AJSE

American Journal of Science & Engineering

Volume 2 Issue 2

August 2021

American Journal of Science & Engineering (AJSE)
Society for Makers, Artists, Researchers and Technologists (SMART)
6408 Elizabeth Ave SE, Auburn 98092, Washington, USA
ISSN: 2687-9530 (Print) and 2687-9581 (Online)

Editor-in-Chief

Dr. Chuck Easttom

University of Dallas, USA & Georgetown University, USA

Research Interest: Cryptography, Cyber Warfare, Engineering Processes and Digital Forensics.

Dr. Chuck Easttom is adjunct lecturer at Georgetown University and University of Dallas. He is the author of 31 books, including several on computer security, forensics, and cryptography. His books are used at over 60 universities. He has also authored scientific papers (over 70 so far) on digital forensics, machine learning/AI, cyber warfare, cryptography, bio-engineering, and applied mathematics. He is an inventor with 22 computer science patents. He holds a Doctor of Science (D.Sc.) in cyber security (dissertation topic: "A Comparative Study of Lattice Based Algorithms for Post Quantum Computing") and a Ph.D. in Technology focused on nanotechnology (dissertation topic: "The Effects of Complexity on Carbon Nanotube Failures"), as well as three master's degrees (one in applied computer science, one in education, and one in systems engineering). He is currently working on third doctorate, a Ph.D. in computer science with emphasis on applied mathematics from the University of Portsmouth (dissertation topic "On the application of algebraic graph theory to network forensics"). He is a Senior Member of the IEEE and a Senior Member of the ACM as well as a member of IACR (International Association of Cryptological Research) a member of APS (American Physical Society), and INCOSE (International Council on Systems Engineering). He is also a Distinguished Speaker of the ACM (Association of Computing Machinery). and a Distinguished Visitor of the IEEE Computer Society, and a frequent speaker at conferences. He also currently holds 55 industry certifications (CISSP, CASP, CEH, etc.) He is a member of IEEE Software & Systems Engineering Standards Committee. He has worked on the DevOps 2675 IEEE standards group 2017 to 2019 and currently a member of the IEEE Engineering in Medicine and Biology Standards Committee. Standard for a Unified Terminology for Brain-Computer Interfaces P2731.

From the Editor's Desk:

A thought about scientific rigor on research: As scientists, we must always be striving to produce not just more research, but better quality research. A researcher should be his or her own harshest critic. Look at your own work with a skeptical eye. Could you provide clearer data? Are your references adequate and current? Is your statistical analysis appropriate and robust? Our goal as scientists is not merely to publish research, but to produce research that is truly impactful. By constantly striving to improve the quality of our own work, we improve the entire body of work in any scientific field.

Editorial Board:

Editor-in-Chief - **Dr. Chuck Easttom** (University of Dallas, USA & Georgetown University, USA)
Associate Editor - **Dr. Nabeeh Kandalaft** (Grand Valley State University, USA)

Board Members -

i) **Dr. Phillip Bradford** (University of-Connecticut-Stamford, USA)
ii) **Dr. Alex "Sandy" Antunes** (Capitol Technology University, USA)
iii) **Dr. Izzat Alsmadi** (Texas A&M, San Antonio, USA)
iv) **Dr. Lo'ai Tawalbeh** (Texas A&M University-San Antonio, USA)
v) **Dr. Doina Bein** (California State University, Fullerton, USA)
vi) **Dr. Hasan Yasar** (Carnegie Mellon University, USA)
vii) **Dr. Moises Levy** (Florida Atlantic University, USA)
viii) **Dr. Christian Trefftz** (Grand Valley State University, USA)
ix) **Dr. Petros Spachos** (University of Guelph, Canada)

A Leader-Follower Game Theoretic Approach to Arrest Cascading Failure in Smart Grid

Sohini Roy, Arunabha Sen
School of Computing, Informatics and Decision System Engineering
Arizona State University
Tempe-85281, Arizona, USA
Email: {sohini.roy, asen}@asu.edu

***Abstract*—The Smart Grid System (SGS) is a joint network comprising the power and the communication network. In this paper, the underlying intra-and-interdependencies between entities for a given SGS is captured using a dependency model called Modified Implicative Interdependency Model (MIIM) [1]. Given an integer K, the K-contingency list problem gives the list of K-most critical entities, failure of which maximizes the network damage at the current time. The problem being NP complete [2] and owing to the higher running time of the given Integer Linear Programming (ILP) based solution [3], a much faster heuristic solution to generate an event driven self-updating K-contingency list [4] is also given in this paper. Based on the contingency lists obtained from both the solutions, this paper proposes an adaptive entity hardening technique based on a leader-follower game theoretic approach that arrests the cascading failure of entities in the SGS after an initial failure of entities. The validation of the work is done by comparing the contingency lists using both types of solutions, obtained for different K values using the MIIM model on a smart grid of IEEE 14-Bus system with that obtained by simulating the smart grid using a co-simulation system formed by MATPOWER and Java Network Simulator (JNS). The K-contingency list obtained for a smart grid of IEEE 14-Bus system also indicate that the network damage predicted by both the ILP based solution and heuristic solution using MIIM are more realistic compared to that obtained using another dependency model called Implicative Interdependency Model (IIM) [2]. Advantage of using the MIIM based heuristic solution is also shown in this paper when larger SGS of IEEE 118-Bus is considered. Finally, it is shown how the adaptive hardening helps in improving the network performance.**

Keywords—Interdependency relations, leader-follower game, entity hardening, contingency list, smart grid.

NOMENCLATURE

A. Power network entities

All the power network entities are denoted as P type entities in this paper.

1) $\boldsymbol{P_a}$: Bus with ID a.
2) $\boldsymbol{P_{a,b}}$: Transmission Line between Bus a and Bus b.
3) $\boldsymbol{P_{BattX}}$: Battery backup with ID X.

B. Communication network entities

All the communication network entities are denoted as C type entities in this paper. These C type entities are divided into 3 types–

1) ***Type 1***: Substation entity, denoted as $(C_{1,X,Y,Z})$. The values of X,Y and Z depends on the following subdivisions.

a) ***Substation Server*** $(\boldsymbol{C_{1,1,Y,Z}})$: Y is the server ID and Z is the substation ID.

b) ***Substation Gateway*** $(\boldsymbol{C_{1,2,Y,Z}})$: Y is the gateway ID and Z is the substation ID.

c) ***LAN wire between server and gateway*** $(\boldsymbol{C_{1,3,Y,Z}})$: Y is the LAN wire ID and Z is the substation ID.

d) ***Optical fiber channel between SONET-Add-Drop Multiplexer (SADM) and substation gateway*** $(\boldsymbol{C_{1,4,Y,Z}})$: Y is the SADM ID and Z is the gateway ID.

e) ***Optical fiber channel between Optical-Add-Drop Multiplexer (OADM) and substation gateway*** $(\boldsymbol{C_{1,5,Y,Z}})$: Y is the OADM ID and Z is the gateway ID.

f) ***Communication channel between Remote Terminal Unit (RTU) and substation gateway*** $(\boldsymbol{C_{1,6,Y,Z}})$: Y is the RTU ID and Z is the substation ID.

g) ***Communication channel between Phasor Measurement Unit (PMU) and substation gateway*** $(\boldsymbol{C_{1,7,Y,Z}})$: Y is the PMU ID and Z is the substation ID.

2) ***Type 2***: Synchronous Optical Networking Ring (SONET– Ring) entity, denoted as $(C_{2,X,Y,Z})$. The values of X,Y and Z depends on the following subdivisions.

a) ***SADM*** $(\boldsymbol{C_{2,1,Y,0}})$: Y is the SADM ID and Z=0 indicates that this type 2 entity is an SADM and not a connection.

b) ***Optical fiber channel between two SADMs*** $(\boldsymbol{C_{2,2,Y,Z}})$: *Y is the first SADM ID and Z is the second SADM ID in the link.*

3) ***Type 3***: Dense Wavelength Division Multiplexing Ring (DWDM–Ring) entity, $(C_{3,X,Y,Z})$

a) ***OADM*** $(\boldsymbol{C_{3,1,Y,0}})$: Y is the OADM ID and Z=0 indicates that this type 3 entity is an OADM and not a connection.

b) ***Optical fiber channel between two OADMs*** $(\boldsymbol{C_{3,2,Y,Z}})$: *Y is the first OADM ID and Z is the second OADM ID in the link.*

Copyright © AJSE Publishers

C. *Entities connecting the P type entities to the C type entities*

Entities that cannot be identified as P type or C type entities are termed as connecting entities. These entities define the interdependencies between the two types of network entities.

1) $\boldsymbol{L_{1,i}}$*:* Power supply line to a substation server where i is the ID of the line.

2) $\boldsymbol{L_{2,i}}$ *:* Power supply line to a substation gateway where i is the ID of the line.

3) $\boldsymbol{L_{3,i}}$ *:* Power supply line to an SADM and i is the ID of the line.

4) $\boldsymbol{L_{4,i}}$ *:* Power supplying channel to an OADM where i is the ID of the line.

5) $\boldsymbol{L_{5,i}}$ *:* Backup power supply line to a substation server from a substation battery where i is the ID of the battery.

6) $\boldsymbol{L_{6,i}}$ *:* Back up power supply line to a substation gateway from a substation battery where i is the ID of the battery.

7) $\boldsymbol{U_i}$ *:* Phasor Measurement Unit (PMU) with ID i.

8) $\boldsymbol{R_i}$ *:* Remote Terminal Unit (RTU) with ID i.

II. Introduction

The Smart Grid System (SGS) can be viewed as a two-layered network where one layer is composed of the power entities and the other layer is formed with communication entities. Yet, both the layers are connected to each other and the components of one layer depend highly on the components of the other layer for their operation. For example, the power system measurements of the smart grid obtained by its sensors must be transferred to the control center by the communication entities. Conversely, the communication entities themselves need power from the smart grid for their continued functionality. It should also be noted that the entities of each layer of the network also exhibit intra-dependencies among them. Therefore, if components of both the layers operate as required then only the SGS as a whole can function properly.

Now, due to this complex intra-and-interdependencies between the entities of two layers, if one or more entities in one layer of the SGS fail then other entities of both the layers will also fail as a result. Those newly failed entities will again initiate the failure of more entities. This is known as cascading failure [5] of entities. This cascading failure continues till a steady state is reached where this chain of failures breaks. Therefore, it is beyond any question that this cascading failure of entities can lead to a catastrophe where the whole SGS can fail. Thus, it is very essential to arrest the cascading failure as soon as it begins and protect the rest of the network from getting affected. In order to do that, the complex dependencies between the smart grid entities should be understood very well.

Identifying the need for clear understanding of the complex dependencies in a joint network, researchers made numerous efforts to come up with a model [6] that can vividly portray the smart grid system. Most of those models are too naïve to capture the complicated nature of interdependencies between the two kinds of networks in a smart grid. In [7] and [8] a high-level idea of the design of a joint network is given using a test system consisting of 14 buses. However, the ground level details of the Information and Communication Technology (ICT) network are missing in them. Moreover, the test systems presented in most of the papers differ a lot from the design of a real joint network. The Boolean logic-based implicative interdependency model (IIM) [2] overcame many of the afore-mentioned drawbacks. However, it also fails to accurately capture the communication network entities as it lacks knowledge of the communication network design. With the help of power utilities in the U.S. Southwest, the Modified Implicative Interdependency Model [1] presents a realistic design of the structure and operation of the power-and-communication network of a typical SGS. MIIM also considers different operational levels of the entities and models the complex dependencies between the two layers using multi-valued Boolean Logic based equations called Interdependency Relations (IDRs). In this paper an overview of the concepts of MIIM [1] are presented and those are used in order to arrest the cascading failures caused due to different attacks on the smart grid.

The SGS can face attacks from different types of attackers like the nature causing natural calamities, human causing physical or cyber-attacks and the smart grid itself can also act as an attacker to itself when cascading failures begin due to no external effects but the SGS entities themselves which fail to operate as desired. This paper proposes a novel approach that can defend any kind of attack to the SGS. This defense mechanism follows the Leader-Follower game theoretic concept [9]. Yet, in order to apply this technique, the Smart Grid Operators (SGOs) need a self-updating K-contingency list [4].

The set of entities, damage of which can result in the failure of maximum number of entities in the smart grid system are identified as the most critical set of entities. Upon this set of most vulnerable entities, the operability of the SGS is contingent and a list of such entities in the system is termed as the contingency list [4]. Usually, SGOs are provided with manuals that contain guidelines for handling different contingencies [10] in the system. Yet, in reality, when simulated contingencies do occur, the actual SGS may lie in a very different state than the simulations assumed. This results in either over-compensated or an under-compensated response. There comes the need for a measurement based self-updating contingency list which can provide real-time information to the operator about the current operational state of the entities. The current goal of the researchers is to find a suitable method to generate a Phasor Measurement Unit (PMU)-measurement based self-updating K-contingency list. In MIIM, each entity is associated with an operational state value of 0 indicating no-operation, 1 indicating reduced operation and 3 indicating full operation and these state values are updated each time a change in the operational level of an entity takes place. Such updating of operational values takes place on the basis of PMU data. Therefore, it can be stated that the state values of the entities in MIIM carry real-time information about the entities in the SGS. Ideally the self-updating contingency list for a given SGS can be identified just by solving the IDRs of the MIIM model each time a change takes place in the system. Efficient hardening [11] techniques followed for such critical entities can save the smart grid from a huge damage.

Yet, even after identifying all the vulnerable entities in the system, the smart grid operator can have a budget constraint of hardening only K entities of the network, where K can be any

Copyright © AJSE Publishers

integer. In that case, it is important to identify the K-most critical entities in the system. The problem of identifying the K-most vulnerable entities in a SGS is already proved to be NP complete in [2]. Therefore, an Integer Linear Programming (ILP) based solution [3] for the problem is given in this paper using the MIIM IDRs. Now, every time an event of failure or recovery or reduced operation takes place in the SGS, the IDRs change, and the K-Contingency list keeps on changing. It becomes very challenging to update the MIIM IDRs and also generate the ILP based K-Contingency list within 33 ms (considering PMU data is obtained at 30 samples per second). Owing to the computation complexity of the problem, it is very difficult to come up with an accurate solution within that time span. Therefore, a much faster heuristic solution for generating a self-updating K-Contingency list within the given 33 ms is also given in this paper. Validation of the results from both the ILP and heuristic solutions is done by co-simulating the two layers of the smart grid network of IEEE 14-Bus system using MATPOWER and Java Network Simulator (JNS). A comparative study of the K-contingency list obtained using the MIIM IDRs is done with that obtained using IIM for a smart grid of IEEE 14-Bus system also. This paper also shows how the heuristic solution is beneficial in case of larger smart grids like that built using IEEE 118-Bus system. The SGOs use this contingency list to defend the attackers and save the SGS from a catastrophe.

The rest of the paper is organized as follows. Section II gives an overview of the Implicative Interdependency Model (IIM) [2] and the Modified Implicative Interdependency Model [1]. Section III describes the K-Contingency list problem and also provides the Integer Linear Programming (ILP) based and heuristic solution for the problem. The simulated solution is also given in section III and the three types of solutions are explained using case studies in this section. Section IV describes the Leader-Follower game theoretic approach followed by adaptive entity hardening to defend different types of attacks on the smart grid system. Performance analysis of the three types of contingency list generation approach and also the adaptive entity hardening technique is discussed in section V of the paper. Finally, the paper is concluded, and future work prospects are given in section VI.

III. Overview of IIM and MIIM

In both IIM [2] and MIIM [1], the smart grid system can be viewed as a multilayer network, represented as a set $J(E, F(E))$, where E represents set of all entities in both the layers of the smart grid and $F(E)$ represents the set of IDRs. The entities in power layer (layer 1) are considered as P type entities where $P = \{P_1, P_2, \dots P_n\}$ and entities in ICT layer (layer 2) are named as C type entities where $C = \{C_1, C_2, \dots C_m\}$. The set $F(E)$ is used in both the models to capture the dependencies among interacting entities in the network. Yet, only structural dependencies are considered to generate the IDRs in IIM and both structural as well as operational aspects of the entities are taken into account while formulating IDRs for MIIM. IIM has a binary nature and the entities in that model can either be operational with a state value of 0 or be non-operational with a state value of 1. The most common feature of reduced operability in critical infrastructures is ignored in IIM. The entities in MIIM can take a value of 0, 1 and 2 indicating no-operation, reduced operation and full operation respectively.

Let C_i, an entity of layer 2, be operational if (i) C_j which is another entity of layer 2 and P_a which is an entity of layer 1, are operational, or (ii) C_k which is an entity of layer 2 and P_b which is an entity of layer 1 are operational, and (iii) C_l which is an entity in layer 2 is operational. Then the corresponding IIM IDR for C_i would be: $C_i \leftarrow \left((C_j \,.\, P_a) + (C_k \,.\, P_b)\right).C_l$. In this IDR, '.' denotes logical AND operation and '+' denotes logical OR operation. Similarly, the IDR for a P type entity can be expressed.

In MIIM, three Boolean operators are used while formulating the IDRs. The first operator is min-AND, denoted by '○', which selects the lowest of its input values. The second operator is max-OR, denoted by '●', which selects the highest of its input values. The third operator is new_XOR, which is denoted by '◉'. If all the inputs of new_XOR are same, then the output is also same as the inputs. In all other cases the output is 1. This new_XOR operator actually denotes the level of operation of an entity. The truth table for all the 3 new operators are given in Table I.

TABLE I. Truth Table for MIIM Operators

Input 1	Input 2	min-AND	max-OR	new_XOR
2	2	2	2	2
2	1	1	2	1
2	0	0	2	1
1	2	1	2	1
1	1	1	1	1
1	0	0	1	1
0	2	0	2	1
0	1	0	1	1
0	0	0	0	0

TABLE II. Evaluation of IIM and MIIM IDRs

	IIM	MIIM
Step 1	$C_l \rightarrow 0$	$C_l \rightarrow 0$
Step 2	$C_i \leftarrow (((2.2) + (2.2)).0)$	$C_i \leftarrow (((2 \circ 2) \bullet (2 \circ 2)) \circledcirc 0)$
Step 3	$C_i \leftarrow ((2 + 2).0)$	$C_i \leftarrow ((2 \bullet 2) \circledcirc 0)$
Step 4	$C_i \leftarrow (2.0)$	$C_i \leftarrow (2 \circledcirc 0)$
Step 5	$C_i \leftarrow 0$	$C_i \leftarrow 1$

In order to illustrate MIIM, let us assume that if an entity in condition (i) or (ii) fails, C_i will still work full operability, but if (iii) is not satisfied then C_i will operate at a reduced level; this relation can be expressed using MIIM IDRs as: $C_i \leftarrow \left((C_j \circ P_a) \bullet (C_k \circ P_b)\right) \circledcirc C_l$. To differentiate between the two models in terms of smart grid system application, the failure of entity C_l for the above IIM and MIIM IDRs are considered and the outcomes are observed in Table II. It is observed in Table II, that for same kind of dependencies, failure of the entity C_l results in the failure of entity C_i in case of IIM but it only reduces the operation level in case of MIIM.

Copyright © AJSE Publishers

IV. K-CONTINGENCY LIST PROBLEM

The operator of a smart grid system relies on the sensor-based data like PMU-data and RTU-data to know about the operational state of each and every entity in the power grid. Therefore, it is equally important for the operators to know about the operational states of the communication entities carrying data from the sensors placed in the substations to the control centers. If the operational level of an entity in the system reduces then immediate actions can be taken by the operator. Hence, at a real time, the entities which are more vulnerable to failure should be identified and proper protection or backup to those entities should be provided. This calls the need for an automated system generating the K-Contingency List for the current smart grid system, so that the maximum damage in the power-communication network can be avoided. When one or more entities fail in the smart grid system, many other entities also fail as a result and this is called cascading failures, and this often might lead to a catastrophe if not arrested in time. This cascade stops when the system reaches a steady state once again. Each time a failure takes place in the smart grid, the set $J\big(E, F(E)\big)$ is updated. All entities that get a state value 0 are removed from the set *E*. As a result, all the IDRs in set *F(E)* are also updated, since all the dependencies with those failed entities are removed. Now, in between two steady states of the system, there are a number of unstable states of the smart grid when the cascade propagates. Propagation of this cascade may not take place instantly and therefore measures can be taken to arrest the cascade by identifying the K-Contingency List at that time. Given an integer K, and a smart grid system represented as set $J\big(E, F(E)\big)$, this problem returns the set of K-most critical entities in the joint network, failure of which can lead to the maximum total number of failed entities in the system at the end of the cascade propagation. It is to be noted that a cascade can only propagate in one direction since an already failed entity cannot be affected again by the cascading failure. Therefore, upper bound of the cascade is $|EG| - 1$; where EG is the total number of edges in the network. A formal definition of the problem using the MIIM [1] model is as follows:

A. Inputs to the Problem

- (a) A joint network $J\big(E, F(E)\big)$; where $E = P \cup C \cup CP$
  - $P = B \cup T \cup Batt$ (Buses, Transmission Lines/Transformers, Batteries)
  - $C = SE \cup SRE \cup DRE$ (Substation Entities, SONET-Ring Entities, DWDM-Ring Entities)
  - $CP = L \cup R \cup U$ (Power supply lines, RTUs and PMUs)
- (b) Two positive integers K and S.

B. Decision version of the Problem

Does there exist a set of K entities in E whose failure at time t would result in a failure of at least S entities in total at the next state of the cascading process?

C. Optimization version of the Problem

Compute the set of K entities in the joint network $J\big(E, F(E)\big)$ whose failure at time t would maximize the number of entities failed or in other words minimize the overall system state values in the next state of cascade propagation.

The problem of finding K-Contingency List is NP complete, which is already proved in [2]. Therefore, an ILP based solution for the problem is given in section IV and a faster heuristic solution is given in Section V of this paper. Also, validation of the results should be done by comparing the ILP based and heuristic solution results with the simulation results.

D. Integer Linear Programming (ILP) based solution

In this section, an Integer Linear Programming (ILP) based solution for the K-Contingency List problem stated in Section III of this paper is given. The variable list for the problem is given below—

1) Variable List: For each entity $e_i \in E$ a variable set $x_{i,t} \forall t, 0 \le t \le |E| - 1$ is created. The value of $x_{i,t}$ is 2 if it is fully operational, 1 if it is operating at a reduced level of operation and 0 if it is non-operational.

2) Objective Function: The objective function for the problem can be defined as:

$$min \sum_{i=1}^{|E|} x_{i,|E|-1} \qquad (1)$$

This implies that, the problem aims at minimizing the system states for all the entities in the smart gird.

3) Constraint Sets:

a) Constraint set 1: $\sum_{i=1}^{|E|} x_{i,0} = K$, entities failed at time step 0 is K.

b) Constraint set 2: $x_{i,d} \le x_{i,t-1}$, $\forall t, 1 \le t \le |E| - 1$. This implies that, an entity can only have a system state value at a time $t > d$, less than or equal to the system state value it had at time d.

c) Constraint set 3: Based on the 3 new Boolean operations adopted by MIIM, IDRs can have the following format: $e_a \leftarrow (e_b \circledcirc e_c) \circ (e_m \bullet e_n)$.

- Step 1: Firstly, the above IDR can be reformed in the following way: $e_a \leftarrow z_{bcmn}$ where the new variable z_{bcmn} can be expressed as: $z_{bcmn} \leftarrow (g_{bc}) \circ (h_{mn})$ where the two new variables g_{bc} and h_{mn} can be further represented as: $g_{bc} \leftarrow e_b \circledcirc e_c$ and $h_{mn} \leftarrow e_m \bullet e_n$.
- Step 2: Now, a linear constraint is developed for the z type variable (associated with min_AND operator). In order to evaluate the IDR: $z_{bcmn} \leftarrow (g_{bc}) \circ (h_{mn})$, z_{bcmn} can be represented as: $z_{bcmn} \le g_{bc,t-1}$ and $z_{bcmn} \le h_{mn,t-1}, \forall t, 1 \le t \le |E| - 1$.
- Step 3: A linear constraint is also developed for the h type variable (associated with max_OR operator). In order to evaluate the IDR: $h_{mn} \leftarrow e_m \bullet e_n$, h_{pq} can be represented as: $h_{mn} \ge x_{m,t-1}$ and $h_{mn} \ge x_{n,t-1}, \forall t, 1 \le t \le |E| - 1$.

Copyright © AJSE Publishers

- Step 4: For the g type variable, associated with the new_XOR operator, the following linear constraint is developed. The IDR: $g_{bc} \leftarrow e_b \circledcirc e_c$ is represented by the following set of linear equations: $g_{bc} \geq 0$, $g_{bc} \leq max\ \_state$, where max_state denotes the state value at the highest level of operability for an entity (2 in this case), and $N \times g_{bc} \leq x_{b,t-1} + x_{c,t-1}, \forall t, 1 \leq t \leq |E| - 1$. Here N denotes the number of operands on which the new_XOR operation is taking place.

E. Heuristic Solution for the problem

The heuristic solution to the self-updating K-Contingency list is completely based on the observations made during the ILP based solutions and simulations.

In order to solve the problem heuristically, first the smart grid system should be considered as a graph $G = (V_P, V_C, E_{PC}, E_{PP}, E_{CC})$ consisting of two different types of vertices V_P *and* V_C and three different types of edges E_{PC}, E_{PP} *and* E_{CC}. In this abstraction, V_P indicate the power network buses and V_C indicate the communication entities except the channels. All the power or communication channels that connect power and communication entities eg: power supply lines to the communication entities are denoted by E_{PC}, Transmission lines and transformers are denoted by E_{PP} and all communication channels are denoted by E_{CC}. We are assuming that any edge cannot be most critical as all power networks are (n-1) fault tolerant and all communication networks can adjust routing technique based on failed channels.

