san400-0.7.2 | 400 | 370 | 370 | 661 | 383 | 15 |
| san400-0.7.3 | 400 | 378 | 378 | 326 | 386 | 15 |
| san400-0.9.1 | 400 | 300 | 300 | 434 | 318 | 56 |
| san1000 | 1000 | 900 | 900 | 769 | 991 | 27 |
| sanr200-0.7 | 200 | 182 | 182 | 11 | 185 | 7 |
| sanr200-0.9 | 200 | 158 | 158 | 351 | 159 | 20 |
| sanr400-0.5 | 400 | 387 | 387 | 70 | 388 | 8 |
| sanr400-0.7 | 400 | 379 | 379 | 80 | 382 | 12 |

TABLE III: Average cover size obtained by Digital Annealer

| $ \nu $ | $p = 0.1$ | $p = 0.2$ | $p = 0.3$ | $p = 0.4$ |
|---------|-----------|-----------|-----------|-----------|
| 64 | 39.6 | 47.5 | 51.3 | 54.0 |
| 128 | 93.6 | 106.2 | 112.1 | 115.7 |
| 192 | 151.5 | 167.2 | 174.1 | 178.3 |
| 256 | 210.9 | 228.8 | 236.9 | 241.3 |
| 320 | 271.4 | 291.3 | 299.8 | 304.6 |
| 384 | 332.5 | 354.1 | 362.9 | 367.9 |
| 448 | 394.3 | 417.0 | 426.3 | 431.6 |
| 512 | 456.4 | 480.2 | 489.7 | 495.2 |
| 576 | 518.8 | 543.4 | 553.3 | 558.9 |
| 640 | 581.1 | 606.8 | 617.0 | 622.7 |
| 704 | 644.1 | 670.3 | 680.7 | 686.3 |
| 768 | 707.1 | 733.8 | 744.4 | 750.1 |
| 832 | 769.9 | 797.4 | 808.2 | 814.0 |
| 896 | 833.1 | 861.0 | 871.9 | 877.9 |
| 960 | 896.3 | 924.7 | 935.8 | 941.7 |
| 1024 | 959.4 | 988.5 | 999.6 | 1005.6 |

For all random graphs, Digital Annealer uses the same input parameters including maximum MCMC iterations. For each execution, output is the best solution found by all replicas.

As it can be seen, the difference between the execution time of Digital Annealer and NOVCA-I becomes less when p decreases. We point out that we have not adjusted Digital Annealer parameters for different graph sizes since NOVCA-I follows a specific algorithm regardless of problem size. Therefore, by optimizing input parameters, including number of iterations, for each graph size same quality of solution for

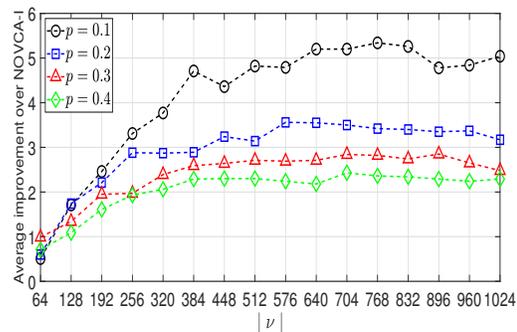


Fig. 1: Average improvement over NOVCA-I.

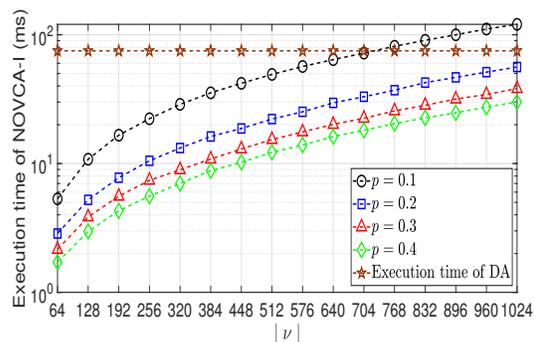


Fig. 2: Execution time of NOVCA-I and Digital Annealer for different values of $|\nu|$ and p .

smaller graphs can be achieved in lower execution time.

V. CONCLUSION

The problem of placing secure phasor measurement units (PMUs) to detect state and topology attacks has been studied. Specifically, it has been shown that placing the minimum number of secure PMUs to disable undetectable state and topology attacks can be achieved by solving the minimum vertex cover problem. In this paper, we proposed a digital annealing based method to place secure PMUs on smart grid. We derived an Ising model formulation for the vertex cover problem in the system topology and minimized its associated Ising energy using Digital Annealer. Numerical experiments indicated that our approach has an enhanced performance compared to other well-known alternatives in the literature. Specifically, Digital Annealer could find the optimal solution for 67 tested DIMACS benchmark graphs after tuning the annealing parameters.

VI. ACKNOWLEDGMENT

The authors would like to thank Fujitsu Laboratories Ltd. and Fujitsu Consulting (Canada) Inc. for providing access to Digital Annealer at the University of Toronto.