1) *Initially all the vertices in the graph are considered to be white in color.*
2) *Input:* $G = (V_P, V_C, E_{PC}, E_{PP}, E_{CC})$, K, set of MIIM IDRs and a state table having the state values of each entity.
3) *Step 1:* The V_P vertices corresponding to generator buses in the actual grid are identified and colored yellow.
4) *Step 2:* The V_P vertices corresponding to buses with a PMU in the actual grid are identified and colored blue. Any V_P satisfying both the criteria of Step 1 and 2 will be green in color.
5) *Step 3:* Step 3 solves the problem for K=1
   - Consider a subgraph $G_1 = (V_P, E_{PP})$; since a failure of any communication entity cannot bring maximum damage to the smart grid.
   - If the graph has pendant vertices:
     - Identify the pendant V_P vertices.
     - Identify the V_P vertices connected to those pendant vertices and color them Pink.
   - Else if the graph does not have pendant vertices:
     - Identify the V_P vertices having minimum connections.
     - Color those nodes pink.
   - Check the total damage caused by failure of each such pink node by solving MIIM IDRs for those entities only.
   - Select the nodes resulting in maximum damage and color them red.
   - A list of all such red nodes comprise the K=1 contingency list.
   - Change all pink nodes to their previous color.
   - If K=1 then, Go to step 6 else go to step 4.
6) *Step 4:* Step 4 solves the problem for K=2
   - Take two empty lists List1 and List2.
   - Consider a subgraph $G_1 = (V_P, E_{PP})$; since a failure of just two communication entities cannot bring maximum damage to the smart grid.
   - Combine each of the red nodes to each of blue, green and yellow nodes to form all pairs of {Red, Green}, {Red,Yellow} and {Red, Blue}.
   - Check the total damage caused by failure of each such pair by solving MIIM IDRs for those entities in each pair only.
   - Find the {Red, G/Y/B} pair(s) failure of which causes the maximum damage.
     - Add the pair(s) in List1
   - Find all V_P vertices having two E_{PP} edges only.
   - Identify the V_P vertices connected to such V_P vertices having two E_{PP} edges only.
   - Color all such V_P vertices grey.
   - For all such pair of grey V_P vertices:
     - Check the total damage caused by the pair
     - Find the pair(s) causing maximum damage.
     - Add the pair(s) in List2
   - Compare the total damage caused by List1 pairs and List2 pairs.
   - Change all grey nodes back to their previously assigned color.
   - All the pairs causing maximum damage, comprise of the K=2 contingency list.
   - If K=2 then go to step 6, else go to step 5.
7) *Step 5:* If K>2 this step is executed
   - Round =0, TList1=Empty, TList2=Empty (Round is a counter and TList1 and Tlist2 are two temporary lists)
   - KCon_List =Empty (KCon_List is the K-Contingency List)
   - Graph G2←G1
   - While (TList2 is Empty)

Copyright © AJSE Publishers

- Find the list of K=2 most vulnerable entities in a list named List_Round (Using step 4 and the input graph G2)
- Remove all the entities in List_Round from the graph G2 and all the connections associated with them.
- Add the pairs in TList1
- If the number of pairs in TList1>=K/2
  - Find all combinations of the pairs in TList1 resulting in a K set.
  - Check the K set causing maximum damage using MIIM IDRs.
  - TList2 ←all such K sets.

- If K is Even
  - KCon_List←TList2
- Else
  - Consider graph (G1-{Entities in TList2})
  - Convert all the previous red nodes to their last assigned colors.
  - Repeat step 3.
  - Combine TList2 with each current red node obtained.
  - Check the damage caused by solving MIIM IDRs.
  - Find all combinations of TList2 and Red node causing maximum damage.
  - KCon_List ←Each such combinations.

6) *Step 6:* Check if any new failure takes place in the system.

- If yes
  - Update the state values in state table.
  - Remove IDRs of those entities.
  - Remove the entities from the input graph.
  - Repeat step 3 to 6.
- If No
  - Check if there are V_C vertices having all edges E_{PC} connecting them to the V_P entities in the failed list.
  - If yes:
    - Color such V_C vertices red
    - Add such V_C vertices in the K=1 contingency list.
  - Check if there are V_C vertices having all edges E_{CC} connecting them to the V_C entities in the failed list.
  - If yes:
    - Color such V_C vertices red
    - Add such V_C vertices in the K=1 contingency list.
  - The V_C vertices having all edges E_{PC} connecting them to the V_P entities in the contingency list, are also colored red and added to the K=1 contingency list.
  - The V_C vertices having all edges E_{CC} connecting them to the V_C entities in the contingency list, are also colored red and added to the K=1 contingency list.

The main goal of the heuristic solution of the self-updating K-Contingency list is to reduce the search space in order to reduce the computation time of the problem.

F. Simulated Solution for the problem

In order to validate the results obtained from the ILP and Heuristic Solutions of the K-Contingency list problem, the smart grid of IEEE 14-Bus system is considered and simulated with various contingencies. To simulate the SGS of IEEE 14-Bus MATPOWER is used for the power network layer and Java Network Simulator is used for the communication network layer. Co-simulation of the two layers is done by passing operation status values of the entities from one layer to the other layer. The same co-simulation platform can be used to simulate larger networks also but finding the K-Contingency list for larger networks by means of simulation and separation of failed entities, is difficult as the problem is NP complete.

G. Case Studies

In order to explain the working of the three types of solutions to find the self-updating K-contingency list, a smart grid system of IEEE 14-Bus is considered. In Fig.1, the P type or power entities and the Type 1 communication entities are shown.

Fig. 2. shows the Type 2 communication entities and Fig. 3. shows the Type 3 communication entities. In the smart grid of IEEE 14-Bus system, there are 14 buses and 34 communication terminals like servers, gateways, SADMs and OADMs. It is considered that the transmission lines and communication channels can fail when the entities at the two ends of it also fail. Therefore, IDRs of those entities are not considered. They can either have a state value 1 indicating they are operational or 0 denoting they have failed. However, the other 48 entities (14 *P type* and 34 *C type*) may depend on these transmission lines or communication channels and thus they are included in the IDRs of those 48 entities. Therefore, while finding the K-most vulnerable entities, only 48 entities are taken into account, but those 48 entities also cover the other entities which belong to categories like transmission lines or communication links.

Copyright © AJSE Publishers

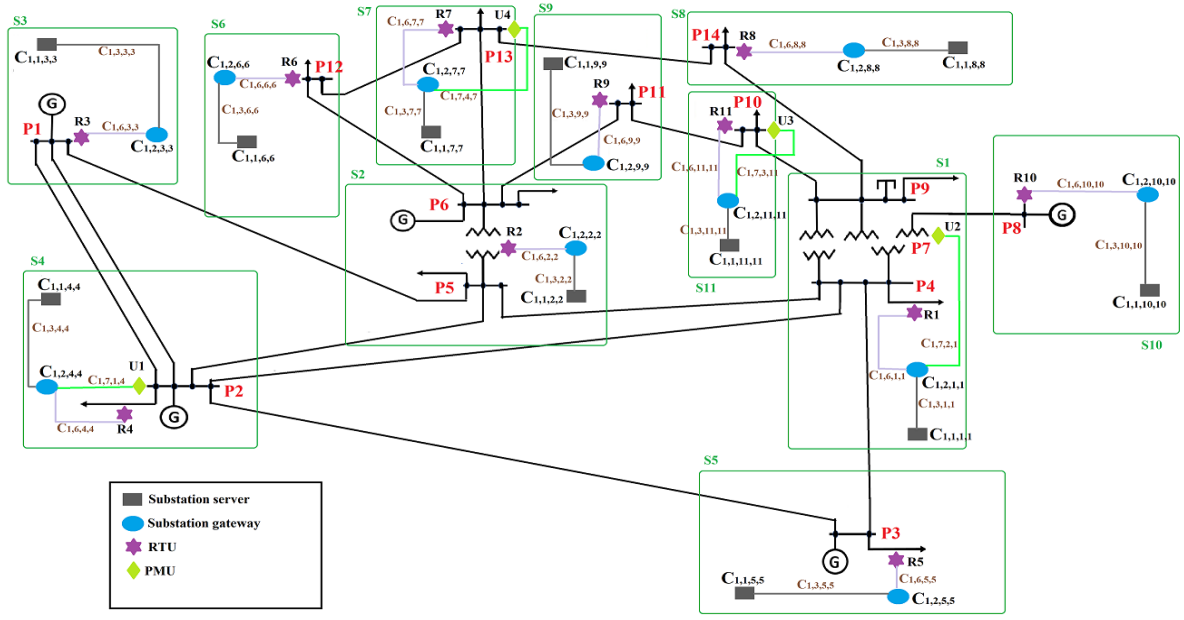


Fig. 1. Power Entities (P) and Type 1 communication entities of a smart grid of IEEE 14-Bus system

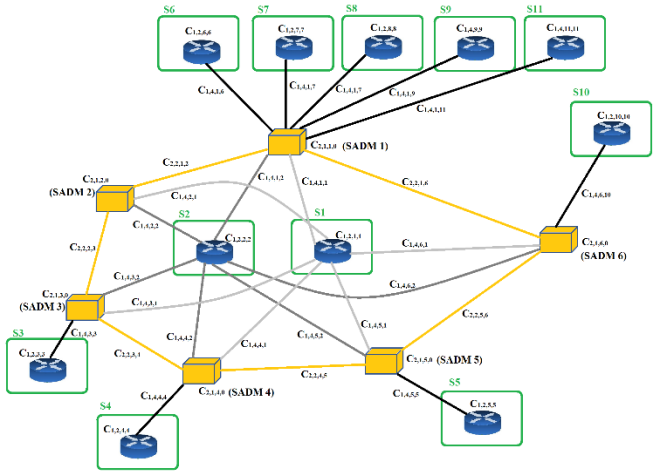


Fig. 2. Type 2 communication entities or SONET-Ring Entities (SRE)

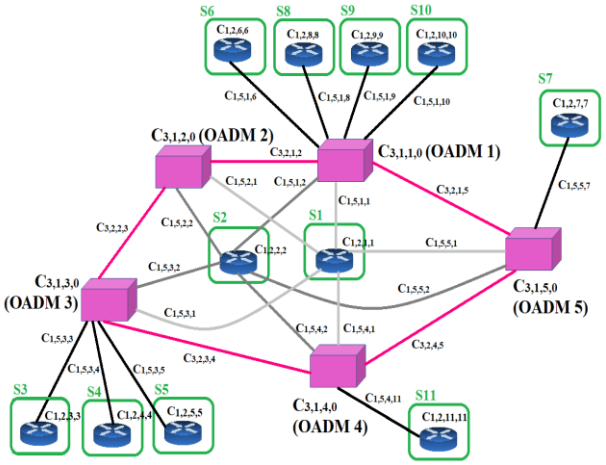


Fig. 3. Type 3 communication entities or DWDM-Ring Entities (DRE)

1) Case: Entity P_{12} fails initially

After bus P_{12} located in substation 6 of the smart grid of IEEE 14-Bus fails initially, the contingency list of the system for the next few seconds is analyzed using the MIIM based ILP, IIM based, heuristic solutions and simulation.

a) Case a: ILP based solution using IIM IDRs and MIIM IDRs

TABLE III. SELF-UPDATING CONTINGENCY LIST

T (ms)	MIIM Contingency List	IIM Contingency List
0	P_{12} fails	P_{12} fails
1	$\{P_7\}, \{C_{1,2,6,6}\}$, $\{C_{1,1,6,6}\}$	$\{P_7\}, \{C_{1,2,6,6}\}, \{C_{1,1,6,6}\}$
2	$\{P_7\}, \{C_{1,2,6,6}\}$, $\{C_{1,1,6,6}\}$	$\{P_7\}, \{C_{1,2,6,6}\}, \{C_{1,1,6,6}\}, \{C_{2,1,1,0}\}$
3	$\{P_7\}, \{C_{1,2,6,6}\}$, $\{C_{1,1,6,6}\}$	$\{P_7\}, \{C_{1,2,6,6}\}, \{C_{1,1,6,6}\}, \{C_{2,1,1,0}\}, \{C_{1,2,7,7}\}$, $\{C_{1,2,8,8}\}$, $\{C_{1,2,9,9}\}$, $\{C_{1,2,11,11}\}$, $\{C_{1,1,7,7}\}$, $\{C_{1,1,8,8}\}, \{C_{1,1,9,9}\}, \{C_{1,1,11,11}\}$
4	$\{P_7\}, \{C_{1,2,6,6}\}$, $\{C_{1,1,6,6}\}$	$\{P_7\}, \{C_{1,2,6,6}\}, \{C_{1,1,6,6}\}, \{C_{2,1,1,0}\}, \{C_{1,2,7,7}\}$, $\{C_{1,2,8,8}\}$, $\{C_{1,2,9,9}\}$, $\{C_{1,2,11,11}\}$, $\{C_{1,1,7,7}\}$, $\{C_{1,1,8,8}\}$, $\{C_{1,1,9,9}\}$, $\{C_{1,1,11,11}\}, \{C_{3,1,1,0}\}, \{C_{3,1,4,0}\}, \{C_{3,1,5,0}\}$
5	$\{P_7\}, \{C_{1,2,6,6}\}$, $\{C_{1,1,6,6}\}$	$\{P_7\}, \{C_{1,2,6,6}\}, \{C_{1,1,6,6}\}, \{C_{2,1,1,0}\}, \{C_{1,2,7,7}\}$, $\{C_{1,2,8,8}\}$, $\{C_{1,2,9,9}\}$, $\{C_{1,2,11,11}\}$, $\{C_{1,1,7,7}\}$, $\{C_{1,1,8,8}\}$, $\{C_{1,1,9,9}\}$, $\{C_{1,1,11,11}\}$, $\{C_{3,1,1,0}\}, \{C_{3,1,4,0}\}, \{C_{3,1,5,0}\}, \{C_{1,2,10,10}\}$, $\{C_{1,1,10,10}\}$

Table III shows the contingency list for 5 ms after P_{12} fails initially, given no new failures take place in the system within this time frame. From the given list below, the K-most vulnerable entities can be selected depending on the K-value. It

Copyright © AJSE Publishers

is also observed that IIM overestimates the number of contingent entities in the network after the initial failure of P_{12}.

For K=1, the most vulnerable entity will be P_7, for K=2 two sets will be obtained with same priority and any one of the sets can be chosen as K=2 most vulnerable entities. The two sets obtained for K=2 are: $\{P_7, C_{1,2,6,6}\}$ and $\{P_7, C_{1,1,6,6}\}$. This can continue till any K value which is less than the total number of entities in the system at the present state.

b) Case b: Heuristic solution using MIIM IDRs

In the heuristic self-updating contingency list solution, after the failure of P_{12}, the node corresponding to P_{12} in the input graph is removed and the state table is updated with the current 0 operational value of P_{12}. Now, the heuristic algorithm runs from step 3. P_8 is the pendant vertex in the graph $G_1 = (V_P, E_{PP})$. Therefore, P_7 is the most vulnerable entity in the network.

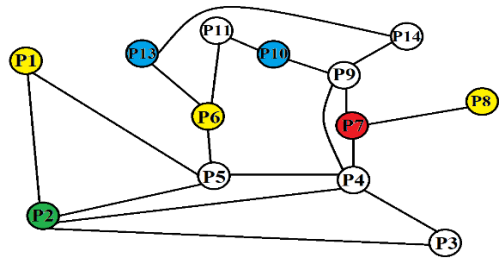


Fig. 4. K=1 most vulnerable entity for initial failure of P_{12}

Now the algorithm executes step 6 and the following entities are added in the list of K=1 contingency list: $\{(P_7), (C_{1,2,6,6}), (C_{1,1,6,6})\}$. Now all these entities are equally vulnerable but for K=1, a P type entity always gets more priority therefore P_7 will remain the most vulnerable entity in the system. In Fig.4., the K=1 most vulnerable entity is shown for initial failure of P_{12}. Yet, for K=2, if no new failures take place in the system, then the K=2 contingency list will have two pairs of entities–: $\{P_7, C_{1,2,6,6}\}$ and $\{P_7, C_{1,1,6,6}\}$.

c) Case c: Simulated Solution

The failure of bus P_{12} is simulated using MATPOWER and Java Network Simulator (JNS). It is observed that after the removal of bus P_{12} from the IEEE 14-Bus system, the power flows converge, and the rest of the network still operates. However, if bus P_7 fails then P_8 is isolated from the rest of the network and the power flows do not converge in this case. Therefore, the simulated solution also suggests that P_7 is the K=1 most vulnerable entity in the SGS. Using JNS, the same C type entities are found in the contingency list as the MIIM based ILP and heuristic solutions. The simulated result also proves that the results obtained using MIIM based ILP and heuristic solutions are valid.

V. Leader-Follower Technique to Arrest Cascading Failure

In a Smart Grid System (SGS) there can be three types of attackers, namely– intelligent attackers, predictable attackers and unpredictable attackers. Intelligent attackers are humans, predictable attackers can be the nature and unpredictable attackers are the smart grid entities themselves. Intelligent attackers can launch two types of attacks: physical attack and cyber-attack. Predictable attacker can launch only one type of attack which includes all different types of natural disasters and unpredictable attackers can self-destruct any SGS entity.

The leader-follower game also known as Stackelberg game is a game played sequentially between two players [9]. The first player is the leader who commits to the strategy first and then the second player or the follower, commits to his own strategy depending on the strategy of the leader. In such a game, a defender must perpetually defend a set of targets T using a limited number of resources. On the other hand, the attacker may or may not be able to observe and learn the defender's strategy and may attack after careful planning or randomly select targets to damage. In a smart grid scenario, the role of leader or follower can be decided on the basis of the type of attack taking place.

A. Different Types of Attacks

1) Game type 1: Attacks launched by intelligent attackers

Leader of Game type 1: In case of attacks launched by intelligent attackers, the Smart Grid Operator (SGO) becomes the leader of this game and also the defender. He selects the most vulnerable entities or the most critical entities in the smart grid beforehand and protect them from attacks or harden them from attacks. The leader also needs to predict the intensions of the attacker and find which of the entities in the network would be an easy target for the attacker. However, the leader has budget constraints and he can only protect a set of entities among all the entities in the smart grid. The mode of protection can be providing a backup device, providing strong security measures for particular entities etc. The type of hardening needed for a particular type of attack is decided by the SGO.

Follower of Game type 1: The follower of this game is the attacker. He learns about the strategy of the operator and plans his attack in such a way that he can maximize the damage in the network. Intelligent attackers of the SGS make a careful analysis of the defender's strategy and then come up with a new strategy to maximize the damage in the smart grid system.

a) Physical Attack: In case of physical or terrorist attacks, the attacker gets the location details of each substation in the SGS. Then after careful analysis of the defender's strategy to harden the smart grid entities, the intelligent attacks define a new strategy to physically damage particular substations in the SGS in such a way that the overall damage in the system is maximized. These intelligent attackers target those substations where the smart grid entities are not hardened. Example of physical attack can be an Electro Magnetic Pulse (EMP) attack [12].

b) Cyber Attack: In cyber-attacks, the target network layer is the communication layer and the entities targeted by the attackers are the ICT entities of the SGS. Just like physical attack, the cyber-attackers are also intelligent attackers and they select the ICT entities in the smart grid in such a way that the overall damage of the communication system can be maximized and as a result, the health monitoring of the critical entities in the SGS can be disrupted. Examples of cyber-attacks can be launching of Denial of Service attack [13], false data injection attack [14] etc.

Copyright © AJSE Publishers

2) Game type 2: Attacks launched by predictable attackers

Leader of Game type 2: In game type 2, the attacker acts as the leader. The attacker or mother nature defines her own strategy in which she selects random entities from the SGS and damage them.

a) Natural Disasters: The attacker in a type 2 game do not analyze the defender's strategy, rather a particular area in the smart grid is targeted by the attacker and entities in that area are damaged, irrespective of the fact that they are critical entities or not. It is very difficult to defend this attacker as it may or may not have a pattern that can be predicted beforehand. Examples of type 2 attacks causing damage to the smart grid system are– hurricanes, earthquakes etc.

Follower of Game type 2: The SGO becomes the follower as well as the defender in this game. The follower in game type 2 tries to predict the strategy of the leader depending on the type of attack the leader wants to launch. For example, if the SGO gets to know that a hurricane is coming, then he first finds the path that will be followed by the hurricane; by the help of weather analysts. Now, based on the path that will be followed by the hurricane, the SGO can predict which of the entities the hurricane can damage and then select the K-most critical entities from that region and harden them before the attacker launches the attack.

3) Game type 3: Attacks launched by unpredictable attackers

Leader of Game type 3: Just as in game type 2, in game type 3 also the attacker acts as the leader.

Here the attacker can be any entity of the smart grid itself which fails to operate. This attacker does not have a strategy and can attack any entity at any point of time. Therefore, they are unpredictable, and the game starts once the attacker launches an attack on the system. Examples of this type of attacks include damage of any ICT entity or failure of any power entity without any external influence.

Follower of Game type 2: The SGO becomes the follower as well as the defender in this game type as well. The follower in game type 3 comes to play once the leader has already started the game by failing one or more entities in the smart grid. Now based on the initial failures, the defender designs his strategy to arrest the cascading failure of entities as well as minimize the damage in the smart grid system.

B. Description of the Leader-Follower Technique

In the smart grid system, each target is a smart grid entity. It can either be a P type entity like bus, transmission line, transformer etc. or a C type entity like a server, gateway, communication channel etc. Each target is associated with a set of payoff values that define the utilities for both the defender and the attacker in case of a successful or failed attack [9]. It is assumed in the Stackelberg Security Games, that the payoff of an outcome depends only on the target attacked, and whether that target is hardened by the defender. For example, if an attacker succeeds in attacking a target entity T_1 of the smart grid, then the penalty for the defender is same, irrespective of the fact, that some other entity T_2 was hardened by the defender or not. This can be explained using the example in Table IV with only two entities T_1 and T_2.

TABLE IV. PAYOFF TABLE FOR DEFENDER AND ATTACKER

	Defender		Attacker	
Target	Hardened	Not Hardened	Hardened	Not Hardened
T_1	2	0	-1	1
T_2	0	-2	-1	1

The payoffs of the security game with only two targets can be shown as in table IV. A set of four payoffs is associated with each target. These four payoffs are the rewards and penalties to the attacker and the defender on the basis of a successful and unsuccessful attack. If a target is attacked, the utility of the defender's utility is given by: $U_D^h(t)$ if the target is hardened, or $U_D^n(t)$ if the target is not hardened. The attacker's utility can also be given in the same way: $U_A^h(t)$ if the target is hardened and $U_A^n(t)$ if the target is not hardened. The table given above shows the utility values. In reality, the $U_D^h(t)$ may correspond to the number of entities that are saved from damage by hardening a target entity and $U_D^n(t)$ may correspond to the number of entities damaged because of a successful attack on a not-hardened entity. Similarly, the utility from the attacker's perspective, $U_A^h(t)$ corresponds to a failed attack with no gain but a penalty of getting detected or penalty of the time devoted or the resources involved to place the attack; and $U_A^n(t)$ may correspond to a successful attack on a not-hardened entity and damage of that entity. It is observed from the table above, that from the defender's perspective, it is always better to harden an entity to gain the maximum utility. On the other hand, from the attacker's perspective, it is always better to have an entity not hardened by the defender so that it can launch the attack and gain a better payoff. However, it may not be feasible to harden all entities in the SGS due to budget constraints.

One approach for the defender would be to find the K-most critical entities in the smart grid system, failure of which can maximize the overall damage of the system at the end of cascading failures. Here K is the budget or the number of entities that the defender can harden at a given time. Now, the defender or the SGO can use K-Contingency list generated using the ILP based or heuristic method to select the smart grid entities which should be hardened. The SGO then hardens those K entities so that the attacker can do no harm to those entities. The hardening approach followed by the SGO is different for different types of entities in the SGS. This is the strategy that the leader of this game will take.

In the similar way, the intelligent attacker can also find the K-most vulnerable entities using the same method as the SGO. It is assumed that the follower or the attacker will know about this strategy and he will know which of the entities in the network are already hardened by the defender. Now, the approach of the attacker should be selecting the M most critical entities in the network where M is the budget of the attacker which denotes the number of entities it can attack at that time. The attacker carefully targets those M entities which should not overlap with the K entities already hardened by the defender. Therefore, if there are E number of entities in the smart grid

Copyright © AJSE Publishers

system, K of them are hardened by the defender, then the remaining entities will be: $(E - K)$. The attacker needs to find the M most critical entities out of the $(E - K)$ entities, the initial failure of which will maximize the damage at the end of the cascading failure process. In this way, the attacker will gain the maximum payoff that he could gain from the current scenario of the network, but the defender will also gain the maximum payoff as he has already hardened the entities, failure of which would have a larger impact on the smart grid.

Now, as the attacker targets the M entities which maximizes the failure of entities in the SGS after K entities are hardened, the next goal of the SGO is to arrest the cascading failure once initial attack on M entities take place.

Similarly, for type 2 attackers which do not have a strategy, the defender tires to predict the M entities out of E entities that might be targeted by the attacker. Then the SGO selects K most critical entities out of those M entities where K<M and hardens those K entities beforehand such that the attacker can possibly harm only $(M - K)$ entities. Thus, protecting the K most vulnerable entities in the targeted region, the defender can arrest the cascading failure of entities in the smart grid.

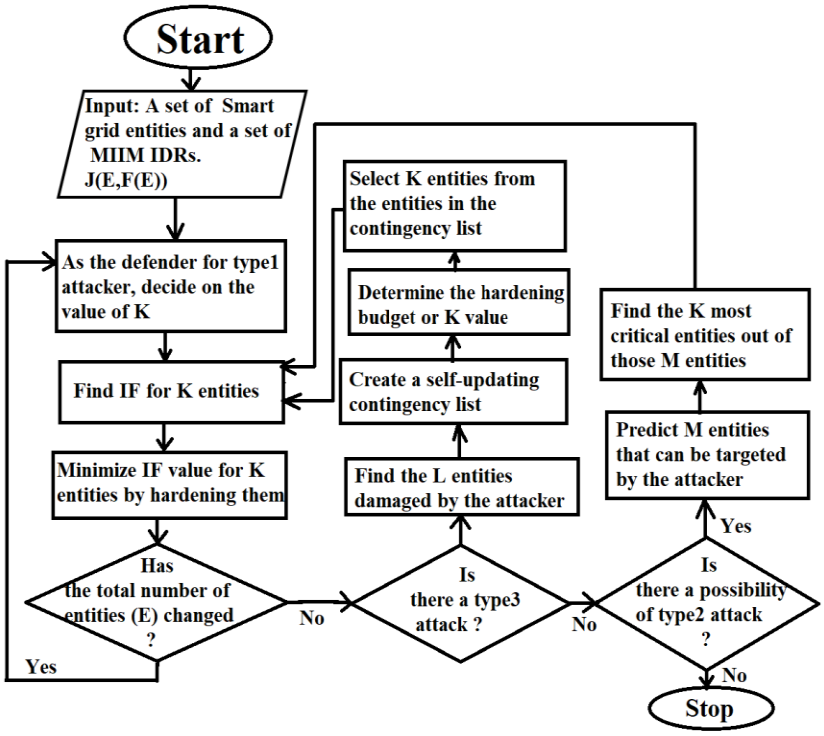


Fig. 5. Flowchart for adaptive hardening of entities

In case of type 3 game, the attacker randomly selects L entities to damage them. They are unpredictable attackers and any preventive measure cannot be taken by the defender. Once the attacker or the leader of this game starts the game by failing L entities, the follower or the defender quickly finds out the list of contingent items that can be harmed as a result. The defender selects K entities out of those $(E - L)$ entities and harden them to arrest the cascading failure of entities in the SGS.

The flowchart given in fig. 5. shows how the cascading failure can be arrested by adaptive hardening of K-most critical entities at a point of time in the SGS. When the K-most vulnerable entities are determined using the MIIM ILP based or MIIM heuristic solution, the impact factor for each such entity in the K-contingency list is determined. Impact Factor (IF) is nothing but the count of the number of entities that will affected as a result of failure of a particular entity. The effect of failure can be change in operational status or complete failure of the affected entity. Now the K-Contingency list is sorted on the basis of this IF value. The entity with the highest IF is selected first. The goal of this method is to arrest the cascading failure by hardening the entity with the highest impact factor. This hardening is done by somehow minimizing the IF value for that entity. Minimizing the IF value for an entity can be done in one or more of the following ways–

- Adjusting generation and load values: if the entity with the highest IF was supplying power to other entities then we can do some load shedding at the buses receiving power from it.
- Adding a backup device for this entity.
- Removing all edges from that entity. That means if the entity with the highest IF is a bus then all transmission channels connecting the entity with the rest of the network is removed and the power flows are adjusted within the rest of the network which now acts as a big separated island. On the other hand, if the entity is a C type terminal entity then all communication paths having that terminal should be avoided for data transmission to the control centers.

Copyright © AJSE Publishers

C. *Case Studies*

With the help of the following case studies, the efficacy of the leader-follower game theoretic approach to arrest cascading failure of smart grid entities can be shown. In order to perform the case studies, a comparatively large smart grid system of IEEE 118-Bus system is considered. The SGS of IEEE 118-Bus is divided into 8 operation zones and 107 substations. Table V shows how the whole SGS is divided into 107 substations and which of the buses are present in which substations. Out of the following 107 substations, substation 61 is selected as the main control center and substation 16 is selected as the backup control center. The basis of control center selection is same as in MIIM [1]. There is a total of 54 SONET Add Drop Multiplexers (SADMs) and 31 Optical Add Drop Multiplexers (OADMs). Fig. 6. shows the different zones in an IEEE 118-Bus system.

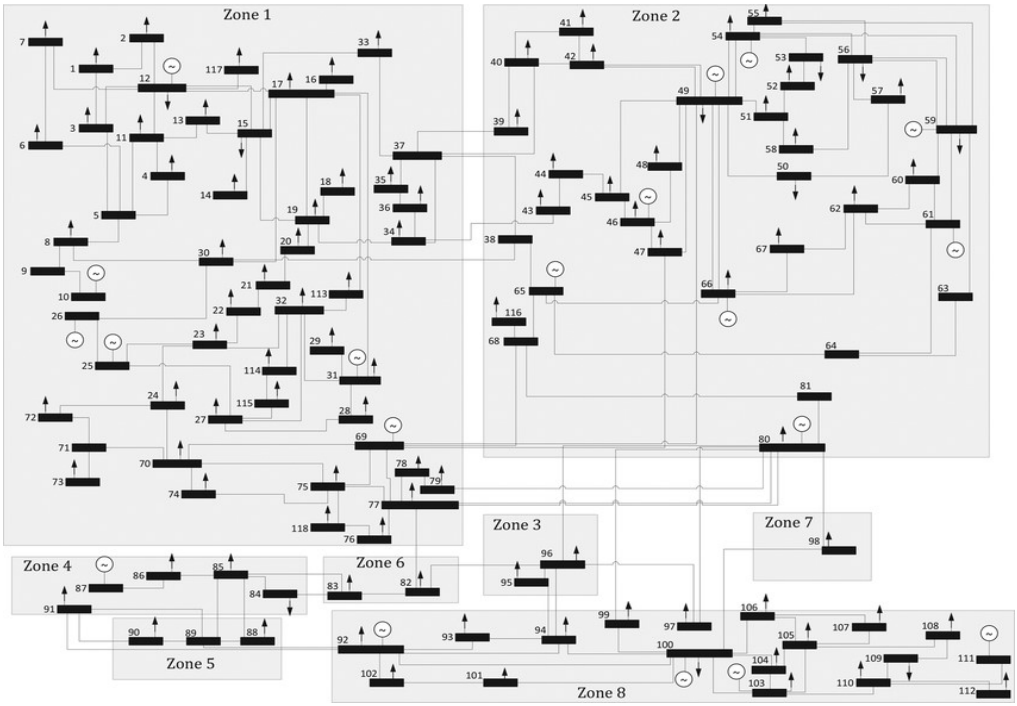


Fig. 6. Zone division of IEEE 118-Bus system

TABLE V. SUBSTATION DIVISION FOR 118-BUS SMART GRID

Substation ID	Buses	Substation ID	Buses	Substation ID	Buses	Substation ID	Buses	Substation ID	Buses
1	P1	23	P24	45	P49	67	P75	89	P99
2	P2	24	P25, P26	46	P50	68	P76	90	P100
3	P3	25	P27	47	P51	69	P77	91	P101
4	P4	26	P28	48	P52	70	P78	92	P102
5	P5, P8	27	P29	49	P53	71	P79	93	P103
6	P6	28	P31	50	P54	72	P80, P81	94	P104
7	P7	29	P32	51	P55	73	P82	95	P105
8	P9	30	P33	52	P56	74	P83	96	P106
9	P10	31	P34	53	P57	75	P84	97	P107
10	P11	32	P35	54	P58	76	P85	98	P108
11	P12	33	P36	55	P59, P63	77	P86, P87	99	P109
12	P13	34	P37, P38	56	P60	78	P88	100	P110
13	P14	35	P39	57	P61, P64	79	P89	101	P111
14	P15	36	P40	58	P62	80	P90	102	P112
15	P16	37	P41	59	P65, P66	81	P91	103	P113
16	P17, P30	38	P42	60	P67	82	P92	104	P114
17	P18	39	P43	61	P68, P69, P116	83	P93	105	P115
18	P19	40	P44	62	P70	84	P94	106	P117
19	P20	41	P45	63	P71	85	P95	107	P118
20	P21	42	P46	64	P72	86	P96		
21	P22	43	P47	65	P73	87	P97		
22	P23	44	P48	66	P74	88	P98		

Copyright © AJSE Publishers

1) Case 1: Natural Disaster (Hurricane)

Case 1 considers a type 2 attack– a hurricane. It is an attack by the type 2 or predictable attacker which do not have any defined strategy. However, the defender or the SGO can know about the path that will be followed by the hurricane from the weather analysts beforehand. Now, the SGO gets to know that a hurricane will pass diagonally through the smart grid region, from the corner of zone 8 towards zone 1. Therefore, the hurricane can affect smart grid entities in zone 8, zone 3 and parts of zone 1. According to the weather analysts, the hurricane can directly damage the following substations–101, 102, 100, 99, 93, 94, 90, 89, 85, 86. Now, the SGO can analyze the effect of the hurricane on the SGS using the MIIM IDRs. By solving the MIIM IDRs, the SGO comes to the conclusion that if the hurricane actually damages the substations mentioned above then many other substations will not be able to send Supervisory Control and Data Acquisition (SCADA) data and Phasor Measurement Unit (PMU) data to the control centers. Table VI shows which of the buses cannot send SCADA and/or PMU data to the control centers due to the damage of the aforementioned substations.

TABLE VI. CONTINGENCY LIST OR LIST OF VULNERABLE ENTITIES PREDICTED BY THE DEFENDER

Substation ID	Vulnerable Entities
85	P_{95},
86	P_{96}
88	P_{98}
89	P_{99}
90	P_{100}
93	P_{103}
94	P_{104}
99	P_{109}
100	P_{110}
101	P_{111}
102	P_{112}

Now, the defender can find the K-most critical entities out of the list of vulnerable entities given in table VI. If it is assumed that the value of K is 5, then the 5-Contingency list is identified by the defender or SGO in the table VII. The IF value for each entity in the 5-Contingency list is also given in the table.

TABLE VII. 5-CONTINGENCY LIST FOR CASE 1 AND THEIR IF VALUE

5-Contingency List	IF value
P_{100}	11
P_{110}	7
P_{96}	4
P_{104}	4
P_{111}	4

Then, the defender hardens these 5 entities in such a way that when the hurricane actually takes place, much lesser number of entities are damaged in the SGS. Fig.7. gives a comparison of the number of operational entities in the SGS if no hardening was done before the hurricane came versus the number of operational entities after the attack when hardening is done. The Fig. 7 also shows the number of operational entities in the normal condition.

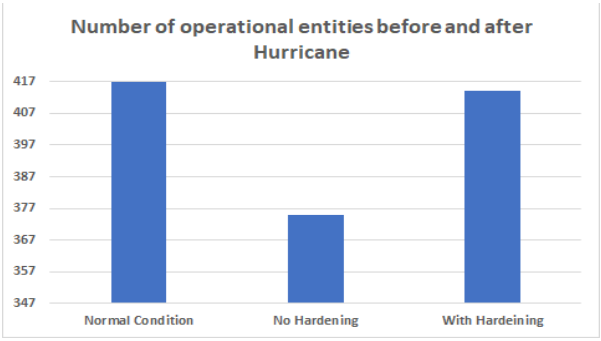


Fig. 7. Number of operational entities before and after the hurricane

2) Case 2: Electro-Magnetic Pulse Attack

In this case, a type 1 attack is considered. This type 1 attack is an Electro Magnetic Pulse (EMP) Attack launched by an intelligent attacker with a well-defined strategy. It is to be noted that, in order to arrest the type 2 attack by a predictable attacker, the defender needs to take action after a possibility of attack has raised. The process of defending type 2 attackers is mainly event driven or possibility driven. On the other hand, the SGO or the defender plans on arresting any attack from type 1 attackers from the set-up phase of the SGS. The defender designs his strategy to fail the type 1 attackers right after the SGS is formed and he uses his strategy to defend the attacker whenever a new device is added to the system or when some initial failure has damaged a portion of the SGS. So, in this case, even before any attack is planned by the type 1 attacker, the defender finds the K-most vulnerable entities in the SGS. If it is assumed that the value of K is 5, then the list of following entities are considered in the 5-Contingency list by the SGO.

TABLE VIII. 5-CONTINGENCY LIST FOR CASE 2 AND THEIR IF VALUE

5-Contingency List	IF value
P_{68}	301
P_{69}	301
P_{17}	90
$C_{1,1,61,61}$	299
$C_{1,2,61,61}$	299

Now, the entities in the contingency list are hardened by the SGO. It is assumed that the intelligent attacker knows about the strategy of the defender and therefore it launches an EMP attack on substation 16 which is the backup control center of the 118-Bus smart grid. In an EMP attack, physical damage of entities take place in the attack location. Therefore, all the entities in substation 16 which includes: buses P_{17} and P_{30}; and ICT entities $C_{1,1,16,16}$ and $C_{1,2,16,16}$ should get damaged. Yet, bus P_{17} is hardened by the defender and cannot be damaged by the attacker. Therefore, physical damage of the rest of the entities in substation 16 is done. Now the cascade also spreads from the site of EMP attack, so the defender finds a new set of K-contingency list again to harden them and thereby stop the cascading failure. It is assumed here that for arresting the cascade, the value of K considered is 3. Fig. 8. shows the number of operational entities before the EMP attack, number of operational entities after the EMP attack if no hardening was done and number of operational entities after the EMP attack when adaptive hardening of entities is done.

Copyright © AJSE Publishers

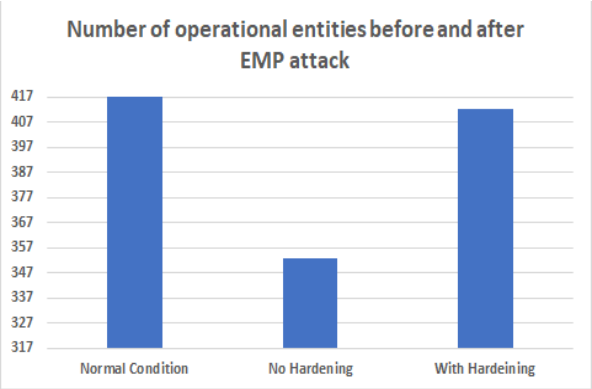


Fig. 8. Number of operational entities before and after the EMP attack

3) Case 3: Failure of an ICT entity without any external influence

In this case a type 3 attack is considered. The attacker in this case is gateway of substation 85 ($C_{1,2,85,85}$) which starts the game by self-destroying. The defender of the game gets alert and immediately finds the list of contingent items which may get affected as a result of the failure of ($C_{1,2,85,85}$). It is assumed that the value of K determined by the defender is 3 and table IX shows the list of entities in the 3-Contingency list which are selected for hardening.

TABLE IX. 3-CONTINGENCY LIST FOR CASE 3 AND THEIR IF VALUE

3-Contingency List	IF value
P_{95}	2
$C_{1,1,85,85}$	2
$C_{2,1,46,0}$	4

Now these entities are hardened by the SGO and fig. 9 shows the number of operational entities before the type 3 attack and after the cascading failure has stopped with hardened and unhardened entities.

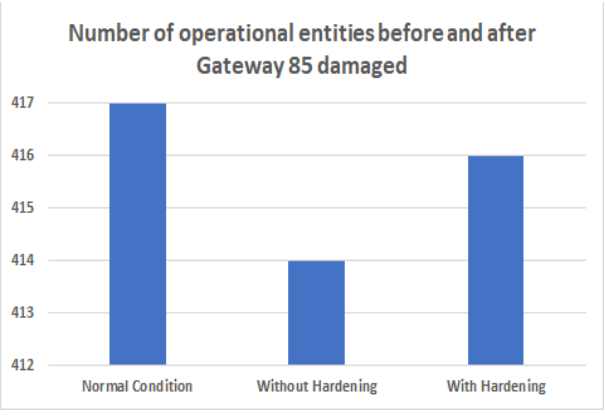


Fig. 9. Number of operational entities before and after Gateway 85 damaged

VI. PERFORMANCE ANALYSIS

In this section, a comparative analysis of the MIIM [1] ILP and heuristic solution, IIM [2] based K-contingency list solution and simulated solution is done. In order to do the performance analysis of the MIIM model based K-Contingency list identification methods (both ILP and heuristic) and thereby comparing the solutions with that based on IIM model and simulation results, both a small SGS of IEEE 14-Bus and a comparatively large SGS of IEEE 118-Bus is considered in this paper. A co-simulation platform using MATPOWER and Java Network Simulator is used in this paper to simulate the smart grid networks and find the K-Contingency list by simulation method. Java and CPLEX is used to run the MIIM and IIM ILP based approach and only Java is used for the MIIM based heuristic approach to find the self-updating K-Contingency list and also the attack based hardening approach. It is also shown in this section how the performance of the SGS is improved after leader-follower based hardening approach for the smart grid entities is followed.

A. Number of entities in the contingency list Vs. Time (for initial failure of P_{12}) in IEEE 14-Bus SGS

A type 3 attack is considered here and after bus P_{12} located in substation 6 of the smart grid of IEEE 14-Bus fails initially, the contingency list of the system for the next few seconds is analyzed using the MIIM based ILP and heuristic solutions, IIM based ILP solution and the co-simulation method in fig.10. It is observed that, the simulation results also give the same contingency list as MIIM. It is assumed that no new failures take place even after 5 ms of the failure of bus P_{12}. Based on the value of K, the most vulnerable entities in the contingency list are selected.

Now for type 3 game, the defender can only select entities from the contingency list and harden them based on the given budget K and the Impact Factor (IF) value of those entities.

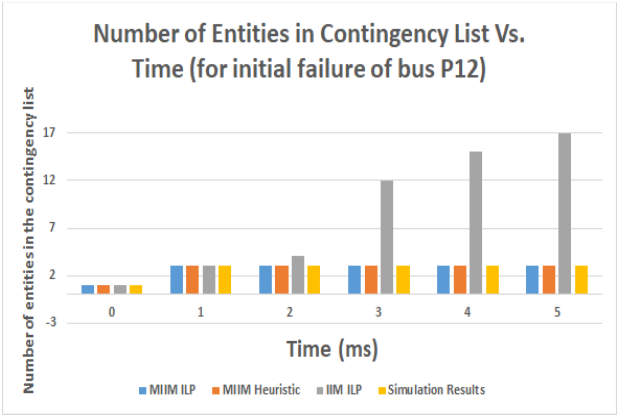


Fig. 10. Number of entities in the contingency list Vs. Time (for initial failure of P_{12})

B. Number of entities in the contingency list Vs. Time (for initial failure of P_1 and P_{12}) in IEEE 14-Bus SGS

Another type 3 attack is considered and in Fig.11 shows the contingency list for MIIM ILP, MIIM Heuristic, IIM ILP and Simulated result after P_1. and P_{12}. fails initially and no new failures take place even after 8 milliseconds. It is observed that the simulated result of contingency list is same as that obtained using MIIM ILP and MIIM Heuristic. The results obtained using IIM ILP differ a lot from the simulated contingency list. This validates the MIIM model and the ILP and heuristic solution based on MIIM.

Copyright © AJSE Publishers

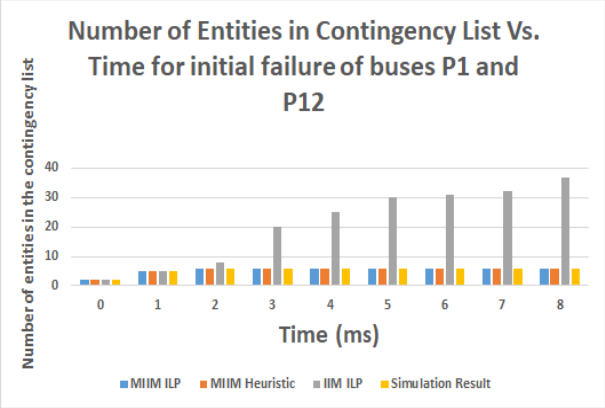


Fig. 11. Number of entities in the contingency list Vs. Time (for initial failure of P_1 *and* P_{12})

C. Number of entities in the contingency list Vs. Time (for a Type 2 attack) in IEEE 118-Bus SGS

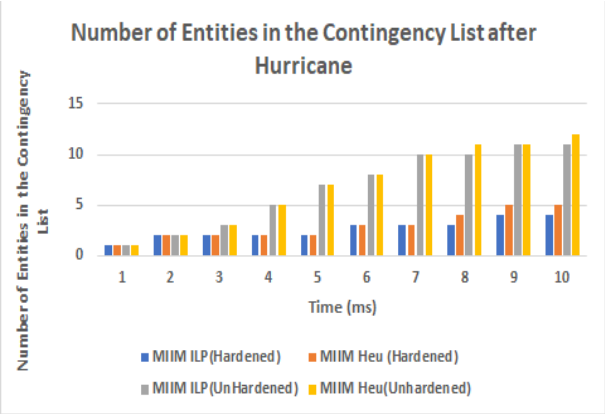


Fig. 12. Number of entities in the contingency list Vs. Time after a Hurricane

A type 2 attack is considered here. A hurricane is predicted to pass over zones 8, 5 and 4; and the SGO gets to know about that. He protects K entities in the contingency list generated on the basis of the prediction. K in this case is 4. Now, as the hurricane actually passes over the three zones, a new self-updating contingency list is generated in real-time to understand which of the entities can get damaged as a result of the attack. Fig. 12 shows a comparison of the MIIM based ILP and heuristic contingency lists for up to 10ms after the hurricane has passed, considering both the situations: (1) region having 4 hardened entities and (2) region having no hardened entities.

D. Number of entities in the contingency list Vs. Time (for an EMP attack on Substation 45) in IEEE 118-Bus SGS

A type 1 attack is considered in this case. It is assumed that the SGO has already hardened 10 entities out of the 417 entities in the 118-Bus smart grid network. Now, the attacker launches an EMP attack on substation 45, knowing that no entities in that substation is hardened beforehand. Fig.13 shows the number of entities in the MIIM ILP and heuristic solution based contingency list for up to 10ms after the attack took place, considering hardened and unhardened entities in the SGS.

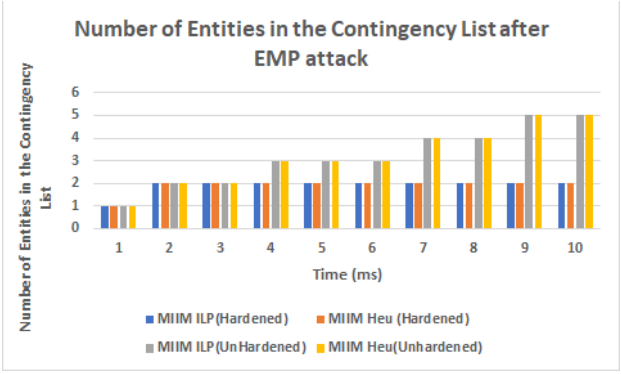


Fig. 13. Number of entities in the contingency list Vs. Time after a Hurricane

It is observed that even after 10 ms, the number of entities in the MIIM ILP as well as heuristic solution based contingency list is the same and consists of 2 vulnerable entities only. On the other hand, there are 5 vulnerable entities in the contingency list when no entities are hardened from beforehand.

It is to be noted that a high K value is given as input for identifying the vulnerable entities after an attack took place.

E. Maximum Entities damaged vs. K value for both IEEE-14 Bus and IEEE-118 Bus SGS

In fig.14., the maximum damage to the unhardened smart grid network of IEEE 14-Bus after the initial failure of K-most vulnerable entities are predicted by the ILP based solution to the problem using MIIM IDRs and IIM IDRs. Result obtained by solving the problem heuristically using MIIM IDRs is also shown in fig.14. The predicted damages are compared with the simulated results for a smart grid system of IEEE-14Bus.

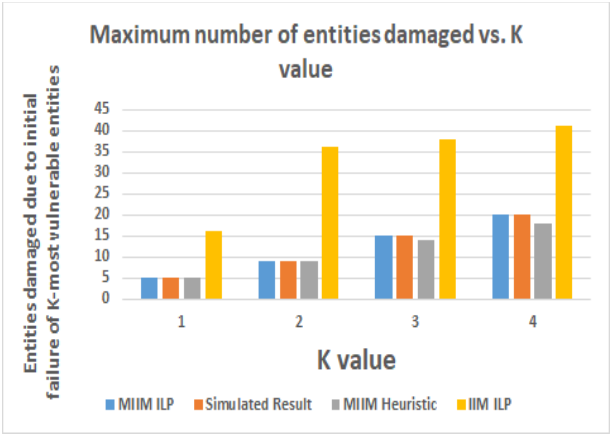


Fig. 14. Maximum number of entities damaged due to the initial failure of K-most vulnerable entities vs. K value (IEEE 14-Bus)

Similarly, in fig.15, the maximum damage to the unhardened and hardened SGS of IEEE 118-Bus after initial failure of the K-most vulnerable entities are shown using MIIM based ILP and heuristic solutions. The hardening is done in the similar manner as type 3 attacks. It is assumed that the K-most vulnerable entities are failing, and the network is not hardened from beforehand. Yet, after the game starts, the defender arrests the failure by hardening that K number of entities and stops the cascade.

Copyright © AJSE Publishers

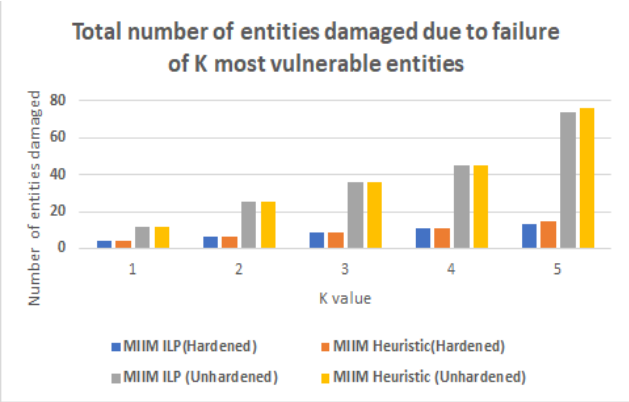


Fig. 15. Maximum number of entities damaged due to the initial failure of K-most vulnerable entities vs. K value (IEEE 118-Bus)

VII. Conclusion and Future Works

The Modified Implicative Interdependency Model (MIIM) used in this paper to determine the K-most critical entities works much better than existing interdependency models for critical infrastructure systems. It is observed from the performance analysis that even for larger networks like a smart grid system of IEEE 118-Bus can be protected from several types of attacks using this model and the proposed leader-follower game theoretic approach. In most of the entity hardening based research works, the scientists try to identify the critical entities beforehand and harden them based on their budget. Yet, this approach is not always helpful. Also, contingencies considered and simulated by the smart grid operators before an actual attack may not always match a real scenario. The proposed work considers all the different situations where the defender needs to decide whether he wants to be the leader of the game and take actions beforehand or to play the role of the follower and take actions after an attack is actually launched or a possibility of attack has arrived. The smart grid operator has to modify his strategy accordingly, so that he can protect the maximum number of entities in the smart grid from an attack or a failure. Again, it is proved in the performance analysis that this situation based adaptive hardening method actually performs better and can help in an improved operation of the smart grid system.

The techniques used in this paper to find the K-Contingency list can also be used for progressive recovery [15] of entities in a smart grid system after an attack has taken place in the system. This can be considered as a scope of future work.