REFERENCES

- [1] Y. Liu, P. Ning and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," in *Proceedings of the 16th ACM conference on Computer and communications security*, 2009, pp. 2132.
- [2] J. Kim and L. Tong, "On Topology Attack of a Smart Grid: Undetectable Attacks and Countermeasures," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 7, pp. 1294–1305, July 2013.
- [3] L. Xie, Y. Mo and B. Sinopoli, "False data injection attacks in electricity markets," in *Proc. IEEE 2010 SmartGridComm*, Gaithersburg, MD, USA., Oct 2010.
- [4] L. Jia, R. J. Thomas and L. Tong, "Malicious data attack on real-time electricity market," in *Proc. 2011 IEEE Intl. Conf. Acoust. Speech and Sig. Proc. (ICASSP)*, Prague, Czech Republic, May 2011.
- [5] J. Kim and L. Tong, "On Topology Attack of a Smart Grid," in *Proc. the fourth Conference on Innovative Smart Grid Technologies (ISGT 2013)*, Washington, DC, February 2013.
- [6] C. Paar and J. Pelzl, *Understanding Cryptography: A Textbook for Students and Practitioners*, Springer, 2010.
- [7] 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 *First Workshop on Secure Control Systems, CPSWEEK 2010*, Stockholm, Sweden, Apr 2010.
- [8] O. Kosut, L. Jia, R. J. Thomas and L. Tong, "Malicious data attacks on smart grid state estimation: attack strategies and countermeasures," in *Proc. IEEE 2010 SmartGridComm*, Gaithersburg, MD, USA, Oct 2010.
- [9] A. Giani, E. Bitar, M. Garcia, M. McQueen, P. Khargonekar and K. Poolla, "Smart grid data integrity attacks: characterizations and countermeasures," in *2011 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, October 2011, pp. 232237.
- [10] T. Kim and H. Poor, "Strategic protection against data injection attacks on power grids," *IEEE Transactions on Smart Grid*, vol. 2, no. 2, pp. 326–333, June 2011.
- [11] J. Kim and L. Tong, "On phasor measurement unit placement against state and topology attacks," in *IEEE International Conference on Smart Grid Communications*, Oct. 2013.
- [12] R. Karp, "Reducibility among combinatorial problems," In R. E. Miller and J. W. Thatcher (eds.), *Complexity of Computer Computations*, Plenum Press, NY, pp. 85-103, 1972.
- [13] S. Bansal and A. Rana, "Analysis of Various Algorithms to Solve Vertex Cover Problem,"
- [14] T. Cormen, C. Leiserson, R. Rivest, "Introduction to Algorithms," *The MIT Press*, pp. 1022-1024, 2001.
- [15] R. Bar-Yehuda and S. Even, "A local-ratio theorem for approximating the weighted vertex cover problem," *North-Holland Mathematics Studies*, vol. 109, pp. 27-45, 1985.
- [16] B. Monien and E. Speckenmeyer, "Ramsey numbers and an approximation algorithm for the vertex cover problem," *Acta Informatica*, vol. 22, pp. 115-123, 1985.
- [17] S. Balaji, V. Swaminathan and K. Kannan, "Optimization of unweighted minimum vertex cover," *World Academy of Science, Engineering and Technology*, 43, 716-729, 2010.
- [18] I. Khan and H. Khan, "Modified Vertex Support Algorithm: A New approach for approximation of Minimum vertex cover," *Research Journal of Computer and Information Technology Sciences ISSN 2320*, (2013): 6527.
- [19] I. Khan, I. Ahmad and M. Khan, "AVSA, Modified Vertex Support Algorithm for Approximation of MVC," *International Journal of Advanced Science and Technology*, vol. 67, (2014), pp. 71-78.
- [20] S. Gajurel and R. Bielefeld, "A fast near optimal vertex cover algorithm (novca)," *International Journal of Experimental Algorithms (IJEAA)*, 3 (2012): 9-18.
- [21] S. Richter, M. Helmert and C. Gretton, "A Stochastic Local Search Approach to Vertex Cover," *Advances in Artificial Intelligence*, (2007): 412.
- [22] X. Xu and J. Ma, "An efficient simulated annealing algorithm for the minimum vertex cover problem," *Neurocomputing*, 69(7), 913-916, 2006.
- [23] K. Corder, J. V. Monaco, M. M. Vindiola, "Solving Vertex Cover via Ising Model on a Neuromorphic Processor," in *Circuits and Systems (ISCAS), 2018 IEEE International Symposium on*, pp. 1-5. IEEE, 2018.
- [24] S. Matsubara, H. Tamura, M. Takatsu, D. Yoo, B. Vatankhahghadim, H. Yamasaki, T. Miyazawa, S. Tsukamoto1, Y. Watanabe1, K. Takemoto1 and Ali Sheikholeslami, "Ising-Model Optimizer with Parallel-Trial Bit-Sieve Engine," in *Conference on Complex, Intelligent, and Software Intensive Systems*, pp. 432-438. Springer, Cham, 2017.
- [25] A. Lucas, "Ising formulations of many NP problems.," *Frontiers in Physics*, 2 (2014): 5.
- [26] DIMACS clique benchmarks, "Benchmark instances made available by electronic transfer at dimacs.rutgers.edu," *Rutgers Univ., Piscataway, NJ*. (1993).
- [27] B. Bollobas, "Random graphs," *2nd Ed., Cambridge, UK.*, Cambridge University press (2001).