References

[1] S. Roy, H. Chandrasekaran, A. Pal and A. Sen, "A New Model to Analyze Power and Communication System Intra-and-Inter Dependencies", *2020 IEEE Conf. on Tech. for Sustainability (SUSTECH 2020)*, Santa Ana, Apr. 2020, pp.181-188.

[2] A. Sen, A. Mazumder, J. Banerjee, A. Das and R. Compton, "Identification of K most vulnerable nodes in multi-layered network using a new model of interdependency," *2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Toronto, ON, 2014, pp. 831-836.

[3] S. Roy and A. Sen, "Identification of the K-most Vulnerable Entities in a Smart Grid System," *2020 3rd International Conference on Advanced Communication Technologies and Networking (CommNet)*, Marrakech, Morocco, 2020, pp. 1-6.

[4] S. Roy and A. Sen, "A Self-Updating K-Contingency List for Smart Grid System", accepted for publication in the *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*, 2021. [online: https://arxiv.org/abs/2101.08896]

[5] A. Das, J. Banerjee and A. Sen, "Root Cause Analysis of Failures in Interdependent Power-Communication Networks," *2014 IEEE Military Communications Conference*, Baltimore, MD, 2014, pp. 910-915.

[6] P. Pederson, D. Dudenhoeffer, S. Hartley and M. Permann, "Critical infrastructure interdependency modeling: a survey of US and international research", 2006.

[7] J. Sanchez, R. Caire, and N. Hadjsaid, "ICT and electric power systems interdependencies modeling," *Int. ETG-Congress Symp. 1: Security Critical Infrastructures Today*, Nov. 2013.

[8] W. Zhu and J. V. Milanović, "Cyber-physical system failure analysis based on Complex Network theory," *IEEE EUROCON 2017 -17th International Conference on Smart Technologies*, Ohrid, 2017, pp. 571-575.

[9] D. Kar. et al, "Trends and applications in Stackelberg security games". In: Basar T., Zaccour G. (eds) Handbook of Dynamic Game Theory. Springer, Cham, 2016.

[10] T. Van Cutsem and T. Weckesser, "Searching for plausible N-k contingencies endangering voltage stability," *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, Torino, 2017, pp. 1-6.

[11] J. Banerjee, A. Das, C. Zhou, A. Mazumder and A. Sen, "On the Entity Hardening Problem in multi-layered interdependent networks," *2015 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Hong Kong, 2015, pp. 648-653.

[12] Q. Wang, X. Zhou, X. Li and R. Jia, "The modeling and experimental investigation on coupling of transmission line network with electromagnetic pulse (EMP)," *2013 IEEE International Conference on Smart Energy Grid Engineering (SEGE)*, Oshawa, ON, Canada, 2013, pp. 1-8.

[13] S. Roy and A.K. Das, "Secure Hierarchical Routing Protocol (SHRP) for Wireless Sensor Network", In: Mauri J.L., Thampi S.M., Rawat D.B., Jin D. (eds) *Security in Computing and Communications. SSCC 2014*. Communications in Computer and Information Science, vol 467. Springer, Berlin, Heidelberg, 2014.

[14] J. Liang, L. Sankar and O. Kosut, "Vulnerability Analysis and Consequences of False Data Injection Attack on Power System State Estimation," in *IEEE Transactions on Power Systems*, vol. 31, no. 5, pp. 3864-3872, Sept. 2016.

[15] Y. Zhao, M. Pithapur and C. Qiao, "On Progressive Recovery in Interdependent Cyber Physical Systems," *2016 IEEE Global Communications Conference (GLOBECOM)*, Washington, DC, USA, 2016, pp. 1-6.

Copyright © AJSE Publishers

External Filtering and Wavelet Domain Thresholding-based Denoising Method for AWGN corrupted images

Sumit Singh Parihar[1], Shailesh Khaparkar[2]
Student[1], Assistant Professor[2]
Department of Electronics & Communication Engineering[1, 2],
Gyan Ganga Institute of Technology & Sciences, Jabalpur

ABSTRACT—In this work an image de-noising method with external bilateral filtering and wavelet domain thresholding has been proposed. In gaussian filtering fails to denoise an image at edges where the spatial variations are not smooth and cause the blurs the edges in the image. Bilateral filter overcomes this by filtering the image in both range and domain (space). Bilateral filtering is a local, nonlinear and non-iterative technique which considers both gray level (color) similarities and geometric closeness of the neighboring pixels. With bilateral filter the approximation sub-band results in loss of some image details, whereas that after each level of wavelet reconstruction flattens the gray levels cause unpleasing output image. To overcome the above issue extension of bilateral filtering with introduction of wavelets for thresholding has been proposed. Instead of direct filtering or direct wavelet domain thresholding of noisy image, the proposed method first obtains the filtered version of image using bilateral filtering and then this filtered version of image undergoes to wavelet domain thresholding using Bayes-shrink rules. In this approach the advantages of both the methods are achieved. To check the effectiveness of the proposed method in image denoising, we have compared the results with recent image denoising methods.

Keywords— Gaussian Noise, Image denoising, Filter Banks and Thresholding, Bilateral Filtering, Discrete Wavelet domain thresholding.

I. INTRODUCTION

An image is often corrupted by noise in its acquisition and transmission. For example during the image acquisition, the performance of imaging sensors is affected by a variety of factors, such as environmental conditions and by the quality of the sensing elements themselves. For instance, in acquiring images with a CCD camera, light levels and sensor temperature are major factors affecting the amount of noise in the resulting image. Images are also corrupted during transmission, due to interference in the channel used for transmission. Image denoising techniques are necessary to remove such random additive noises while retaining as much as possible the important signal features. The main objective of these types of random noise removal is to suppress the noise while preserving the original image details. Statistical filters like Average filter [1] [2], Wiener filter [3] can be used for removing such noises but the wavelet based denoising techniques proved better results than these filters. In general, image de-noising imposes a compromise between noise reduction and preserving significant image details. To achieve a good performance in this respect, a denoising algorithm has to adapt to image discontinuities. The wavelet representation naturally facilitates the construction of such spatially adaptive algorithms. It compresses essential information in a signal into relatively few, large coefficients, which represent image details at different resolution scales. In recent years there has been a fair amount of research on wavelet thresholding and threshold selection for signal and image denoising [4] [5] [6] [7] [8] [9], because wavelet provides an appropriate basis for separating noisy signal from image signal. Many wavelet based thresholding techniques like VisuShrink [10], BayesShrink [11] have proved better efficiency in image denoising. We describe here an efficient thresholding technique for denoising by analyzing the statistical parameters of the wavelet coefficients.

II. LITERATURE REVIEW

A. External Bilateral Filter

Filters based on Gaussian functions are of particular importance because their shapes are easily specified and both the forward and inverse Fourier transforms of a Gaussian function are real Gaussian functions. Further if the frequency domain filter is narrower, the spatial domain filter will be wider which attenuates the low frequencies resulting in increased smoothing/blurring. These Gaussian filters are typical linear filters that have been widely used for image denoising. Gaussian filters assume that images have smooth spatial variations and pixels in a neighborhood have close values, by averaging the pixel values over a local neighborhood suppresses noise while preserving image features. However, this assumption fails at edges where the spatial variations are not smooth and the application of Gaussian filter blurs the edges.

Bilateral filter overcomes this drawback by filtering the image in both range and domain (space). Bilateral filtering is a local, nonlinear and non-iterative technique which considers both gray level (color) similarities and geometric closeness of the neighboring pixels. Mathematically, the bilateral filter output at a pixel location p is calculated as follows:

$$I_f(p) = \frac{1}{w}\sum_{q\in S} G_{\sigma_s}(\|p-q\|)G_{\sigma_r}(|I(p)-I(q)|)I(q). \quad (2)$$

where, $G_{\sigma_s}(\|p-q\|) = e^{-\frac{\|p-q\|^2}{2\sigma_S^2}}$, represents geometric closeness function,

$G_{\sigma_r}(|I(p)-I(q)|) = e^{-\frac{|I(p)-I(q)|^2}{2\sigma_r^2}}$, represents gray level similarity function,

$W = \sum_{q\in S} G_{\sigma_s}(\|p-q\|)G_{\sigma_r}(|I(p)-I(q)|)$, represents a normalization constant.

$\|p-q\|$, represents Euclidean distance of p and q, with S as spatial neighborhood of p.

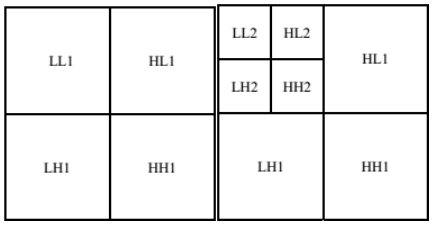
The parameters σ_s and σ_r control the performance of the bilateral filter. σ_s is responsible for blurring of image i.e., for large values of σ_s image will be blurred more; whereas σ_r is the noise

Copyright © AJSE Publishers

estimation parameter, and when equals to the noise standard deviation σ_n, results perfect denoising.

B. DISCRETE WAVELET TRANSFORM

The DWT is identical to a hierarchical sub-band system where the sub-bands are logarithmically spaced in frequency and represent octave-band decomposition. Due to the decomposition of an image using the DWT [12] the original image is transformed into four pieces which is normally labeled as LL, LH, HL and HH as in the



(a) One-level (b) Two-level

Fig. 1 Image decomposition by using DWT

schematic depicted in Fig.1(a). The LL sub-band can be further decomposed into four sub-bands labeled as LL2, LH2, HL2 and HH2 as shown in
Fig.1(b).

The LL piece comes from low pass filtering in both directions and it is the most like original picture and so is called the approximation. The remaining pieces are called detailed components. The HL comes from low pass filtering in the vertical direction and high pass filtering in the horizontal direction and so has the label HL. The visible detail in the sub-image, such as edges, have an overall vertical orientation since their alignment is perpendicular to the direction, of the high pass filtering and they are called vertical details. The remaining components have analogous explanations. The filters LD and HD shown in Fig. 2 are one-dimensional Low Pass Filter (LPF) and High Pass Filter (HPF) respectively for image decomposition. To obtain the next level of decomposition, sub band LL1 alone is further decomposed. This process continues until some final scale is reached. The decomposed image can be reconstructed using a reconstruction filter as shown in Fig. 3. Here, the filters LR and HR represent low pass and high pass reconstruction filters respectively. Here, since the image size is not changed after decomposition this DWT is called critically sampled transform without having any redundancy.

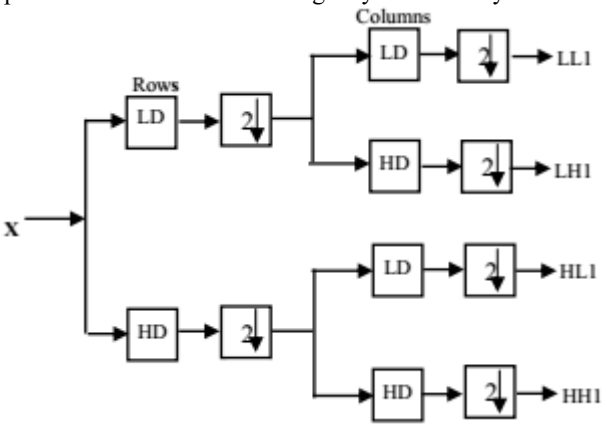


Fig. 2 Wavelet Filter bank for one-level image decomposition

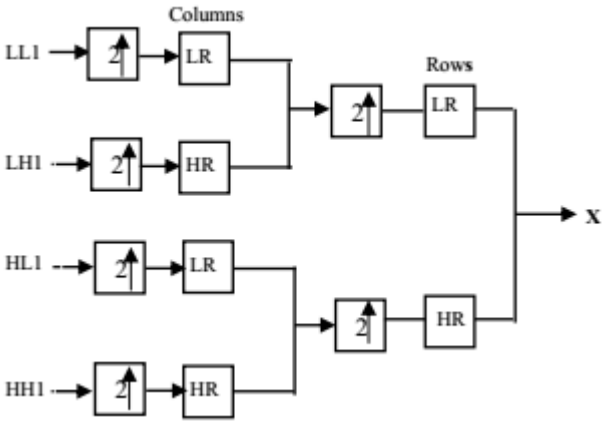


Fig. 3 Wavelet Filter bank for one-level image Reconstruction

An image is often corrupted by noise during its acquisition or transmission. The de-noising process is to remove the noise while retaining and not distorting the quality of the processed image .The traditional way of image de-noising is filtering. Recently, a lot of research about non-linear methods of signal de-noising has been developed. These methods are mainly based on thresholding the Discrete Wavelet Transform (DWT) coefficients, which have been affected by additive white Gaussian noise. Simple denoising algorithms that use DWT consist of three steps.

- Discrete wavelet transform is adopted to decompose the noisy image and get the wavelet coefficients.
- These wavelet coefficients are denoised with wavelet threshold.
- Inverse transform is applied to the modified coefficients and get denoised image.

The second step, known as thresholding, is a simple nonlinear technique, which operates on one wavelet coefficient at a time. In its most basic form, each coefficient is thresholded by comparing threshold, if the coefficient is smaller than threshold, set to zero; otherwise it kept as it is or it is modified. Replacing the small noisycoefficient by zero and inverse wavelet transform on the resulted coefficient may lead to reconstruction with the essential signal characteristics and with less noise.

During the last decade, a lot of new methods based on wavelet transforms have emerged for removing Gaussian random noise from images. The denoisingv process is known as wavelet shrinkage or thresholding. Both VisuShrink and SureShrink are the best known methods of wavelet shrinkage proposed by Donoho and Johnstone.

For VisuShrink, the wavelet coefficients w of the noisy signal are obtained first. Then with the universal threshold T (is the noise level and N is the length of the noisy signal), the coefficients are shrinked according to the softshrinkage rule is used to estimate the noiseless coefficients. Finally, the estimated noiseless signal is reconstructed from the estimated coefficients. VisuShrink is very simple, but its disadvantage is to yield overly smoothed images because the universal threshold T is too large.

Just like VisuShrink, SureShrink also applies the soft shrinkage rule, but it uses independently chosen thresholds for each subband through the minimization of the Stein's unbiased risk estimate (SURE) (Stein, 1981). VisuShrink performs better than SureShrink, producing more detailed images.

Copyright © AJSE Publishers

C. WAVELET THRESHOLDING

The first step in the denoising process is to obtain the wavelet transform of the signal x (n) using a suitable basis function. Then, a threshold is obtained using one of the above thresholding techniques [5]. Figure 4 shows the nature of thresholding.

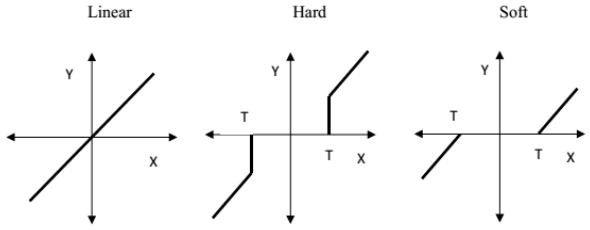


Figure 4: Linear, Hard and Soft Thresholding functions

The hard thresholding zeroes out, or shrinks the coefficients that have magnitudes below the threshold, and leaves the rest of the coefficients unchanged Soft thresholding extends hard thresholding by shrinking the magnitude of the remaining coefficients by , producing a smooth rather than abrupt transition to zero. The smooth transition to zero results in noticeably fewer artifacts upon reconstruction, especially when dealing with image denoising. Hence, soft thresholding is generally better for denoising due to its inherent smoothing, whereas hard thresholding is better suited for data compression. In either case, perfect reconstruction is not possible since some of the signal components are thrown away with the undesired noise. Furthermore, any thresholding technique other than the universal threshold will preserve some of the noise-only coefficients. Some significant research has been done using wavelet based de-noising.

The hard-thresholding T_H can be defined as:

$$T_H = \begin{cases} x, & |x| \geq t \\ 0, & inother \end{cases}$$

where, t is the threshold value. A plot of hard thresholding i.e., T_H is shown in Figure 4;

Thus, all coefficients whose magnitude is greater than the selected threshold value t remain as they are and the others with magnitudes smaller than t are set to zero. It creates a region around zero where the coefficients are considered negligible. Soft thresholding is where the coefficients with greater than the threshold are shrunk towards zero after comparing them to a threshold value. It is defined as follows.

$$T_s = \begin{cases} sign\ (x)(|x| - t), & |x| > t \\ 0, & otherwise \end{cases}$$

In general, it is observed that the hard thresholding technique is much better than soft thresholding and yields more visually pleasant images. This is because the soft thresholding technique is discontinuous and yields abrupt artifacts in the recovered images. Also, the hard thresholding technique yields a smaller minimum mean squared error compared to hard form of thresholding. Apart, from the soft and hard thresholding a custom trimmed thresholding is also considered in the work.

III. PROPOSED ALGORITHM

For better & easy understanding the proposed algorithm steps are as follows:

Step 1: Select input test image 'Lena' and resize it for fast computation.
Step 2: Add AWGN gaussian noise to obtain corrupted image.
Step 3: Apply bilateral filtering to noisy image.
Step 4: Subtract the filtered image from noisy image.
Step 5: The subtracted image undergone to wavelet decomposition.
Step 6: Then Bayes Thresholding is applied on the decomposed detailed wavelet coefficients.
Step 7: Select thresholding type i.e., soft, hard or trimmed.
Step 8: Reconstruction of image by taking IDWT of decomposed coefficients and adding them with filtered output.
Step 9: Quality measure calculations i.e., PSNR, MSE, MAE and SSIM.
The pictorial representation of above algorithm is shown in Figure 5.

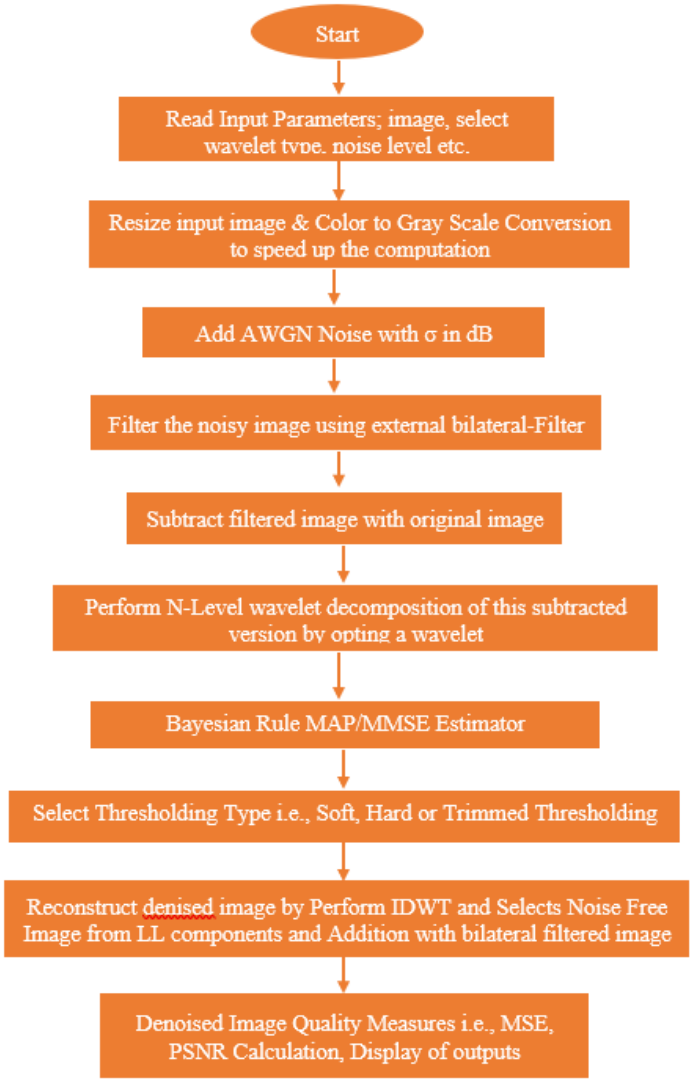


Figure 5: Proposed Algorithm

IV. SIMULATION RESULTS & DISCUSSIONS

To check the performance of the proposed image denoising using external bilateral filtering and wavelet domain thresholding technique, simulation has been performed for Lena test image using bilateral filters only and with the proposed method. The performance of the proposed method has been compared for MSE, MAE, PSNR and SSIM image quality parameters with

Copyright © AJSE Publishers

recent similar kind of image denoising methods. Simulation results values of PSNR & MSE for different wavelets are tabulated in Table-I.

Figure 6: Original test image 'Lena' used for simulation.

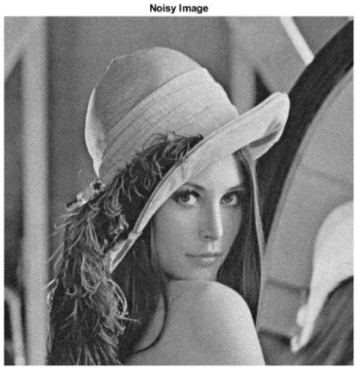


Figure 7: Gaussian noise corrupted test image 'Lena' for noise variance σ=10.

Figure 8: Denoised noisy image 'Lena' using external bilateral filtering only.

Figure 9: Denoised noisy image 'Lena' using proposed external bilateral filtering and wavelet domain thresholding.

For simulation first image is taken and it is corrupted by gaussian noise from range of 5 to 30 dB, to check the effectiveness of the proposed approach for a wide range of noise variance, then the image is denoised using external bilateral filter.

Table-I. Simulations Results Summary

Noise Variance	Noisy Image				Using external bilateral Filtering only				Using proposed external bilateral Filtering with wavelet domain thresholding			
σ	PSNR (dB)	MSE	MAE	SSIM	PSNR (dB)	MSE	MAE	SSIM	PSNR (dB)	MSE	MAE	SSIM
5	34.14	25.08	22	0.8421	34.58	22.65	38	0.9021	**37.26**	**12.32**	**26**	**0.9319**
10	28.13	100.02	45	0.6070	32.04	40.60	53	0.8687	**34.07**	**25.52**	**43**	**0.8884**
15	24.61	224.22	67	0.4466	30.64	56.18	71	0.8419	**32.01**	**40.98**	**56**	**0.8554**
20	22.13	398.11	89	0.3409	29.73	69.17	82	0.8172	**31.57**	**56.98**	**71**	**0.8269**
25	20.23	616.60	111	0.2700	29.05	80.84	88	0.7922	**29.38**	**75.01**	**83**	**0.7970**
30	18.70	877.39	133	0.2202	28.47	92.31	92	0.7668	**29.57**	**90.42**	**90**	**0.7685**

Table-II. Simulations Results Comparison

Noise Variance σ → / Algorithm ↓	**σ =10**	**σ =20**	**σ =30**
This Work	**34.07**	**31.57**	**29.57**
IDBP-CNN [1]	33.94	31.17	29.19
P&P-BM3D [1]	33.56	30.41	28.53
IRCNN [1]	33.13	31.17	29.31
IDBP-BM3D [1]	33.62	30.70	28.93
CD-B-k D [2]	33.95	31.35	29.55
HMT [3]	33.81	30.36	28.45
NIG-NSCT [5]	33.74	31.18	29.09
NIG-WT [5]	31.97	28.42	26.27
Bayes-Shrink [3]	33.29	30.14	28.26
NIG-CT [7]	33.32	31.06	29.33
AS-CT [10]	33.77	31.48	29.64
Visu-shrink [12]	30.65	27.76	26.33

To check the effectiveness of the proposed work, simulations results comparison has been done, which is shown in Table-II. It can be seen that the proposed approach seems to outperforms bilateral filtering and many of the existing denoising methods, in terms of denoised image PSNR for various values of noise variance.

V. CONCLUSION

In this work an image de-noising method with external bilateral filtering and wavelet domain thresholding has been proposed. To overcome the drawbacks of gaussian and bilateral filtering methods extension of bilateral filtering with introduction of wavelets for thresholding has been proposed. Instead of direct filtering or direct wavelet domain thresholding of noisy image, the proposed method first obtains the filtered version of image using bilateral filtering and then this filtered version of image undergoes to wavelet domain thresholding using Bayes-shrink rules. In this approach the advantages of both the methods are achieved. To check the effectiveness of the proposed method in image denoising, we have compared the results with recent image denoising methods. Simulation results shows that the proposed method outperforms many of existing image denoising methods.

REFERENCES

[1] Tom Tirer et al., "Image Restoration by Iterative Denoising and Backward Projections", IEEE Transactions on Image Processing, Volume: 28, Issue: 3, March 2019.

Copyright © AJSE Publishers

[2] H. Sadreazami et al., "Contourlet Domain Image Denoising based on the Bessel k-form Distribution", IEEE 28th Canadian Conference on Electrical and Computer Engineering Halifax, Canada, May 3-6, 2015.
[3] Liqiang Shi, "An Improved Image Denoising Algorithm", Seventh IEEE International Conference on Measuring Technology and Mechatronics Automation, 2015.
[4] Cuong Cao Pham et al., "Efficient image sharpening and denoising using adaptive guided image filtering", IEEE, IET Image Processing Magazine, Pp. 71 – 79, 2015.
[5] Wangmeng Zuo et al., "Gradient Histogram Estimation and Preservation for Texture Enhanced Image Denoising", IEEE Transactions on Image Processing, Volume 23, Nn. 6, JUNE 2014.
[6] Vikas Gupta et al., "Image Denoising using Wavelet Transform method", Tenth IEEE International Conference on Wireless and Optical Communications Networks (WOCN), Pp 1-4, 2013.
[7] Fuqing Jia et al., "Image Denoising Using Hyper-Laplacian Priors and Gradient Histogram Preservation Model", 12th IEEE International Conference on Signal Processing (ICSP), 2014.
[8] Ajay Boyat et al., "Image Denoising using Wavelet Transform and Median Filtering", Nirma University IEEE International Conference on Engineering (NUiCONE), 2013.
[9] Paras Jain & Vipin Tyagi, "Spatial and frequency domain filters for restoration of noisy images", IETE Journal of Education, 54(2), 108-116, 2013.
[10] Maggioni, M., Katkovnik, V., Egiazarian, K., Foi, "A.: Nonlocal transform-domain filter for volumetric data denoising and reconstruction", IEEE Transaction on Image Processing, 22(1), 119–133, 2013.
[11] Silva, R.D., Minetto, R., Schwartz, W.R., Pedrini, H.: Adaptive edge-preserving image denoising using wavelet transforms. Pattern Analysis and Applications. Springer, Berlin doi:10.1007/s10044-012-0266-x, 2012.
[12] Zhang, Y., Li, C., Jia, J, "Image denoising using an improved bivariate threshold function in tetrolet domain", 2013.
[13] Dai, L., Zhang, Y., Li, Y.: Image denoising using BM3D combining tetrolet prefiltering. Inf. Technol. J. 12(10), 1995–2001, 2013.
[14] He, K., Sun, J., Tang, X.: Guided image filtering. In: Proceedings European Conference on Computer Vision, pp. 1–14, 2010.
[15] Porikli, F, "Constant time O(1) bilateral filtering", In Proceeding IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, pp. 1–8, 2008.
[16] Yang, Q., Tan, K.H., Ahuja, N, "Real-time O(1) bilateral filtering", In Proceedings IEEE Conference on Computer Vision and Pattern Recognition, Miami, pp. 557–564, 2009.
[17] Farbman, Z., Fattal, R., Lischinski, D., Szeliski, "R.: Edge-preserving decompositions for multi-scale tone and detail manipulation", ACM Transactions on Graphics 27(3), 1–10, 2008.
[18] Paris, S., Durand, F, "A fast approximation of the bilateral filter using signal processing approach", In the Proceeding of European Conference on Computer Vision, pp. 568–580, 2006.
[19] Gonzalez, R.C., Woods, R.E.: Digital image processing, 3rd edn. Prentice-Hall, Upper Saddle River, 2008.
[20] Blu, T., Luisier, F, "The SURE-LET approach to image denoising", IEEE Transaction Image Processing, 16(11), 2778–2786, 2007.
[21] Paris, S., Durand, F.: A fast approximation of the bilateral filter using signal processing approach. In: Proceeding European Conference on Computer Vision, pp. 568–580, 2006.
[22] Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising with block-matching and 3D filtering. In: SPIE electronic imaging: algorithms and systems, vol. 6064, pp. 606414-1–606414-12, 2006.
[23] Yuan, X., Buckles, B.: Subband noise estimation for adaptive wavelet shrinkage. In: Proceeding 17th International Conference on Pattern Recognition, vol. 4, pp 885–888, 2004.
[24] Elad, M.: On the origin of the bilateral filter and ways to improve it. IEEE Transaction Image Processing, 11(10), 1141–1151, 2002.
[25] Sendur, L., Selesnick, I.W.: Bivariate shrinkage functions for wavelet-based denoising exploiting interscale dependency. IEEE Trans. Signal Process. 50(11), 2744–2756, 2002.
[26] Chang, S., Yu, B., Vetterli, M.: Spatially adaptive wavelet thresholding based on context modeling for image denoising. IEEE Trans. Image Process. 9(9), 1522–1531, 2000.

Copyright © AJSE Publishers

Compositional Behavioral Modeling of Analog Neural Networks

Ahmad Tarraf
Institute for Computer Science
Goethe University Frankfurt, Germany
tarraf@em.cs.uni-frankfurt.de

Lars Hedrich
Institute for Computer Science
Goethe University Frankfurt, Germany
hedrich@em.cs.uni-frankfurt.de

***Abstract*—This paper contributes to the automatic abstraction of analog circuits at transistor level. Specifically, this paper targets neuronal networks (NNs). As these circuits consist of millions of repeated neurons, simulation as well as verification routines are prohibitively time consuming. However, these netlists usually consist of repeated arrangements of neurons, which can be individually considered as subsystems. Starting with a neuron described as a Spice netlist, an abstraction methodology is presented that automatically generates an accurate behavioral model as a hybrid automaton (HA) in SystemC-AMS/Verilog-A while still preserving the internal voltages and currents of the subsystem. The abstracted model can replace the neuron in simulation as well as in verification routines with significant speedup factors while still achieving high accuracy.**

***Index Terms*—compositional abstraction, neuron, neuronal network, hybrid automaton, behavioral modeling, Verilog-A**

I. Introduction

Nowadays, NNs are gaining more importance due to their application spectrum. From applications including predication and association in autonomous driving, speech recognition, and process control, modern NNs are used even in the classification tasks in medicine. Thanks to the internet of things (IOT), the application spectrum of NNs is and will continue to increase. Even though NNs are widely applicable and user-friendly, the circuit simulation of such large circuits consisting of millions of neurons is a computational extensive task. Moreover, due to the state space explosion, a formal verification of such analog circuits is not feasible. In this contribution, we target these problems by presenting an automatic abstraction methodology capable of generating accurate behavioral models suitable for both, verification and simulation.

The abstraction of a single neuron, as described in Sec. III, is examined in this paper. Our methodology consists of two steps: sampling the Spice netlist and generating a HA from the sampled data. In Sec. IV the sampling process is examined, while in Sec. V the behavioral abstraction is described. A comparison between the abstract model and the Spice netlist is given in Sec. VI. Finally, a conclusion is stated in Sec. VII.

II. Previous Work

The abstraction of a neuron can speedup simulations routines and permit verification routines if the model is accurate enough. According to [1], formal verification of AMS circuits typically involves working on a higher level of abstraction, as this typically results in significant speedup factors. As NN consist of millions of neurons, an automated abstraction approach is mandatory. Different approach exist that generate behavioral models [2]–[5]. These techniques are not targeting hybrid automata and mainly improve the simulation speed. For high level continuous systems, methods modeling the analog circuit as a hybrid system are widely used [6]–[9]. These methods are able to handle up to tens of state variables, if the underlying locations use linear ordinary differential equations (ODEs) to describe the system behavior. To close the chain of proof at transistor level, HAs are usually not suitable as the ODEs become nonlinear differential algebraic equations (DAEs). In [10], an automated abstraction approach that generates accurate HAs from Spice netlists with BSIM 4 accuracy was presented. As the HAs have linear system descriptions, they are suited for compositional abstraction. In the following the applicability of this approach on a NN is examined by abstracting first a single neuron, followed by compositionally linking the abstracted models.

978-0-7381-4394-1/21/$31.00 ©2021 IEEE

III. Running Example: Neuron at Transistor Level

To demonstrate our approach, consider the single neuron shown in Fig. 1 with an adjustable ReLU activation function. The shown neuron is described in Spice with BSIM4 accuracy.

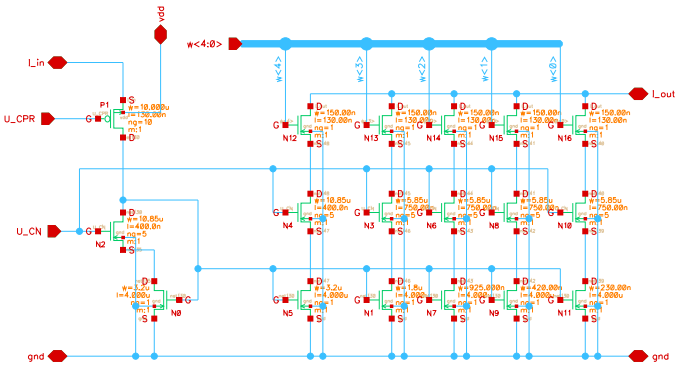


Fig. 1: Schematic of the neuron circuit on transistor level implementing a ReLU function.

IV. Sampling the State Space with *Vera*

The target of the abstraction methodology is to build a HA with a finite set of locations as described later in Section V.

For that, the circuit is sampled numerically using *Vera* [11]. The starting point is a nonlinear netlist description in Spice at transistor level. Significant data, such as the nonlinear consistent operating point, the eigenvalues of the linearized system and their nonlinear evolution, and a local linearized system description for each sample point are computed by *Vera* during the sampling at transistor level. The circuit is described by a system of nonlinear differential-algebraic equations:

$$\boldsymbol{f}(\boldsymbol{x}(t), \dot{\boldsymbol{x}}(t), u(t)), \tag{1}$$

where $\boldsymbol{x}(t) \in \mathbb{R}^n$ represents the vector of unknown voltages and currents in the original state space $\mathcal{S}_o$ of the system and $u(t)$ represents the input signal. For simplicity Eq. (1) is given for a SISO system. *Vera* steps through the state space as described in [11], linearizing the system locally and calculating the reachability of the sampled points of the overall nonlinear system. The result is a data set of sampled reachable points connected by a directed graph [12]. The sampled data additionally contains the operating points $\boldsymbol{x}_{DC} \in \mathbb{R}^n$, the conduction and capacitance matrices $\boldsymbol{G}$ and $\boldsymbol{C}$, and the input vector $\boldsymbol{b}$. The sampling results in the linearized system equation for each sample point:

$$\boldsymbol{C}\Delta\dot{\boldsymbol{x}} + \boldsymbol{G}\Delta\boldsymbol{x} = \boldsymbol{b}\Delta u \tag{2}$$

With $\Delta\boldsymbol{x} = \boldsymbol{x} - \boldsymbol{x}_{DC}$ and $\Delta u = u - u_{DC}$. With the transformation matrices $\boldsymbol{F}$ and $\boldsymbol{E}$, the system is transformed to a Kronecker form. Note that $\boldsymbol{F}$ is calculated from the right eigenvectors of the generalized eigenvalue problem associated with Eq. (2), while $\boldsymbol{E}$ is a proper calculated matrix from the same problem. For that, first the state space vector $\boldsymbol{x} \in \mathbb{R}^n$ is transformed from the original state space $\mathcal{S}_o$ to the new state space $\mathcal{S}_s$ with the state space vector $\boldsymbol{x}_s \in \mathbb{R}^n$ conform to:

$$\Delta\boldsymbol{x} = \boldsymbol{F}\Delta\boldsymbol{x}_s \tag{3}$$

Substituting Eq. (3) in Eq. (2), and multiplying the equation with $\boldsymbol{E}$ yields the Kronecker form:

$$s\begin{bmatrix}\boldsymbol{I}_1 & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{N}\end{bmatrix}\Delta\boldsymbol{x}_s + \begin{bmatrix}-\boldsymbol{J} & \boldsymbol{0}\\ \boldsymbol{0} & -\boldsymbol{I}_2\end{bmatrix}\Delta\boldsymbol{x}_s = \boldsymbol{E}\boldsymbol{b}\Delta u \tag{4}$$

Where $\boldsymbol{I}_1 \in \mathbb{R}^{r\times r}$ and $\boldsymbol{I}_2 \in \mathbb{R}^{(n-r)\times(n-r)}$ are identity matrices, the matrix $\boldsymbol{J} \in \mathbb{R}^{r\times r}$ is in general in the Jordan normal form corresponding to the finite eigenvalues, and $\boldsymbol{N} \in \mathbb{R}^{(n-r)\times(n-r)}$ is a nilpotent matrix corresponding to the infinite eigenvalues from the underlying generalized eigenproblem. Hence, the initial system with n variables has r dynamic variables and $(n-r)$ algebraic ones. Moreover, η represents the order of nilpotency of $\boldsymbol{N}$. By performing a dominant pole reduction similar to [13] on the obtained equation, the large order resulting from parasitic poles is reduced to the functional needed. That is, the order is reduced from r to m. For simplicity, two assumptions are made: 1) the finite eigenvalues are distinct, and 2) the index of nilpotency is $\eta \leq 1$. Thus, the following equation is obtained:

$$\underbrace{s\begin{bmatrix}\boldsymbol{I}_\lambda & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{0}\end{bmatrix}}_{\boldsymbol{ECF}}\begin{bmatrix}\Delta\boldsymbol{x}_{s,\lambda}\\ \Delta\boldsymbol{x}_{s,\infty}\end{bmatrix} + \underbrace{\begin{bmatrix}-\boldsymbol{\Lambda} & \boldsymbol{0}\\ \boldsymbol{0} & -\boldsymbol{I}_\infty\end{bmatrix}}_{\boldsymbol{EGF}}\begin{bmatrix}\Delta\boldsymbol{x}_{s,\lambda}\\ \Delta\boldsymbol{x}_{s,\infty}\end{bmatrix} = \underbrace{\begin{bmatrix}\tilde{\boldsymbol{b}}_\lambda\\ \tilde{\boldsymbol{b}}_\infty\end{bmatrix}}_{\boldsymbol{Eb}}\Delta u \tag{5}$$

With the identity matrices $\boldsymbol{I}_\lambda \in \mathbb{R}^{m\times m}$ and $\boldsymbol{I}_\infty \in \mathbb{R}^{(n-m)\times(n-m)}$, and the diagonal (or band-diagonal) matrix $\boldsymbol{\Lambda} \in \mathbb{R}^{m\times m}$ filled with the m finite eigenvalues. Note that transformed vectors are marked with a tilde ($\tilde{}$). Eq. (5) can be split into a dynamic part with subscript (λ) and the state vector $\boldsymbol{x}_{s,\lambda} \in \mathbb{R}^m$ in the reduced state space $\mathcal{S}_\lambda$, and an algebraic part with subscript (∞) and the state vector $\boldsymbol{x}_{\infty,\lambda} \in \mathbb{R}^{(n-m)}$ in the $\mathcal{S}_\infty$ space. For simplicity, $\boldsymbol{x}_{s,\lambda}$ will be denoted as $\boldsymbol{x}_\lambda$. Hence, there exist a relationship between the three state spaces and their corresponding state space vectors given by:

$$\Delta\boldsymbol{x} = \begin{bmatrix}\boldsymbol{F}_\lambda & \boldsymbol{F}_\infty\end{bmatrix}\begin{bmatrix}\Delta\boldsymbol{x}_{s,\lambda}\\ \Delta\boldsymbol{x}_{s,\infty}\end{bmatrix} \tag{6}$$

This shows the strength of our approach: to calculate the state vector $\boldsymbol{x}$ using Eq. (6), only the differential part from Eq. (5) must be calculated. The algebraic part can be found using the second row of Eq. (5). However, Eq. (5) is only pointwise valid. Thus, the objective is to model in each location of the HA the system behavior using a generalized form of Eq. (5) valid for a set of sample points of the same location.

V. Automatic Behavioral Abstraction to HA

The abstraction approach aims to deploy an abstract model in the form of a HA. For that, the finite set of locations are first determined. In each location the current invariant, the guards to the neighbor locations, and the system behavior given by the linear state space representation are identified. Additionally, the starting location of the HA and the jump conditions associated with the guards are found. A formal definition of the HA is similar to [14], with some restrictions on the jumps and guards. We aim to deploy a deterministic HA, that is, the invariants are only used to find the guards, a transition occurs immediately once a guard becomes valid, and during simulation the HA is only in one location. In the following, the detailed construction process is examined.

A. Finding the Locations of the HA

After the sampling process, a data set is obtained containing:

(a) all the $\boldsymbol{x} \in \mathbb{R}^n$ and $\boldsymbol{x}_\lambda \in \mathbb{R}^m$ sampled values from the $\mathcal{S}_o$ and $\mathcal{S}_\lambda$ state spaces, respectively
(b) the transformation matrices $\boldsymbol{F} \in \mathbb{R}^{n\times n}$ and $\boldsymbol{E} \in \mathbb{R}^{n\times n}$
(c) the eigenvalues of the linearized system after model order reduction $\boldsymbol{\Lambda} \in \mathbb{R}^{m\times m}$

Additionally, the data set contains as well the input vector $\boldsymbol{b} \in \mathbb{R}^n$ and the directed edges between the sampled points contained in a directed graph. For the neuron circuit from Fig. 1, the system has $n = 35$ nodal voltages and currents. Upon linearized, the system has a dynamic order of $r = 9$, with eigenvalues ranging from -8.68×10^1 to -2.6503×10^{15}, which is reduced by *Vera* to a reduced order of $m = 2$.

The next task is to find the finite set of locations $loc \in Loc$, where the sampled point exhibit similar dynamic behaviors. In order to find the locations of the HA, a group identification is performed that clusters points with similar eigenvalues into the same groups. Our algorithm uses an extended version of k-means clustering, where the number of clusters is either determined by the algorithm using the silhouette coefficient or

Copyright © AJSE Publishers

provided by the user for a higher accuracy. For the neuron from Fig. 1 the result of the clustering is illustrated in Fig. 2. As

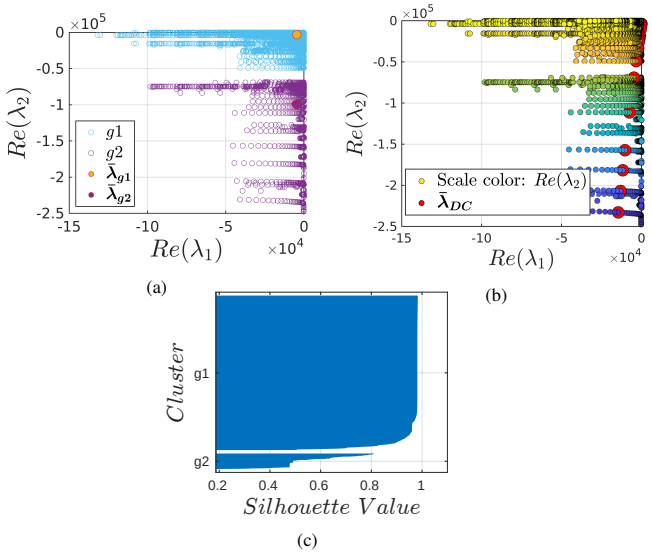


Fig. 2: In (a) the clustered eigenvalues of the linearized system are shown, while (b) presents the unclustered eigenvalues. (c) presents the corresponding Silhouette coefficients of the data points.

illustrated, two groups $g1$ and $g2$ were identified. According to Fig. 3b, the system is well clustered, especially for $g1$.

The second step required to find the locations of the HA, is the region identification. The region identification splits the obtained groups into regions. This is necessary in case several disjoint portions in the $\mathcal{S}_\lambda$ space belong to the same group. However, as Fig. 3 shows, this is not the case for the current example. In general, by using the directed graph obtained by the sampling process, or by using a clustering algorithm such as dbScan on the obtained groups, regions can be found.

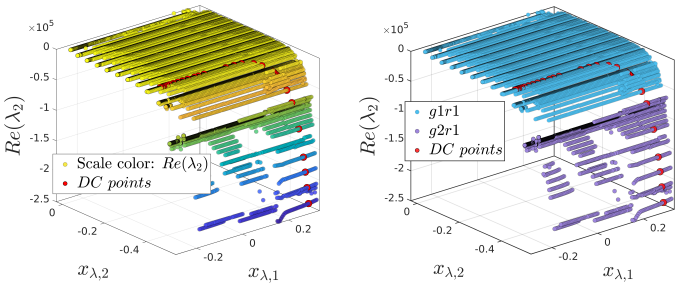


Fig. 3: $\mathcal{S}_\lambda$ space illustrated against the real part of the second eigenvalue of the linearized system. The large red points show the DC points. In (a) the initial unclustered data set is illustrated, while (b) presents the result of the region identification.

Together, the group and region identification compose the location identification. With gj denoting the j^{th} group and rk the k^{th} region, a location $loc_i \in Loc$ of the HA is denoted as:

$$loc_i = gjrk \tag{7}$$

For the current example, all point in a group gj belong to the same region $r1$. Hence, the locations of the HA are $Loc = \{g1r1, g2r1\}$ corresponding to a linear location $g1r1$ modeling the firing behavior of the neuron, and a blocking location $g2r1$ where the neuron is inactive.

B. Invariants and Guards of the HA

With the HA's locations at hand, the invariants and guards are identified next. As the data set sampled by *Vera* is distributed in accordance to the identified locations, the invariants (inv_{gjrk}) are simply found by enclosing the sampled points in the $\mathcal{S}_\lambda$ space with polytopes, zonotopes, or interval hulls. For the current application, polytopes are used (see Fig. 4). As we restricted the scope of this paper to deterministic HAs, invariants are only used to find the guards. Note that optionally, the model can issue warnings once the HA leaves an invariant.

To find the guards, the edges (facets in general) of the invariants are examined for the presence of points from the neighbor locations. In case the number of sampled points is above a specified tolerance value, the edge is taken as a guard. In general, guards can be modeled as halfspaces, polytopes, zonotopes, or interval hulls. For this paper, guards are always modeled as halfspaces. The l^{th} guard from the current location loc_i to the target location loc_t is denoted as:

$$grd_l : loc_i \rightarrow loc_t \qquad | \ loc_i, loc_t \in Loc \tag{8}$$

For the neuron from Fig. 1, the result of the abstraction approach is illustrated in Fig. 4. Note that $g1r1$ represents the starting location, as it contains the reference point $\boldsymbol{x}_\lambda = \mathbf{0}$.

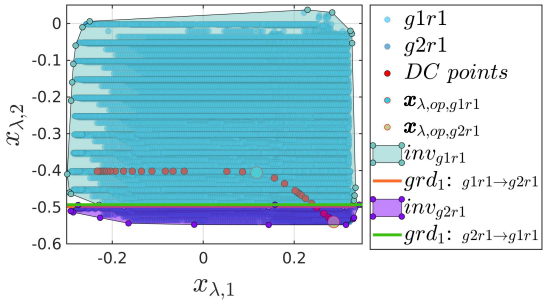


Fig. 4: $\mathcal{S}_\lambda$ state space of the HA with two locations.

C. Dynamics of the HA

Next, the dynamic system behavior in each location of the HA is described using the linear state space representation. For that, the operating point in each location is first identified. From the set of DC points (red points in Fig. 4), suitable operating points are chosen ($\boldsymbol{x}_{\lambda,op,gjrk}$). With these operating points, the system behavior in a location $loc_i \in Loc$ of the HA is described as:

$$\Delta\dot{\boldsymbol{x}}_\lambda = \boldsymbol{A}_{loc_i}\Delta\boldsymbol{x}_\lambda + \boldsymbol{B}_{loc_i}\Delta u\,, \tag{9}$$

such that:

$$\Delta\boldsymbol{x}_\lambda = \boldsymbol{x}_\lambda - \boldsymbol{x}_{\lambda,op,loc_i} \qquad \Delta u = u - u_{op,loc_i} \tag{10a}$$

$$\boldsymbol{A}_{loc_i} = \bar{\boldsymbol{\Lambda}}_{loc_i} \qquad \boldsymbol{B}_{loc_i} = \bar{\boldsymbol{E}}_{\lambda,loc_i}\bar{\boldsymbol{b}}_{\lambda,loc_i} \tag{10b}$$

Matrices with subscript λ correspond to sub-matrices belonging to $\boldsymbol{x}_\lambda$ (see Eq. (5)). $\boldsymbol{A}$ and $\boldsymbol{B}$ are computed from the mean values of the matrices of the sampled points belonging to the same location. For the running example, as the poles are always real, $\bar{\boldsymbol{\Lambda}}_{loc_i}$ is a diagonal matrix with the mean of the eigenvalues for each location (see different colored dots in Fig. 2a).

Copyright © AJSE Publishers

Using Eq. (9), $\boldsymbol{x}_\lambda$ is calculated in the $\mathcal{S}_\lambda$ space. In order to obtain the values of the nodal voltages and currents in the original state space $\mathcal{S}_o$, $\boldsymbol{x}$ must be calculated. This is done by a back-transformation obtained by combining Eqs. (5, 6):

$$\boldsymbol{x} = \boldsymbol{x}_{op,loc_i} + \bar{\boldsymbol{F}}_{\lambda,loc_i}\Delta\boldsymbol{x}_\lambda - \bar{\boldsymbol{F}}_{\infty,loc_i}\bar{\boldsymbol{E}}_{\infty,loc_i}\bar{\boldsymbol{b}}_{\infty,loc_i}\Delta u \quad (11)$$

D. Model deployment in Verilog-A/SystemC-AMS

The last modeling step is the deployment of the HA in the target language. For the Verilog-A models, the system behavior can be described using nodal equations and the *ddt* operator, while the SystemC-AMS models make use of the Time-Data-Flow model of computation (TDF-MoC) and use the TDF solver for the state space description (sca_tdf::sca_ss). In both cases, the guards are realized using if conditions. Once a guard becomes valid, the HA performs a location transition activating the jump condition by changing the operating points in Eq. (10a). In the new location, the system behavior is described by the corresponding matrices in Eqs. (9, 11).

VI. Results

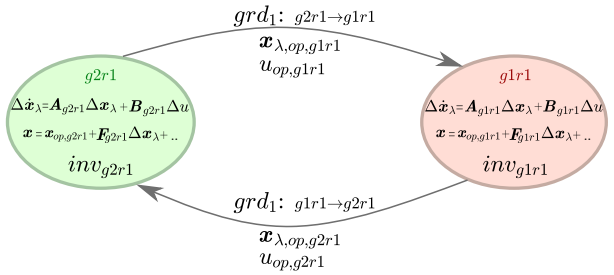


Fig. 5: Generated HA with two locations: a linear location ($g2r1$) and a nonlinear limiting location ($g1r1$).

As shown in Fig. 5, an abstract model in the form of a HA was generated in the previous sections from the circuit of a single neuron (see Fig. 1). The HA hast two locations: $g1r1$ and $g2r1$. During simulation, the HA starts from the center location $g1r1$ and computes the dynamic behavior using Eq. (9) with the corresponding values for Eq. (10) at $g1r1$. During each time step, the nodal voltages and currents (x values in $\mathcal{S}_o$) are calculated using the back-transformation Eq. (11) with the values of the matrices and operating points corresponding to $g1r1$. In case the guard $grd_1 : g1r1 \rightarrow g2r1$ becomes valid, that is:

$$\begin{aligned} 0 * \Delta\boldsymbol{x}_{\lambda,1} + 0.1650 * \Delta\boldsymbol{x}_{\lambda,2} < -0.01275 \\ \boldsymbol{x}_{\lambda,2} - (-0.4207) < -0.0773 \end{aligned} \quad (12)$$

the HA performs a transition to the location $g2r1$. Once the HA is in location $g2r1$, the corresponding values are used in Eqs. (9, 11). In case the guard of a location stays invalid during a simulation, the HA stays in the current location till the simulation time has elapsed. In the following, several examples are handled that use all this model, but with different arrangements and number of instances. All simulation were performed on a Linux PC with a four core i5-7300HQ CPU at 2.50 GHZ and 16 GB of RAM.

First, the generated HA (in Verilog-A) is compared against its original Spice netlist. The result is shown in Fig. 6. The last row of this figure shows the input voltage provided to both systems. Both models are simulated for 1 s with a timestep of 0.1 ms. As observed, the output voltages (second row) of both systems are nearly identical. Only immediately after/before a location transition of the HA, as indicated by the vertical purple lines, the voltages deviate. This is also shown in the first row of Fig. 6, which illustrates the output difference δ_y between the systems.

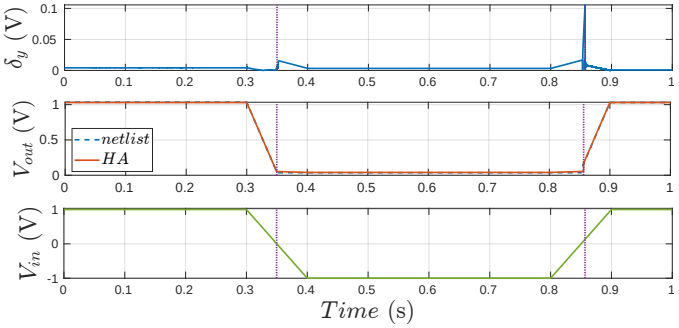


Fig. 6: Result of simulating the generated HA and the Spice netlist. The vertical lines indicated a location transition of the HA.

The detailed results are given in Table I (ex.1). Note that the HA can as well reconstruct internal voltages, however, this is skipped here.

A second example (ex.2) uses the same neuron in a configuration shown in Fig. 7. Three instances of the previous

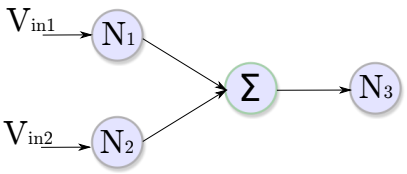


Fig. 7: Simple neuronal network consisting of three neurons.

generated HA are used to build the circuit. The circuit is compared to a Spice circuit with three neurons. For an input voltage as shown in the third row of Fig. 8, the result behave similar to the previous example for the same simulation conditions.

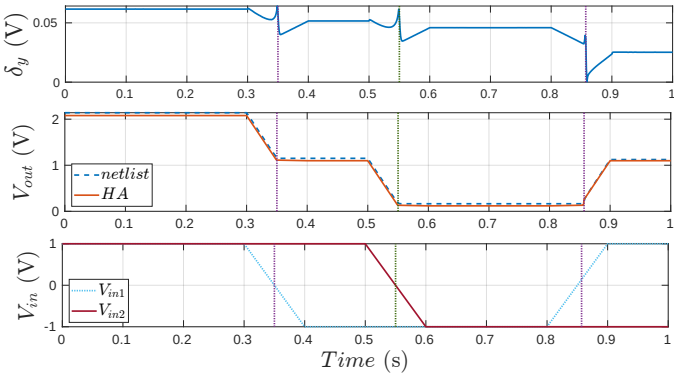


Fig. 8: Comparison of the simulation results from ex.2 (see Fig. 7).

The third example (ex.3) uses the 2-1 configuration from Fig. 7 several times to build-up a NN with 15 neurons in

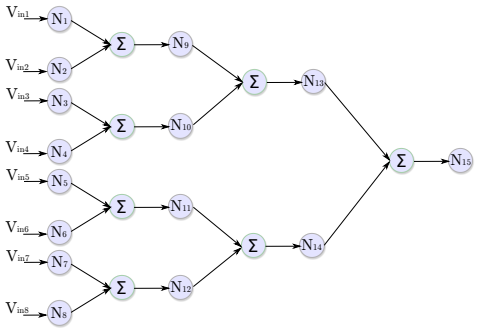


Fig. 9: Simple neuronal network consisting of 15 neurons.

an 8-4-2-1 configuration as shown in Fig. 9. The circuit is simulated for 2 s with a timestep of 0.1 ms. The eight inputs of the system switch 0.1 s apart, as illustrated in the last row of Fig. 10. As shown in the second row of Fig. 10, the output voltage V_{out} of the 15 combined HAs is similar to the voltage of the Spice netlist with 15 instances of the same neuron. The detailed result is presented in Table I.

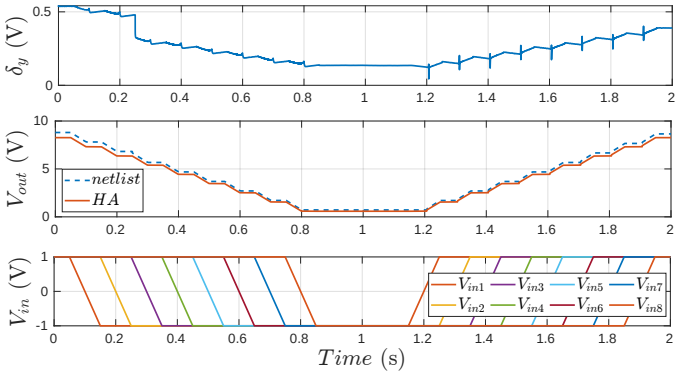


Fig. 10: Comparison of the simulation results from ex.3 (see Fig. 9).

A forth example (ex.4), using a 16-8-4-2-1 configuration, is presented in Table I. This example was simulated with 16 input firing 50 ms apart in a similar fashion to the previous example for a simulation time of 2 s with a timestep of 0.1 ms. Summing up this section, as Table I presents, by using the

TABLE I: Comparison of the Spice netlists and the generated HAs

Model		complexity	Time steps	Runtime (s)	Speedup	$\hat{\delta}_{y,r}$ (%)
ex.1	Netlist	18 tr	10032	3.55	-	-
	HA	-	10034	0.17	20.88	10.56
ex.2	Netlist	54 tr	10045	6.54	-	-
	HA	-	10053	0.25	26.16	3.04
ex.3	Netlist	270 tr	20134	75.16	-	-
	HA	-	20214	2.38	31.57	6.32
ex.4	Netlist	558 tr	20246	253.84	-	-
	HA	-	20404	4.63	54.82	5.9

tr stands for transistors and $\hat{\delta}_{y,r}$ stands for the maximum value of $\delta_{y,r}$, the relative output error normed over the output range

HAs significant speedups are obtained while maintaining an adequate output deviation compared to the Spice netlists.

VII. Conclusion

In this paper, the automatic model abstraction of a neuron with BSIM4 accuracy was examined. As the results show, the obtained model exhibits a significant speedup factor compared to the Spice netlist while maintaining acceptable deviations. Moreover, the speedup scales with the increasing complexity of the circuit, making this approach favorable for the attraction of NNs. Future work will involve controlling the error during the transition of the location and abstracting larger NNs. Additionally, parameter variations of the circuit will be handled with enhanced system descriptions utilizing range arithmetic. As the main target is the formal verification of NNs, future work will as well focus on the hierarchical verification of these circuits and their exhibited behavior.

VIII. Acknowledgment

This paper presents result of the project faveAC funded by the DFG under the project number 286525601.

References

[1] G. Gielen, N. Xama, K. Ganesan, and S. Mitra, "Review of methodologies for pre- and post-silicon analog verification in mixed-signal SOCs," in *2019 Design, Automation Test in Europe (DATE)*, 2019.

[2] W. Zheng, Y. Feng, X. Huang, and H. Mantooth, "Ascend: Automatic bottom-up behavioral modeling tool for analog circuits," in *Circuits and Systems, 2005. ISCAS 2005. IEEE International Symposium On*, pp. 5186–5189 Vol. 5, May 2005.

[3] C. Borchers, "Symbolic Behavioral Model Generation of Nonlinear Analog Circuits," *IEEE Transactions on Circuits and Systems II: Analog & Digital Signal Processing*, vol. 45, no. 10, pp. 1362–1371, 1998.

[4] J.-Y. Chen, S.-W. Wang, C.-H. Lin, C.-N. Liu, Y.-J. Lin, M.-J. Lee, Y.-L. Luo, and S.-Y. Kao, "Automatic behavioral model generator for mixed-signal circuits based on structure recognition and auto-calibration," in *2015 International SoC Design Conference (ISOCC)*, Nov. 2015.

[5] L.-Y. Song, C. Wang, C.-N. J. Liu, Y.-J. Lin, M.-J. Lee, Y.-L. Lo, and S.-Y. Kao, "Non-regression approach for the behavioral model generator in mixed-signal system verification," in *2017 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)*, Oct. 2017.

[6] T. Dang, A. Donzé, and O. Maler, "Verification of analog and mixed-signal circuits using hybrid system techniques," in *Formal Methods in Computer-Aided Design*, pp. 21–36, Springer, 2004.

[7] G. Frehse, C. Le Guernic, A. Donzé, S. Cotton, R. Ray, O. Lebeltel, R. Ripado, A. Girard, T. Dang, and O. Maler, "SpaceEx: Scalable verification of hybrid systems," in *Computer Aided Verification*, pp. 379–395, Springer, 2011.

[8] M. Fränzle, H. Hungar, C. Schmitt, B. Wirtz, B. Becker, W. Damm, Martin, E.-R. Olderog, A. Podelski, and R. Wilhelm, "HLang: Compositional Representation of Hybrid Systems via Predicates," Reports of SFB/TR 14 AVACS 20, SFB/TR 14 AVACS, July 2007.

[9] M. Althoff, A. Rajhans, B. H. Krogh, S. Yaldiz, X. Li, and L. Pileggi, "Formal verification of phase-locked loops using reachability analysis and continuization," *Communications of the ACM*, vol. 56, no. 10, pp. 97–104, 2013.

[10] A. Tarraf and L. Hedrich, "Behavioral Modeling of Transistor-Level Circuits using Automatic Abstractionto Hybrid Automata," in *Design Automation and Testin Europe (DATE)*, (Florence), 2019.

[11] S. Steinhorst and L. Hedrich, "Advanced methods for equivalence checking of analog circuits with strong nonlinearities," *Formal Methods in System Design*, vol. 36, no. 2, pp. 131–147, 2010.

[12] S. Steinhorst and L. Hedrich, "Trajectory-directed discrete state space modeling for formal verification of nonlinear analog circuits," in *International Conference on Computer-Aided Design (ICCAD)*, 2012.

[13] J. Phillips, J. Afonso, A. Oliveira, and L. M. Silveira, "Analog macromodeling using kernel methods," in *IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2003.

[14] O. Stursberg and B. H. Krogh, "Efficient representation and computation of reachable sets for hybrid systems," in *Hybrid Systems: Computation and Control* (O. Maler and A. Pnueli, eds.), (Berlin, Heidelberg), pp. 482–497, Springer Berlin Heidelberg, 2003.

Copyright © AJSE Publishers

A New Approach Method of Crossover Process Based On Genetic Algorithm Using High Dimensional Benchmark Functions

Mustafa TUNAY
Department of Computer Engineering
Istanbul Gelisim University
Istanbul, Turkey
mtunay@gelisim.edu.tr

***Abstract*— The design of the improved genetic algorithm (GA+) is based on a meta-heuristic search for optimization problems. In this paper, the crossover process in the original genetic algorithm is improved. The improvement of the crossover process is renewed by applying two conditions. One of them is keeping the last genes (constant) for each population; the second one is about rotating genes according to the defined range of points between each two selected populations. The improved genetic algorithm (GA+) has the possibility of accelerating local convergence. Therefore, it gets a chance to search for better values globally using these conditions. All processes in the improved genetic algorithm have been represented in this paper. The performance of the proposed algorithm is evaluated using 7 benchmark functions (test functions) on different dimensions. Ackley function, Rastrigin function and Holzman function are multi-modal minimization functions; Schwefel 2.22 function, Sphere function, Sum Squares function and Rosenbrock function are uni-modal minimization functions. These functions are evaluated by considering cases that are minimized by having a set of dimensions as 30, 60, and 90. Additionally, the performance of the GA+ is compared with the performance of comparative optimization algorithms (meta-heuristics). The comparative results have shown the performance of the GA+ that performs much better than others for optimization functions.**

Keywords—Benchmark Functions, Genetic Algorithms, Improved Crossover Process, Metaheuristic Search.

I. INTRODUCTION

In this section, a literature review of optimization algorithms [1-5] is given. The research-based on evolutionary algorithms [6-8] including differential evaluation [9,10] and especially genetic algorithms [11-13] have been analyzed. The design of the high accuracy genetic algorithm for solving dimensional optimization problems is noticed.

Optimization issues usually need to use mathematical algorithms for seeking out a good solution iteratively in analytical solutions. In this spirit, different optimization strategies have been designed for finding a good solution. There is an optimization method which is simultaneous perturbation stochastic approximation (SPSA) for multivariate optimization. This optimization technique finds a good solution in issues such as feedback control, simulation optimization, image processing, adaptive modeling, estimation of distribution algorithms, atmospheric modeling. The proposed method uses gradient approximation in any case of the dimension of the optimization problem. The SPSA method decreases especially in problems that need to be optimized due to the cost of optimization solutions. More details are referred to in [14].

Many optimization problems [15-17] are primarily to find the best solution within their specific ranges. This kind of optimization problem usually refers to the best solution functions for solving using applied mathematics functions. Optimization problems include searching "the best solution" from the values of some objective function ranges for different types of objective optimization. The solutions of nonlinear optimization acquire great importance.

The mathematical model [18,19] is applied to many fields as economics, industry, computer science, game theory, artificial intelligence (AI), and many other areas in the real world. Many mathematicians studied to explore a wide range of complex tasks and focused on systems with multiple factors that they interact in nonlinearly. They obtained two cases as a result of this study. They are major effects of co-adaptation and co-evolution. Thus, the mathematical model describes how to change the traditional process of mathematical genetics.

The traditional process of optimization techniques is applied for obtaining the solution of different optimization problems. Traditional computational intelligent systems are based on private "internal" cognitive and computational processes. However, the traditional optimization techniques are based on finding the derivative of objective functions that means a locally optimized solution. Additionally, there are many other various problems in the proposed techniques that do not fare well for finding the global optimal solution of the functions. Many multi-objective applications of evolutionary algorithms have found increasing applications in the domain of data mining problems.

A new heuristic approach is applied to minimize nonlinear and non-differentiable continuous space functions [20]. The proposed algorithm selects the difference between two vectors randomly. Additionally, the proposed algorithm perturbs an existing vector in chosen population vectors. The perturbation is done for every population vector. The proposed method is demonstrated to converge faster for multi-objective optimization.

Genetic algorithms provide complex adaptive systems for

Copyright © AJSE Publishers

economic theory using machine learning methods. Adaptation is a biological process that is to survive in environments confronting organisms that evolve by rearranging genetic material. Several scientists studied to seek out a solution for nonlinearity. Holland presents a mathematical model for complex interactions [21].

One of the widely used adaptive heuristic search algorithm used for multi-objective optimization. The proposed algorithm relies on the evolutionary conception of natural selection [22]. Natural evolution is randomly generated by individuals from a population.

There is also a viable new approach to stochastic combinatorial optimization which is inspired by the behavior of the ant. The proposed algorithm's main features are constructive greedy heuristic, distribution of computation, and positive feedback. Firstly, the greedy heuristic finds acceptable solutions for the search process. Secondly, the distribution of computation avoids premature convergence. Finally, the positive feedback explains the fast discovery of the best solutions. The proposed methodology is applied in practical problems to solve a set of problems for the robustness of the approach [23].

The genetic algorithm builds using the differing qualities of a new strategy in the detection of epileptic seizures from electroencephalogram (EEG). The detection of epileptic seizures from electroencephalogram transforms packets including energy, entropy, kurtosis, and skewness using discrete wavelet for creating features of signals. Clinical EEG data is commonly used in the experiment of epileptic and normal subjects. The proposed is improved GA with a support vector machine (GA-SVM). This means it is a new method for finding a good solution using a hybrid approach with wavelet packet decomposition [24].

The design of the tactical berth allocation problem (TBAP) is a biased random key in the modification of the genetic algorithm. It contains both the minimization of the housekeeping expenses; the first one is from the transshipment compartment streams in the middle of ships, the second one is about the amplification of the aggregate estimation of the quay crane profiles doled out of the ships. The acquired results for handling the TBAP have demonstrated that the proposed calculation is appropriate to proficiently take care of this issue [25].

A new structured population approach, which is a hierarchy of hypercube is represented as the population of GA. This approach generally leads to a more superior performance than palmitic GA [26]. Additionally, this research does not build sub-populations that are based on the information of the genes of individuals. The structure of subpopulations could help to achieve better performance and a more efficient searching strategy. The proposed approach can build the structure of a population by dividing the searching space.

The improved artificial chromosomes with a genetic algorithm (ACGA) is a new tendency for search optimization. The proposed algorithm has been applied to real-world problems successfully for solving scheduling problems. However, ACGA can not perform well in some scheduling problems. It does not consider variable interactions if sequence-dependent setup times are considered particularly. Thus, the previous one will improve variable interactions to influence the processing time. The improvement of ACGA is successfully applied to single machine scheduling problems. This improvement of ACGA is improved with a bi-variate probabilistic model. This is called the design of ACGA II. It includes some heuristics and local search algorithms and variable neighborhood search (VNS) [27]. The proposed method is successfully demonstrated to solve single machine scheduling problems with sequence-dependent setup times for the dating environment.

Many real-world optimization problems are using mathematical algorithms. It seeks an iterative solution because the function or the constraints of the objective problem can be improved over time. If these cases are undefined past in the optimization process, we are called dynamic for these types of problems. There are some difficulties in optimizing dynamic environments with the goal that the calculations for rationalization in these situations would be to use some systems keeping in mind the ultimate objective of overcoming difficulties. There are many algorithms for optimization problems.

There is a new technique for crossover operator. This is called an inversed bi-segmented Average Crossover (IBAX). It improves the offspring generation of the genetic algorithm for variable and numerical optimization problems. It attempts to come into view with a new mating scheme that in generating new offspring is under the crossover function using the IBAX operator. It has a more efficient and optimization solution for variable minimization on premature convergence problem using GA [28, 29].

There is another optimization problem for the bin packing problem (BPP). GA is one way to solve the Bin Packing (BPP) problem. The goal of BPP is to minimize the number of containers used by maximizing their content. A combination of BPP and GA is applied to the printed paper in digital printing. GA improvement is enhanced by random crossover and dynamic mutation. With this application, GA performance in the case of BPP can solve the problem of premature convergence and maximize print distribution [30].

The genetic algorithm is improved by a new technique for planning collision-free paths with static obstacles. The improved genetic algorithm is realized in the crossover process. This process has an important rule about the fitness value of the progeny. It should compare with the fitness value of the parents. Thus, the parents (chromosome) of the best fitness value will be needed for the mutation process. Meanwhile, the worse fitness value will be excluded from the best fitness value completely. This technique is applied to the intelligent navigation system for autonomous mobile robots such as differential drive mobile robots [31].

As it was mentioned above different multi-objective evolutionary algorithms such as genetic algorithms (GAs) and differential evaluation algorithms have been designed. These algorithms have found many practical applications. These algorithms have been applied in optimization issues successfully to solve many difficult optimization search problems. Many improvements have been done to develop optimization to search for the best solution to the problems.

In this paper, the improved genetic algorithm includes the selection process, the improved crossover process, and the mutation process. The crossover process in the original genetic algorithm is improved with a new technique that is

Copyright © AJSE Publishers

keeping the last genes for each population and applying processing as rotating genes according to the assigned random numbers. These numbers define the range of genes for all populations. After that, they are replaced one by one between the two selected populations. It will continue between the selected populations (groups of two) with the same specific locations in them. Thus, the improved genetic algorithm (GA+) has the possibility of accelerating for local convergence and it gets a chance to search for better values globally. For all processes in the improved genetic algorithm have been represented in this paper. Moreover, the performance of the GA+ was tested on some benchmark functions. The performance of the GA+ has shown much better convergence rates than comparative optimization algorithms.

The organization of this article is included as follows: Section 2 describes the improved genetic algorithm (GA+). Section 3 describes the test functions. The information of seven benchmark functions are given. The performance of proposed algorithm is tested on seven test functions and is compared with metaheuristic algorithms on different dimensional functions. Section 4 describes the conclusion in this article.

II. Related Work

In this study is mainly presented the improvement of the crossover process that is renewed by applying two conditions. One of them is keeping the last genes (constant) for each population. The second one is rotating genes according to the defined range of points (the assigned two random numbers). The details regarding the pseudo-code of the GA+ are illustrated in Algorithm 1.

```
Start
   Initialization population (Ps)
      Done= false
While not done Do
Calculate the fitness of each individual population
PS = PS+1
   Selection (PS -1)
      Modification Crossover Cp
    *Keeping last gene constant for each population
    *Rotating genes according to the defined range of points
     *Rapleced them with other one
      Mutation Mp
   Done = Optimization
Stop?
End While
Display `all best solution`
End
```

Algorithm 1. The pseudo-code of the improved genetic algorithm

The GA+ is based on the meta-heuristic search for optimization problems. The main steps of the GA+ consist of selection process, improved crossover process, and mutation process. All processes in the improved genetic algorithm have been designed in Figure 1.

The improvement of the crossover process is explained in all details in the next section.

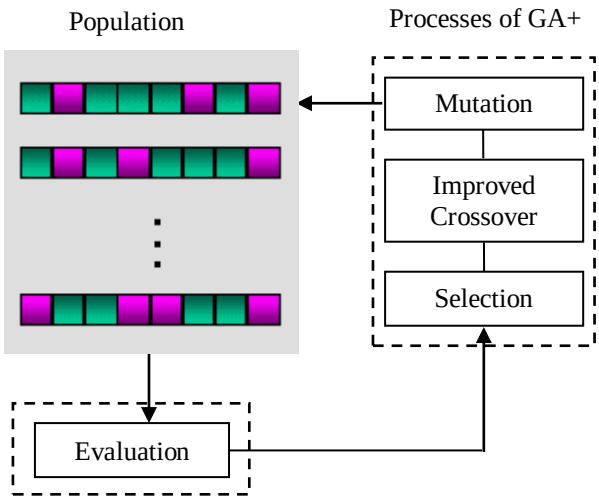


Fig. 1. The flow-chart of the GA+

2.1 *Selection Process*

The aim of the selection process selects good solutions and eliminates bad solutions in a population. This process is a genetic operator used in the improved genetic algorithms to select potentially useful solutions for recombination. This process uses the "fitness function" for assigning fitness to possible solutions or chromosomes. It is in fitness proportionate selection, as in all selection methods.

The fitness function is used to associate a probability of selection with each chromosome. If it is the fitness of individual i in the population, its probability of being selected is:

$p_i = \frac{f_i}{\sum_{j=1}^{N} f_j}$ where N is the number of individuals in the population. (1)

2.2 *Improved Crossover Process*

The crossover process is a genetic operator that combines two individuals. It produces a new population according to the specific processing rules. There are several kinds of crossover operators such as two-point crossover, multi-point crossover, and uniform crossover.

Firstly, the GA+ begins with the generation of the population and all populations are generated randomly. All these populations are matched (the selected groups of two; population 1 and population 2) to each other randomly. The two random numbers are generated randomly between the length of the population and are determined by the range of location of columns for each population. That is, the assigned two random numbers are defined for the range of genes. After all, these genes in the specified range of location of columns are rotated, then these rotated genes are replaced between the selected two populations one by one with a defined range of location of the columns that are formed in the same way. After that, all the last constant genes in the population will be replaced between the selected ones one by one according to the same location of columns. It will keep going between the selected

Copyright © AJSE Publishers

populations (groups of two) with the same specific locations in them. The detail regarding the visualization of the new approach of the improvement of the crossover process is shown in Figure 2.

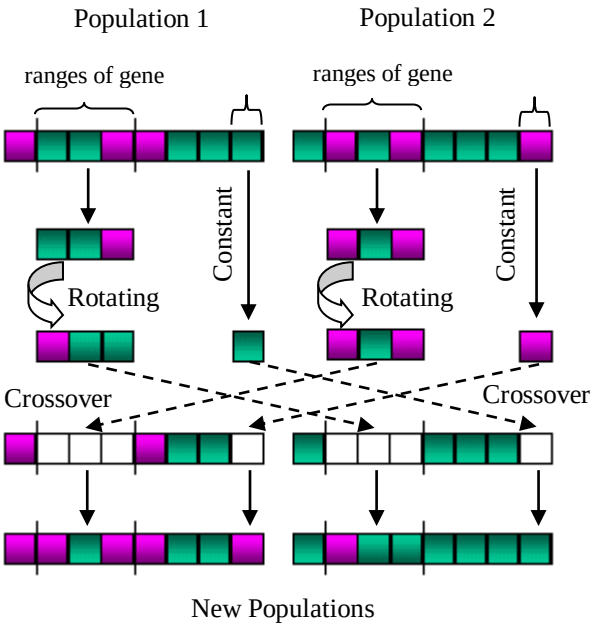


Fig. 2. The new approach of the improved crossover process for two new populations

Secondly, there is an important point about defining a value of the specific number in the modification of the crossover process for creating new chromosomes in a population. If the length of the population is greater than the value of the defined specific number, the population is divided into new pieces of chromosomes according to the value of the defined specific number and for each chromosome in the population have the same size (length for each chromosome). Provided that the last genes for each new chromosome in the population should be kept constant. The two random numbers are generated randomly between the length of chromosomes in the population and are determined the range of location of columns for each chromosome in the population. That is, the assigned two random numbers are defined for the range of genes for each chromosome in the population. After all, these genes in the specified range of location of columns are rotated, these rotated genes are replaced between the selected two populations (for each chromosome in population) one by one with a defined range of location of the columns. After that, the last genes for each chromosome in the population will be replaced between the selected ones one by one according to the same location of columns (the range of genes in the chromosome). It will keep going between the selected populations (groups of two) with the same specific locations in them.

The detail regarding the visualization of the new approach of the improvement of the crossover process for a piece of chromosomes in a population is shown in Figure 3.

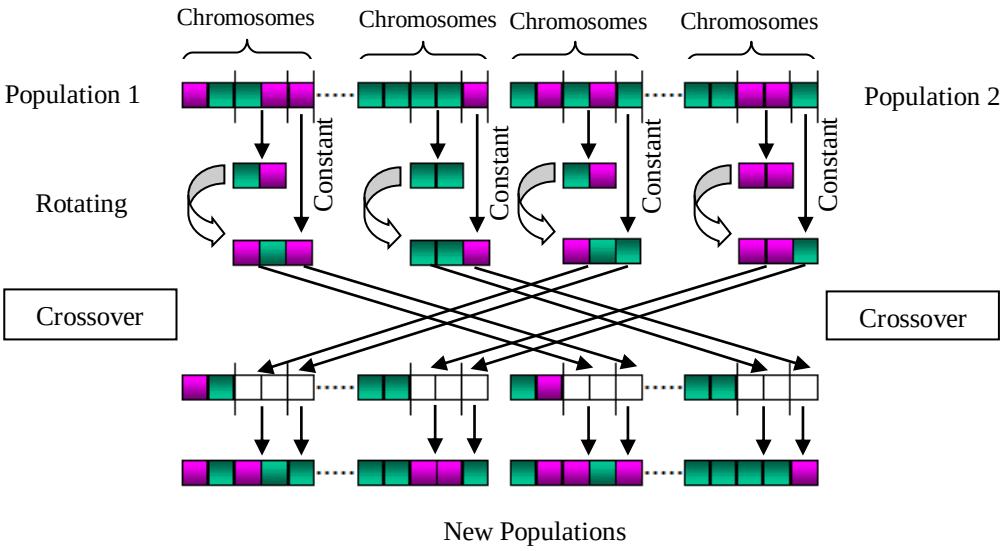


Fig. 3. The new approach of improved crossover process for a piece of chromosomes in a population

2.3 *Mutation Process*

This process occurs at each position in a bit string with a specific probability. This specific probability is generally defined between 0.1 or less according to the length of the population. The principle of this process assigned anyone a random number. According to the assigned value of the number is defined which location of the column in population. If its value is "1", it will be "0"; otherwise. The details regarding the visualization of the mutation process are shown in Figure 4.

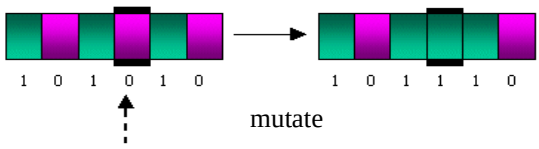


Fig. 4. Mutation process

Copyright © AJSE Publishers

However, there is an important point for defining the length of the population. If the length of the population is greater than the value of the defined specific number, the population is divided into new pieces of chromosomes according to the value of the defined specific number and each chromosome in the population has the same size (length for each chromosome). The principle of this process is to assign a random number in the mutation process. According to the assigned value, the number is defined for each chromosome in the population, that is, it will be the locations of the column in chromosomes for each population. If its value is "1", it will be "0"; otherwise.

III. Test Functions

The performances of the improved genetic algorithm (GA+) was implemented on Matlab-Simulink Version 2017. The GA entails setting several parameter values. The primary parameters of genetic algorithms included as crossover rate (0.60) and mutation rate (0.1). The features of the hardware and software tools of the computer used are as follows; CPU: Intel (R) Core (TM) i3-4005U M, SPEED: 1.70 GHz – x64, RAM: 4.00 GB and OS: Microsoft Windows 8.1. The improved genetic algorithm (GA+) performed on specific benchmark functions (test functions).

They consists of seven optimization test functions, namely, Ackley function (F1), Rastrigin function (F2), Schewefel 2.22 function (F3), Sphere function (F4), Sum Squares function (F5), Holzman function (F6) and Rosenbrock function (F7). For more information about these benchmark functions including implementation codes and more, we refer to the reader; http://benchmarkfcns.xyz/fcns.

Ackley function, Rastrigin function and Holzman function are multi-modal minimization functions; Schwefel 2.22 function, Sphere function, Sum Squares function and Rosenbrock function are uni-modal minimization functions. The properties of these functions are briefly shown in Table 1 as equation of test functions and range. The performances of the GA+ is evaluated with the use of these functions in the next section.

3.1 The Performance of GA+ on Test Functions

The performance of proposed algorithm was compared on seven test functions in this section. The GA+ was evaluated to minimize functions having the set of dimensions as 30, 60, and 90. The optimization experiments of the proposed algorithm (GA+) was performed on the three different dimensions for the 7 benchmark optimization functions. The parameters for the algorithm was the set as follows: size of population = 50, iterations = $5x10^2$ and 10^3 and dimension (D)= 30, 60 and 90.

TABLE 1
THE GA+ PERFORMED ON TEST FUNCTIONS FOR DIMENSIONS AS 30, 60 AND 90

Test Functions		30 Dimension		60 Dimension		90 Dimension	
		GA+		GA+		GA+	
		500 iterations	1000 iterations	500 iterations	1000 iterations	500 iterations	1000 iterations
F1	*Best*	1.28e-13	1.28e-13	1.14e-13	1.14e-13	1.14e-13	1.17e-13
	Mean	6.45e-06	5.59e-10	1.70e-12	2.23e-12	2.05e-08	1.05e-11
	Std.	2.75e-05	2.05e-09	6.50e-12	9.25e-12	8.88e-08	1.40e-11
F2	*Best*	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00
	Mean	1.60e-04	4.75e-06	8.88e-16	0.00e+00	3.65e-15	0.00e+00
	Std.	5.80e-04	1.75e-05	3.89e-15	0.00e+00	1.59e-14	0.00e+00
F3	*Best*	4.44e-13	1.25e-12	4.73e-27	4.73e-27	7.09e-27	7.09e-027
	Mean	2.92e-05	2.45e-05	8.29e-23	5.15e-27	6.19e-17	4.67e-017
	Std.	8.60e-05	8.06e-05	1.79e-22	0.00e+00	1.39e-16	2.18e-016
F4	*Best*	7.49e-28	7.49e-28	1.34e-27	1.18e-27	1.94e-27	1.77e-27
	Mean	7.70e-06	1.97e-07	3.13e-22	1.35e-24	1.51e-16	3.56e-18
	Std.	2.45e-05	8.08e-07	1.25e-21	0.00e+00	6.44e-16	1.49e-17
F5	*Best*	2.18e-21	3.44e-26	1.39e-25	1.39e-25	3.32e-25	3.15e-25
	Mean	2.89e-04	2.15e-05	8.40e-22	2.65e-22	1.59e-15	2.32e-18
	Std.	8.33e-04	8.19e-05	3.66e-21	6.90e-22	4.30e-15	1.50e-17
F6	*Best*	7.19e-54	1.31e-53	1.10e-53	1.10e-53	2.49e-53	2.49e-53
	Mean	1.59e-08	7.29e-16	2.88e-37	3.83e-44	3.47e-25	5.15e-31
	Std.	6.95e-08	2.40e-15	0.00e+00	0.00e+00	0.00e+00	0.00e+00
F7	*Best*	2.82e+01	2.82e+01	5.80e+01	5.76e+01	8.83e+01	2.82e+01
	Mean	2.88e+01	2.88e+01	5.86e+01	5.86e+01	8.87e+01	8.86e+01
	Std.	1.92e-01	1.45e-01	2.60e-01	3.59e-01	2.92e-01	2.33e-01

The performance of the improved genetic algorithm was evaluated the same dimensions (30, 60, 90) and as well using a varying number of iterations in solving seven benchmark functions. The properties of a benchmark functions having standard parameters were implemented on Matlab program. These functions are summarized as the best, the mean, the standard deviation and evaluated over successful 100 runs. The performance of the improved proposed algorithm is shown in Table 1.

Copyright © AJSE Publishers

The algorithm, which finds the best solution and solves optimization problems is designed. The design of the improved crossover process in a convergence state help to the best position to jump out of possible local optimal solution to further increase the performance of proposed algorithm. Thus, the search strategy in the proposed algorithm has proven to be a success global optimal solution, convergence optimal solution, allows speeding the learning of the system with faster convergence rates for all these optimization problems.

The performance of the GA+ is also compared with the performance of meta-heuristic algorithms in the next section.

3.2 Comparison between GA+ and Metaheuristic Algorithms

This section presents the comparison of the performance of the proposed algorithm with meta-heuristics algorithms. Examples of meta-heuristic algorithms include the ant colony optimization (ACO), the bat algorithm (BAT), particle swarm optimization (PSO), ant lion optimizer (ALO), krill herd (KH), monarch butterfly optimization (MBO) and moth-flame optimization (MFO). Other metaheuristics have also been developed based on the evolutionary theory including differential evolution (DE). The above meta-heuristics are classified as stochastic optimization techniques. All algorithms were evaluated by considering the cases in which functions having the set of dimensions as 30, 60, 90 for 50 iterations and averaged over 100 experimental runs. The population size is also set to 50.

TABLE 2
THE GA+ COMPARED WITH METAHEURISTIC ALGORITHMS

Test Functions	D	ACO	BAT	DE	PSO	GA+
F1	*30*	1.85E+001	1.99E+001	1.87E+001	1.87E+001	**7.32e-002**
	60	1.90E+001	1.99E+001	1.90E+001	1.90E+001	**2.87e-005**
	90	1.91E+001	1.99E+001	1.91E+001	1.91E+001	**1.98e-004**
F2	*30*	1.63e+002	4.34e+002	1.73e+002	1.73e+002	**2.89e+000**
	60	3.74e+002	9.33e+002	3.99e+002	4.00e+002	**4.90e-008**
	90	6.03e+002	1.44e+003	6.41e+002	6.31e+002	**7.67e-005**
F3	*30*	1.13E+002	2.95E+012	5.38E+001	1.14E+002	**2.39e-001**
	60	2.48E+002	2.29E+028	1.71E+002	2.49E+002	**3.00e-007**
	90	3.88E+002	6.75E+043	2.97E+002	3.89E+002	**5.91e-004**
F4	*30*	1.63E+002	1.67E+002	2.79E+001	5.12E+001	**9.21e-003**
	60	3.76E+002	3.91E+002	1.74E+002	2.13E+002	**6.14e-006**
	90	6.02E+002	6.19E+002	3.80E+002	4.29E+002	**5.65e-004**
F5	*30*	9.37E+003	9.27E+003	1.29E+003	2.30E+003	**1.79e-001**
	60	4.39E+004	4.29E+004	1.65E+004	1.62E+004	**1.29e-007**
	90	1.03E+005	1.03E+005	5.51E+004	5.06E+004	**3.70e-003**
F6	*30*	4.19E+005	4.19E+005	2.51E+004	6.73E+004	**1.80e-002**
	60	2.24E+006	2.13E+006	6.43E+005	9.31E+005	**6.05e-012**
	90	5.50E+006	5.24E+006	2.65E+006	3.19E+006	**1.79e-009**
F7	*30*	1.06E+008	2.32E+008	1.59E+007	2.25E+007	**3.04e+001**
	60	5.87E+008	5.97E+008	2.12E+008	2.11E+008	**5.89e+001**
	90	1.00E+009	9.99E+008	5.81E+008	7.39E+008	**8.89e+001**

The performances of all algorithms were compared using the same common parameters [32] and the same number of iterations for seven benchmark functions. The performance of the GA+ was compared with a selected collection of comparative algorithms that have been evaluated. The comparative algorithms are ACO, BAT, DE, and PSO. The comparative results demonstrated the performance of the GA+ which is also much better than the selected collection of the meta-heuristics algorithms (ACO, BAT, DE, and PSO) for seven standard benchmark functions. The proposed algorithm obtained 7.32e-02, 2.87e-05 and 1.98e-04 using Ackley function; 2.89e+00, 4.90e-08 and 7.67e-05 using Rastrigin function; 2.39e-01, 3.00e-07 and 5.91e-04 using Schwefel 2.22 function; 9.21e-03, 6.14e-06 and 5.65e-04 using Sphere function; 1.79e-01, 1.29e-07 and 3.70e-03 using Sum Squares function; 1.80e-02, 6.05e-12 and 1.79e-09 using Holzman function; 3.04e+01, 5.89e+01 and 8.89e+01 using Rosenbrock function having the set of dimensions as 30, 60 and 90 respectively. The best mean for each function is marked in bold and all details are shown in Table 2.

The performance of the GA+ was compared with other metaheuristic algorithms that have been evaluated. The comparative algorithms are ALO, KH, MBO, and MFO. The comparative results demonstrated the performance of the GA+ which is also much better than the selected collection of the other meta-heuristic algorithms for seven benchmark functions. The best mean for each function is marked in bold and all details are shown in Table 3.

The performance of the GA+ was compared with the performance of four comparative optimization algorithms, namely, GA, PSO, GAPSO, and the improved genetic particle swarm optimization algorithm (IGAPSO) using Rastrigin function and Sphere function having the set of dimension as 30. They are used here with the same parameters [33].

Copyright © AJSE Publishers

TABLE 3
THE GA+ COMPARED WITH OTHER METAHEURISTIC ALGORITHMS

Test Functions	D	ALO	KH	MBO	MFO	GA+
F1	*30*	1.37e+001	4.84e+000	1.41e+001	1.85e+001	**7.32e-002**
	60	1.53e+001	7.16e+000	1.60e+001	2.01e+001	**2.87e-005**
	90	1.62e+001	7.80e+000	1.57e+001	2.04e+001	**1.98e-004**
F2	*30*	1.29e+002	1.06e+001	9.96e+001	2.85e+002	**2.89e+000**
	60	3.79e+002	2.78e+001	2.37e+002	7.37e+002	**4.90e-008**
	90	6.64e+002	5.08e+001	4.47e+002	1.22e+003	**7.67e-005**
F3	*30*	1.06e+002	1.14e+001	5.24e+001	4.66e+002	**2.39e-001**
	60	1.31e+017	2.45e+014	1.44e+002	1.13e+017	**3.00e-007**
	90	1.57e+031	3.56e+027	2.74e+002	1.37e+032	**5.91e-004**
F4	*30*	1.57e+001	4.63e–001	6.38e+001	6.57e+001	**9.21e-003**
	60	5.10e+001	4.75e+000	1.93e+002	2.70e+001	**6.14e-006**
	90	8.84e+001	9.04e+000	3.58e+002	4.97e+002	**5.65e-004**
F5	*30*	7.56E+002	4.21E+001	5.24E+003	3.09E+003	**1.79e-001**
	60	5.44E+003	5.33E+002	2.51E+004	2.63E+004	**1.29e-007**
	90	1.39E+004	1.55E+003	6.47E+004	7.70E+004	**3.70e-003**
F6	*30*	3.23E+003	-2.21E+008	1.82E+005	8.48E+004	**1.80e-002**
	60	3.33E+004	-6.49E+012	1.08E+006	1.14E+006	**6.05e-012**
	90	1.06E+005	-1.65E+017	2.59E+006	3.76E+006	**1.79e-009**
F7	*30*	1.89E+006	8.69E+003	5.85E+007	4.74E+007	**3.04e+001**
	60	9.56E+006	2.28E+004	2.22E+008	3.61E+008	**5.89e+001**
	90	2.09E+007	3.85E+004	3.07E+008	7.55E+008	**8.89e+001**

TABLE 4
THE GA+ COMPARED WITH GA, PSO, GAPSO AND IGAPSO ALGORITHMS ON 30 DIMEMSION

30D	Iterations	Rastrigin Function Mean	Iterations	Sphere Function Mean
GA+	200	**1.83e-002**	200	**1.95e-005**
GA	401	8.596e+001	265	8.89e-002
PSO	373	1.200e+002	202	6.02e-003
GAPSO	361	9.05e+000	197	4.20e-004
IGAPSO	342	9.01e+000	186	4.12e-004

The comparative results suggest that the overall convergence rates of the GA+ still perform much better than others for Rastrigin function and Sphere function and obtained 1.83e-002 and 1.95e-005 respectively for $2x10^2$ iterations. Moreover, the best mean for each function is marked in bold and all details are shown in Table 4.

IV. CONCLUSIONS

The improved genetic algorithm (GA+) was designed for evaluation of the GA+ with the aim of the number of iterations. The number of iterations has shown obtained optimal solutions and convergence optimal solutions for optimization problems. The proposed algorithm recommended using a population of 50 as well as using a varying number of iterations ranging from 50, $5x10^2$ and 10^3 in solving specific benchmarking functions. The proposed algorithm obtained 7.32e-02, 2.87e-05 and 1.98e-04 using Ackley function; obtained 2.89e+00, 4.90e-08 and 7.67e-05 using Rastrigin function; obtained 2.39e-01, 3.00e-07 and 5.91e-04 using Schwefel 2.22 function; obtained 9.21e-03, 6.14e-06 and 5.65e-04 using Sphere function; obatined 1.79e-01, 1.29e-07 and 3.70e-03 using Sum Squares function; obtained 1.80e-02, 6.05e-12 and 1.79e-09 using Holzman function; obtained 3.04e+01, 5.89e+01 and 8.89e+01 using Rosenbrock function having dimensions as 30, 60 and 90 respectively for 50 iterations. Secondly, the proposed algorithm obtained 6.45e-06, 1.70e-12 and 2.05e-08 using Ackley function; obtained 1.60e-04, 8.88e-16 and 3.65e-15 using Rastrigin function; obtained 2.92e-05, 8.29e-23 and 6.19e-17 using Schwefel 2.22 function; obtained 7.70e-06, 3.13e-22 and 1.51e-16 using Sphere function; obtained 2.89e-04, 8.40e-22 and 1.59e-15 using Sum Squares function; obtained 1.59e-08, 2.88e-37 and 3.47e-25 using Holzman function; obtained 2.88e+01, 5.86e+01 and 8.87e+01 using Rosenbrock function for $5x10^2$ iterations. Finally, the proposed algorithm obtained 5.59e-10, 2.23e-12 and 1.05e-11 using Ackley function; obtained 4.75e-06, 0.00e+00 and 0.00e+00 using Rastrigin function; obtained 2.45e-05, 5.15e-27 and 4.67e-17 using Schwefel 2.22 function; obtained 1.97e-07, 1.35e-24 and 3.56e-18 using Sphere function; obtained 2.15e-05, 2.65e-22

Copyright © AJSE Publishers

and 2.32e-18 using Sum Squares function; obtained 7.29e-16, 3.83e-44 and 5.15e-31 using Holzman function; obtained 2.88e+01, 5.86e+01 and 8.86e+01 using Rosenbrock function for 10^3 iterations.

The improvement of the crossover process was renewed by applying two conditions. That is the method existing in the cross-over process renewed in the original GA. Briefly, the proposed algorithm keeps its original form without any external additions. For this reason, we believe that the GA+ is not very popular among researchers who compare with today's novel optimization algorithms in this study. However, the performance of GA+ was compared with many metaheuristic optimizers, including ACO, BAT, PSO, ALO, KH, MBO, MFO, DE, GA, GAPSO and IGAPSO; the set of comparative optimization algorithms as well as a collection of 11 algorithms. The comparative results included the best mean for the obtained optima. All those results showed an outstanding performance of GA+ in the majority of the evaluation cases in this paper.

REFERENCES

[1] R. Abiyev and M. Tunay, "Optimization of High Dimensional Functions through Hypercube Evaluation," *Computational Intelligence and Neuroscience*, volume 2015, pp. 1-11, 2015.

[2] M. Tunay, "Evolutionary Search Algorithm Based on Hypercube Optimization For High-Dimensional Functions," *International Journal of Computational and Experimental Science and Engineering (IJCESEN)*, 6(1), 42 – 62, 2020.

[3] R. Abiyev and M. Tunay, "Optimization Search Using Hypercubes," 2020 4th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), Istanbul, Turkey, October 22-24, 2020.

[4] M. Tunay, "A New Design of Metaheuristic Search Called Improved Monkey Algorithm Based on Random Perturbation for Optimization Problems," *Scientific Programming*, volume 2021, pp. 1-14, 2021.

[5] R. Abiyev and M. Tunay, "Experimental Study of Specific Benchmarking Functions for Modified Monkey Algorithm," *Procedia Computer Science*, volume 102, pp. 595-602, 2016.

[6] J. Xiu , Q. He, Z. Yang and C. Liu, "Research on a multi-objective constrained optimization evolutionary algorithm," 2016 4th International Conference on Cloud Computing and Intelligence Systems (CCIS), Beijing, August 17-19, 2016.

[7] A. Slowik & H. Kwasnicka, "Evolutionary algorithms and their applications to engineering roblems," *Neural Comput & Application*, 32, pp. 12363–12379, 2020.

[8] F. Gu, K. Li and Y. Liu, "A Hybrid Evolutionary Algorithm for Solving Function Optimization Problems," 2016 12th International Conference on Computational Intelligence and Security (CIS), Wuxi, December 16-19, 2016.

[9] A. W. Mohamed, H. Z. Sabry, M. Khorshid, "An alternative differential evolution algorithm for global optimization, " *Journal of Advanced Research*, 3(2), 149–165, 2012.

[10] A. Draa, K. Chettah and H. Talbi, "A compound sinusoidal differential evolution algorithm for continuous optimization," *Swarm and Evolutionary Computation*, 50, pp. 100450, 2019.

[11] L. Deng, P. Yang and W. Liu, "An Improved Genetic Algorithm," 2019 IEEE 5th International Conference on Computer and Communications (ICCC), Chengdu, China, December 06-09, 2019.

[12] A. Lambora, K. Gupta and K. Chopra, "Genetic Algorithm-A Literature Review," 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-Con), Faridabad, India, Feb. 14-16, 2019.

[13] D. E. Goldberg, *Genetic Algorithms in Search Optimization and Machine Learning*. Boston: Addison–Wesley Publishing Company, 1989.

[14] J. Spall, "An overview of the simultaneous perturbation method for efficient optimization," *Johns Hopkins APL Technical Digest*, 19: 482-492, 1998.

[15] X. S. Yang, *Computational Optimization*. Methods and Algorithms, 2011.

[16] M. Jamil & X. S. Yang, "A Literature Survey of Benchmark Functions For Global Optimization Problems," *International Journal of Mathematical Modelling and Numerical Optimisation*, 4, 2014.

[17] S. Pietro, M. W. Oliveto, T. Weise, B. Wróbel and A. Zamuda, "Black-Box Discrete Optimization Benchmarking," Workshop at the 2018 Genetic and Evo. Computation Conference (GECCO), Japan, July 15-19, 2018.

[18] H. Eren, "Mathematical Methods of Optimisation," *Handbook of Measuring System Design*, volume 3, 452-455. 2005.

[19] J. M. Powers and M. Sen, *Mathematical Methods in Engineering*. Cambridge University Press, 2015.

[20] R. Storn and K. Price, "Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," *Journal of Global Optimization*, 11, 341-359, 1997.

[21] J. Holland, *Adaptation in natural and artificial systems*. University of Michigan Press, Extended new Edition, MIT Press, Cambridge, 1975.

[22] M. Tunay and R. Abiyev, "Hybrid Local Search Based Genetic Algorithm and Its Practical Application," *International Journal of Soft Computing and Engineering*, 5(2):21-27, 2015.

[23] M. Dorigo, V. Maniezzo and A. Colorni, "The ant system: optimisation by a colony of cooperative agents," *IEEE Transactions on System Man Cybernet*, 26, 29-41, 1996.

[24] R. Dhiman and J. S. Priyanka, "Genetic algorithms tuned expert model for detection of epileptic seizures from EEG signatures," *Applied Softcomputing*, 19, 8-17, 2014.

[25] E. Lalla-Ruiz, J. L. González-Velarde, B. Melián-Batista, J. M. Moreno-Vega, "Biased random key genetic algorithm for the tactical berth allocation problem," *Applied Softcomputing*, 22, 60-76, 2014.

[26] M. A. Belal and M. H. Haggag, "A structured-population genetic-algorithm based on hierarchical hypercube of genes expressions," *International Journal of Computer Applications*, 64(22), 5-18, 2013.

[27] S. H. Chen, M. C. Chen and Y. C. Liou, Artificial chromosomes with genetic algorithm 2 (ACGA2) for single machine scheduling problems with sequence-dependent setup times," *Applied Softcomputing*, 17, 167-175, 2014.

[28] A. J. Delima, A. Sison and R. Medina, "A modified genetic algorithm with a new crossover mating scheme," *Indonesian Journal of Electrical Engineering and Informatics*, 7, pp. 165-181, 2019.

[29] D. Chaudhary, A. K. Tailor, V. P. Sharma and S. Chaturvedi, "HyGADE: Hybrid of Genetic Algorithm and Differential Evolution Algorithm," 2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Kanpur, India, July 6-8, 2019.

[30] H. F. Sulaiman, B. T. Sartana and U. Budiyanto, "Genetic Algorithm With Random Crossover and Dynamic Mutation on Bin Packing Problem," 2019 6th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), Bandung, Indonesia, September 18-20, 2019.

[31] N. S. Utami, A. Jazidie and R. E. A. Kadier, "Path Planning for Differential Drive Mobile Robot to Avoid Static Obstacles Collision using Modified Crossover Genetic Algorithm," 2019 International Seminar on Intelligent Technology and Its Applications (ISITIA), Surabaya, Indonesia, August 28-29, 2019.

[32] W. A. H. M. Ghanem and A. A. Jantan, "Cognitively Inspired Hybridization of Artificial Bee Colony and Dragonfly Algorithms for Training Multi-layer Perceptrons," *Cogn Comput*. 10, 1096–1134, 2018.

[33] H. Zhang, S. Li and X. Liu, "Research on Function Optimization Based on Improved Genetic Particle Swarm Optimization," *Journal of Physics: Conference Series*, 1549, 042133, 2020.

Copyright © AJSE Publishers

Feasibility of Satellite Sabotage via TrojanCube

Andrew T. Rath
Capitol Technology University
andrew.t.rath@gmail.com

Alex Antunes
Capitol Technology University
aantunes@captechu.edu

***Abstract*—The U.S. military requires strategic capabilities in the space domain and viable solutions must avoid causing collisions while also avoiding detection. A new method for satellite sabotage is to deliberately impair the attitude (but not the orbit) of a target spacecraft via the direct attachment and thrust output of TrojanCube, a sabotage picosatellite using the CubeSat form factor. The small attitude perturbations created can be varied to mimic anomalies that preoccupy the target satellite ground systems team. This project utilized a demonstration model and simulated momentum build-up potential to showcase feasibility and found that TrojanCube is an effective method to sabotage spacecraft functions and operation.**

Keywords—Aerospace and electronic systems, Aerospace engineering, Attitude control, Low Earth orbit satellites, Military satellites

I. Introduction

As access to, and reliance on, space infrastructure continues to expand, Earth orbital space is becoming a more contested operational environment. Space Policy Directive-3 (SPD-3) of the National Space Traffic Management Policy highlights the fact that a congested space system presents challenges for the safety, stability, and sustainability of U.S. space operations. [1]. Since space is the ultimate high ground, the US military needs viable options to deceive, degrade, deny, disrupt, or destroy adversary access to assets in the space domain.

However, viable solutions must avoid causing fragmentation and scattering of orbital debris, which pose a threat to friendly assets and can impair overall domain access. Traditional kinetic weapons do not meet this constraint as evidenced by the results of the People's Republic of China's (PRC) 2007 direct ascent anti-satellite test as detailed in a 2007 congressional research report. The anti-satellite missile test resulted in a debris cloud. " This debris cloud (estimated at 950 pieces 4 inches or bigger plus thousands of smaller pieces) threatens space assets in LEO…" [3]. Deorbiting a space asset is likewise a suboptimal solution, as a lost satellite is typically replaced with a new upgraded satellite. A solution to the defined problem space is to suriptitiously and deliberately impair the attitude but not the orbit of a target spacecraft via the direct attachment and thrust output of TrojanCube, a sabotage cubesat. This impairs the ground operations and requires the adversary to continually adjust their Concept of Operations (ConOps) while keeping the space assets themselves intact, but less useable. This method of spacecraft attack provides the non-kinetic disablement/impairment of a target which prevents debris scattering and causes both acute and aggregate long term negative effects to its target.

The project described in this paper explored this idea by physically demonstrating the above concept, as well as a potential software solution, in a neutral buoyancy test environment. The physical demonstration produced thruster commands based on attitude sensor feedback to repeatedly disturb a model spacecraft's orientation. This physical demonstration was supplemented with more accurate computational simulation work to gauge likely real world effects and explored torque profiles for commercially available small form thrusters as well as momentum buildup potential. The result of this project was an analysis of achieved perturbations and their potential effects on a target's attitude as well as the resultant potential feasibility of this asset vulnerability.

The stage of the proposed solution that will be explored by this project depends on prerequisite technologies that are engineering challenges in their own right and must be acknowledged and addressed for future development of this system. Most critically, there must exist vulnerability windows or blind spots in which rapid deployment and maneuvering of the cube can deliver it to the intended target proximity without detection by the adversary or a third party. Next, timely selective deployment of the Trojan Cube system would likely require a "nested" approach where the cube, or cubes, is delivered to orbit by a larger space asset when required by mission. This secondary deployment of small form satellites is already beginning to raise post deployment identification and tracking concerns such as the case of Spaceflight's SSO-A mission (Figure 1) [4]. During deployment, the cube would need to be able to sense its target, intercept, match attitude, and attach to a surface. Finally, this project's stage of development exists at a proof on concept level and does not include design of the codependent subsystems of a fully integrated system. Generation of this weapon system would require a complete system design.

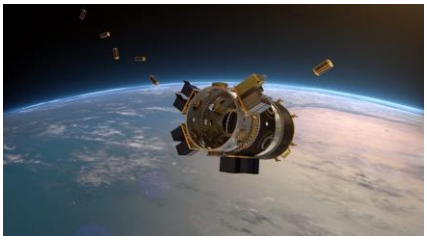


Fig. 1. "Spaceflight's SSO-A mission will deploy 64 satellites, raising concerns from some experts about the ability to accurately track and identify them after deployment." [4]

Copyright © AJSE Publishers

The TrojanCube technology at its essence is designed as a targeted offensive sabotage capability. The intended employment of this technology should result in the disruption of the primary mission of a target spacecraft or the services it provides, the drain of resources and manpower, and the degraded capacity of the space asset in its entirety.

II. BACKGROUND

Commercial and military entities increasingly rely on space infrastructure, resulting in congested and contested orbital lanes. This need for maintaining US space superiority is laid out in the first doctrinal publication of the U.S. Spaceforce, Spacepower. "When warranted, offensive operations are designed to achieve a relative advantage by negating an adversary's ability to access, or exploit the space domain and are therefore essential to achieving space superiority."[2]. Spacepower goes on to define offensive operations as those which "target an adversary's space and counterspace capabilities, reducing the effectiveness and lethality of adversary forces across all domains."[2].This gives rise to the need for non-kinetic offensive capabilities in the space domain. Deliberate impairment to the attitude of a target spacecraft via the direct attachment and thrust output of a Trojan cubesat is a technological solution to this problem at efficient cost and a smaller scale than conventional 'space tug' proposals.

The most important and practical mitigation technique is actually the development of responsive, defensive technology solutions. We have seen this occur in other domains as well. For example, the development of chemical gas weapons drove advancements in soldier personal protective equipment. Similarly, in this particular case, responsive, defensive mitigation strategies would result in engineering solutions that would protect space assets from this particular sabotage vulnerability. In developing a new technology, one must necessarily understand how a system responds to the new technology which, in turn, exposes potential defensive mitigation strategies. Going through this process, which forces one to examine both the offensive capabilities of a new technology as well as the potential strategies to defend against that technology, is the single strongest argument for the advancement of national defense technologies.

Technological solutions must be explored once defined because if they are left unknown, someone else will inevitably explore them. By accepting responsibility to explore "harmful-by-nature" technologies, one also ensures exposure to the potential to design a shield from that harm should someone else develop it. The United States Department of Defense is charged with protecting the assets and interests of the State; that is their ethical obligation. As our reliance and dependence on space infrastructure and assets grows exponentially, so too will the necessity and urgency of defense capabilities. This ethical framework is, and will be, shared by all spacefaring entities past, present, and future. The right to defense is innate, expected, and the way that nations attempt to ensure their survival. Pragmatically, the question, "Is development of this technology ethical?", is shortsighted and misses the bigger picture that technology development fits into. Technology is just the relatively recent extension of a several billion-year-old evolutionary arms race. This kind of system of progress, at that kind of scale, is likely as innate and inevitable as can be. Therefore, not only is the development of this technology application ethical, but it is an eventuality.

III. DEMONSTRATION METHOD AND RESULTS

In order to physically demonstrate the capability of a TrojanCube to disturb the attitude of a spacecraft, and thus sabotage its mission functions and operation [7], we built a scaled physical model spacecraft and attached a motor and propeller to simulate the TrojanCube. The system architecture and test methodology for this model system is described below.

The demonstration model consisted of a 1/10th scale geometric approximation of an Iridium NEXT spacecraft. The model structure was built using a 1⁄2 inch PVC frame, an acrylic mounting sheet, and an acrylic watertight tackle box. The tackle box housed a Raspberry Pi 4, L298N motor driver, MPU-6050 accelerometer/gyroscope, and two battery banks. The tacklebox was affixed to the upper half of the frame to achieve positive buoyancy on the top of the model as a result of the air pocket within the compartment. To achieve negative buoyancy on the bottom of the model, sections of the frame were drilled to allow flooding while sections of pipe filled with gravel and sand were affixed to the bottom of the frame. Figure 2 presents a top down view of the completed model system.

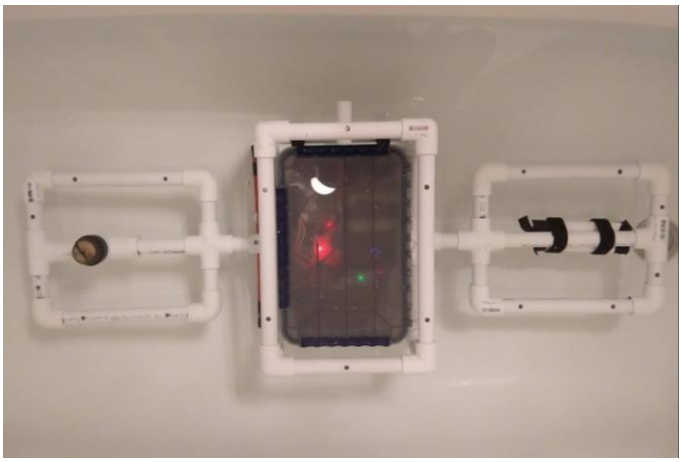


Fig. 2. Early controlled system functionality testing with the completed model spacecraft submerged in a bathtub (Top-down view).

This combined structural arrangement had the effect of a relatively neutrally buoyant overall frame with clear positive and negative buoyant poles. This effect was intentional and allowed for the test rig to right itself passively and was meant to function as a passive attitude and control system. To simulate a Trojan Cube attached to a solar panel of the Iridium NEXT spacecraft, a small waterproof DC motor with a 3D printed propeller protruded from the PVC piping of the model spacecraft to roughly represent a 1/10th scale 6U cubesat. DC power was delivered to the motor via a cable routed through the PVC frame. This allowed for the ability to provide repeatable force at the Trojan attachment point. The model was connected to a physical stand via a floating wooden dowel inserted into the PVC frame. This allowed for freedom of rotation in a single axis as a result of applied thruster force. Areas intended to remain dry were sealed

Copyright © AJSE Publishers

with plumbers wax and marine sealant to avoid unwanted water exposure.

A test routine program was written in Python to allow autonomous operation of the thruster while the entire test rig was submerged in a pool. After a time delay, the program began sampling the MPU-6050 at a rate of 10x per second and compared each successive sample to the last. If the variance between samples was within a defined tolerance, a static counter incremented. At 20 seconds of static movement, which simulated the target operating nominally, the thruster would fire. The counter would then reset and wait for the target to again resume nominal operating orientation for 20 seconds before the next thruster fire. Sampled attitude data during the test runtime was stored in CSV files and saved onboard the Pi. After completion of the test routine, the CSV files were run through a post-process filter to determine the target's roll in the X-axis over test duration. In addition to the stored sensor data, a high-definition video camera was used to record the test iterations for future demonstration and comparison to the stored data. Video snapshots from the recorded test runs are shown in Figure 3.

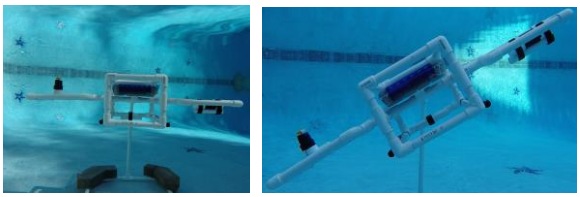
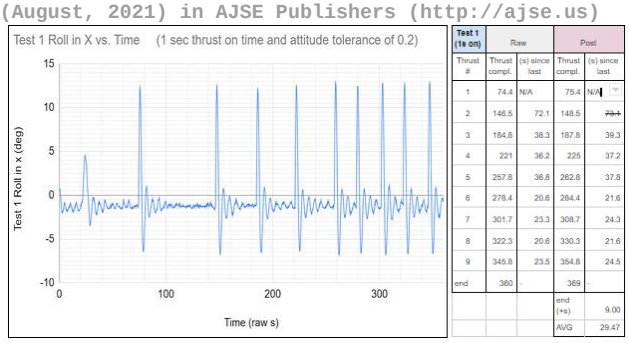


Fig. 3. (Top) Model and test rig submerged in a pool at approximately nominal orientation. (Bottom) Model and test rig submerged in a pool with significant attitude disturbance after thruster fire, which manifested as a +X roll in the model's frame.

The physical demonstration yielded 4 successful test runs, each greater than 6 minutes in length. Each run produced raw sensor data from the MPU-6050 and a text file with test event readouts. The sensor data was then able to be used to compile a data visualization for roll in the model's X-axis that coincides with the underwater video of the test runs. The TrojanCube concept was clearly demonstrated and the thruster disturbance properly executed every time the program recognized that the target had returned to a static attitude. Average time between thrusts was greater than the static limit imposed for all test runs, confirming that a notable attitude perturbation was induced.

After successful test runs were recorded, the sensor data was run through a filter to provide the relative roll in the model's X-axis. This data was then compiled in a spreadsheet and graphed with raw test time to provide a visual representation of the roll over time. The visual representation of the first of the four recorded successful test runs can be seen in Figure 4. The largest consistent spikes in the disturbance of the model's X-axis represent thruster actuation and can be differentiated from the early smaller spike that resulted from the test conductor physically handling the model.



Test 1 (1s on)	Raw		Post	
Thrust #	Thrust compl.	(s) since last	Thrust compl.	(s) since last
1	74.4	N/A	75.4	N/A
2	146.5	72.1	148.5	~~73.1~~
3	164.8	38.3	187.8	39.3
4	221	36.2	225	37.2
5	257.8	36.8	262.8	37.8
6	278.4	20.6	284.4	21.6
7	301.7	23.3	308.7	24.3
8	322.3	20.6	330.3	21.6
9	345.8	23.5	354.8	24.5
end	360		369	
			end (+s)	9.00
			AVG	29.47

Fig. 4. Roll in X (deg) versus test time (raw s) for the first test run. Test run 1 had a one second thrust setting as well as an attitude tolerance of 0.2. Raw thrust complete times and thruster fire spacing is displayed in the center "Raw" columns with corrected values under the "Post" columns. The bottom two cells represent the collective test time creep and the average time between thruster firings.

Beyond the initial successful test run, risk of water intrusion and the frigid water temperatures for the test conductor resulted in test iterations being rapid and chaotic. This resulted in the tolerance variable being incremented between tests in the wrong direction and was discovered later in post-test analysis. It was wrongly assumed that perceived delays between thruster firings was due to too restrictive of a tolerance when, in fact, it was actually due to a timing anomaly. This anomaly was a coding oversight, which presented as test time gain relative to thruster on-time. It was discovered that the test timer was not incrementing, nor were sensor readings being recorded, during thruster on-time. There were other variables in the test environment not originally accounted for, which presented in the data to include wind effects on surface water, sun versus shade temperature differences, and currents from the test conductor's physical entry and exit of the pool; however, these variables did not functionally affect the intended demonstration and proof of the concept.

Four test runs were completed and the aggregate data was plotted. The combined graph (Figure 5) shows the roll in X for the last 160 raw test seconds for each test run. With each successive test, as the thruster on-time was increased, the amplitude of the roll in X likewise increased.

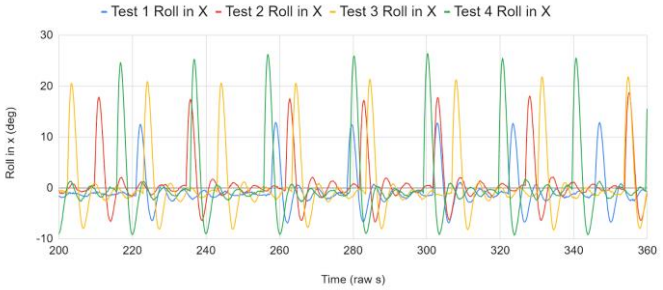


Fig. 5. Aggregate visual representation of the roll in X over the test period from 200 seconds to 360 seconds for four separate test runs. Test 1 used a 1 second thrust on-time and attitude tolerance of 0.2. Test 2 used a 1.5 second thrust on-time and attitude tolerance of 0.5. Test 3 used a 2 second thrust on-time and attitude tolerance of 1. Test 4 used a 3 second thrust on-time and attitude tolerance of 1.

This paper asserts that attitude disorientation of a spacecraft is an effective method of spacecraft sabotage. The degree of disorientation would be dependent on some mission specific tolerance to be deemed meaningful enough to adequately disturb the spacecraft. Here, the first test case used a tolerance of 0.2 difference between subsequent raw sensor readings. Since activation of the

Copyright © AJSE Publishers

thruster is dependent on the sensor determining a static state, meaningful perturbation was determined because the time between thruster activations was greater than the static time limit.

As shown in Figure 6, the response of the target satellite to the Trojan perturbation was linear with the range tested; doubling the duration of the applied force roughly doubled the induced deviation in the satellite movement. These measurements were expected and parallel results from simulations. This validates the concept that a small TrojanCube using low impulse microthrusters can provide a meddlesome perturbation to the larger target satellite. The linear response also makes for easier design of perturbation profiles so that the TrojanCube can produce its choice of steady, erratic, or itinerant perturbation in order to confuse ground operations as to the source of the deviations.

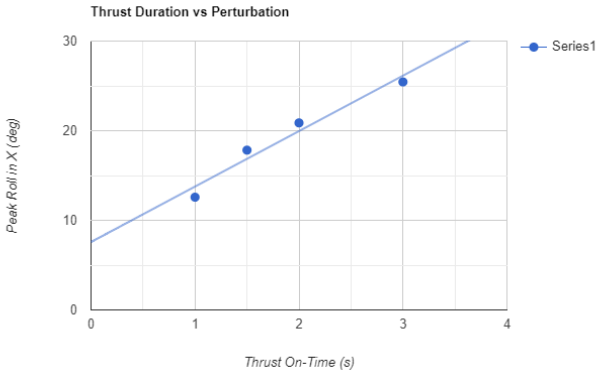


Fig. 6. Thrust duration versus perturbation. Vertical axis defined by peak average disturbance per test run. Horizontal axis defined by test run 1-4 thruster on-times.

IV. Simulation Method and Results

In parallel with the physical demonstration, we also created a simulation wherein the target Iridium NEXT and Trojan models are represented by basic geometric shapes with uniform density. As with the physical model, the moment of inertia changed when the TorjanCube was attached, which also changed the disturbance torque to the target from the Trojan thruster. We modeled both the disturbance momentum build up and the approximate fuel required to handle momentum dumping. We simulated the attitude perturbation potential of a target satellite as a result of a TrojanCube's thrust output, which required calculations of: the moment of inertia changes due to attachment of the TrojanCube, achievable disturbance torque to the target from the Trojan thruster, momentum build up over an orbit, and potential fuel required to handle momentum dumping. The program Matlab and its corresponding language were used to write scripts, which performed these calculations. Each script used a set of user-defined initial conditions, took into account certain general assumptions, and stepped through the required calculations. Figure 7 depicts the system inputs, assumptions, and script functions for the disturbance torque modelling.

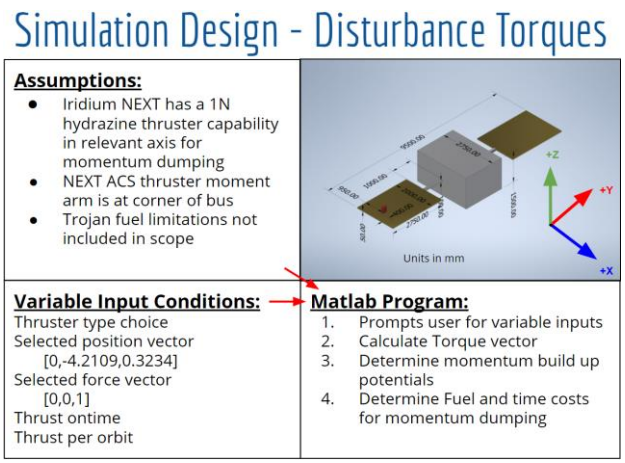


Fig. 7. Overview of disturbance torque script architecture and functions.

In calculating the moment of inertia changes, it was assumed that all components had uniform density, were cuboid in shape, and existed in the same reference frame. The script first determined individual component mass using an estimated total mass and percentage of total volume for each component. Next, the individual moments of inertia were calculated using the standard derivation for a cuboid moment. The center of mass of the model was defined, from an arbitrary reference point, as were the offsets from center of mass for each individual component. Then, the parallel axis theorem was used to move the moments to the center of mass. Next, the combined moment at the center of mass was found. Finally, the TrojanCube model was included in the system and a new center of mass was determined. Each of the prior steps was then repeated to find the new combined moment of inertia.

For the torque and momentum calculations, it was assumed that the Iridium NEXT had a 1N hydrazine thruster capability in the relevant axis for momentum dumping [8], the NEXT's thruster moment arm was at the corner of the spacecraft bus, and TrojanCube's fuel limitations would not be considered in the project scope. To achieve these calculations, the Matlab script prompted the user for a set of variable input conditions. First, it prompted the user to select a Trojan thruster from a set of commercially available small satellite thrusters. Second, it asked the user to provide the attachment point or position vector of the Trojan thruster, as well as the force vector. Last, it requested input of the thrust on-time and frequency of thruster firings per orbit. The Matlab script took these inputs and determined the resultant torque applied. From this torque calculation, momentum build up was also calculated for portions of an orbit over multiple orbits, as well as fuel requirements to dump the gained momentum under the simulated configuration. The table in Table 1 displays the relationship between thruster type selection, the "as-configured" torque applied by the thruster, and the selected thruster on-time per orbit. Since thruster force varied so widely by type, the thrust on-times were selected in order to constrain the momentum buildup outputs so that they were more visually comparable. It is, however, worth noting that the actual thrust on-time in a real-world mission setting would be dictated by the mission target and fuel constraints [9].

Copyright © AJSE Publishers

Thruster	Applied Torque (Nm)	Thrust Time Per Orbit
EPSS-C1 1N BOL to 0.22N EOL (0.61N AVG)	-2.5686 0 0 X Y Z	1 sec
Regulus EPS 0.55mN nominal Plasma thruster	-0.0023 0 0 X Y Z	20 min
Halo 4-34mN (19mN AVG) hall-effect thruster	-0.0800 0 0 X Y Z	30 sec
PM200 0.5N nominal bi-propellent thruster	-2.1054 0 0 X Y Z	1 sec
IFM Nano Thruster 350uN nominal FEEP thruster	-0.0015 0 0 X Y Z	20 min

TABLE I. Table displays thruster type selection, the "as-configured" torque applied by the thruster, and the selected thruster on-time per orbit.

The script output graphs showing momentum buildup and responsive fuel mass required to handle momentum unloading over multiple orbits for each thruster selection; these graphs are included below. The first graph shows that in every thruster configuration, significant momentum build up can be reached with limited thruster on-time over a small number of orbits relative to a standard mission duration. As applied to a real-world TrojanCube mission, significant disturbances can be induced over a relatively short period of overall target mission time because the TrojanCube would likely be employed in low Earth orbit, which means that the orbital period would consequently be constrained to between 84 and 127 minutes.

The second graph below (Figure 8) shows that the fuel mass expenditure required of the target to handle the applied disturbance is minimal. For context, Iridium NEXT carries 141 Kg of Hydrazine monopropellant [9]; yet, the graph shows that the "as-configured" fuel mass expenditure never rises above 0.025 Kg, even after 30 orbits. This means that the fuel expenditure incurred on the target is acutely insignificant. Given the likely scale difference between fuel reserves for a cubesat versus a larger spacecraft, fuel expenditure can likely be deemed an ineffective vulnerability vector. However, given the precise pointing requirements imposed on a typical spacecraft mission, even slight disturbance to attitude will have immediate negative effects imposed on the target.

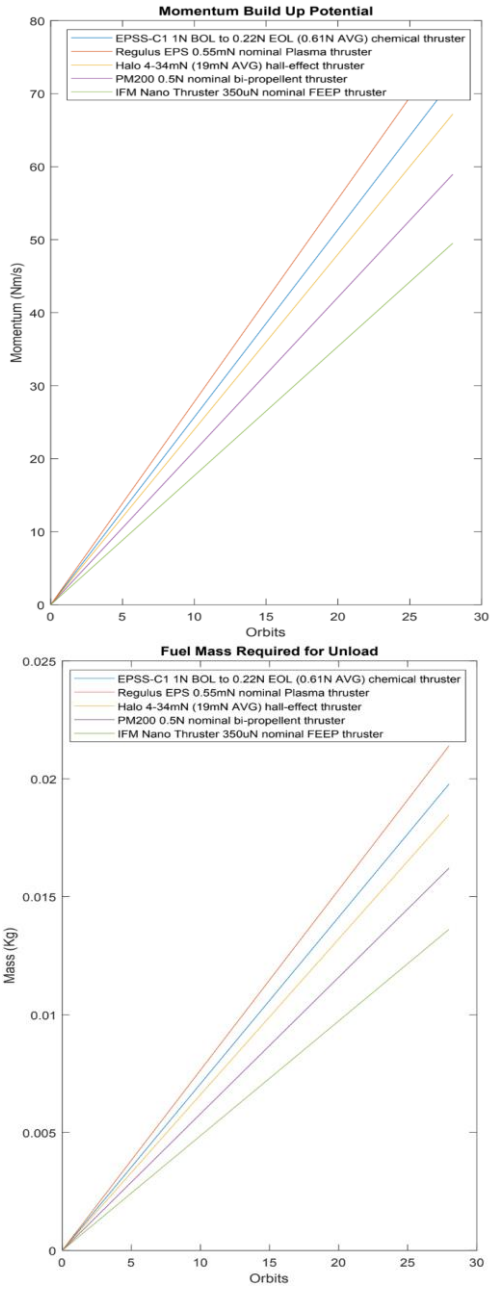


Fig. 8. Script output graphs of momentum (Top) and responsive fuel mass requirement (Bottom) over multiple orbits for each thruster selection.

Additionally, as one would expect, the simulation showed that higher-force thrusters provided higher torques and took less time to accumulate substantial momentum buildup. The lower-force, Ion-type thrusters had to run for longer; although, given more time, they could achieve the same momentum buildup. In all thruster cases, however, appreciable torques can be induced quite rapidly, resulting in disruption of the primary function of the target spacecraft.

Based on the resulting simulation data, it is apparent that the TrojanCube method is a viable option for spacecraft sabotage.

V. Ethical Concerns

As with any technology, this new idea for satellite sabotage raises certain ethical concerns regarding its

Copyright © AJSE Publishers

intended uses, its unintended uses, and how this technology fits in with humanity. We establish that this technology's intended effect of sabotaging spacecraft is harmful by nature, but argues that its ethicality should be judged on how it achieves its desired goals — by minimizing the effect on orbital crowding and debris risk — and not merely on its harmful nature. Here, the technology is designed to incapacitate a spacecraft, yet it does so without adding to the orbital debris problem, thus achieving its goal in an ethical way. Since this method employs the use of cubesats, which tend to be inexpensive, this could easily be scalable and repeatable.

Because this method of attack affects the attitude and orbit of a victim spacecraft, it is conceivable that it could also be used to deliberately steer that spacecraft into situations where a collision has a greater likelihood. Although unintentional, the collision of Iridium 33 and Cosmos 2251 produced greater than 1800 new orbital debris in low Earth orbit (LEO)[5]. By misusing the technology, a bad actor could intentionally congest orbital lanes with more debris, which could threaten and potentially cut-off access to critical space infrastructure through a runaway process known as Kessler Syndrome [6]. This is precisely the opposite effect of the problem for which this technology attempts to provide a solution. Due to the potential for this misuse, the question again must be asked: is this technology ethical to pursue? The potential for misuse exists with every tool man has ever created and, in fact, often serves as the driving force behind new technological problems that need answers. Developing creative solutions to address these problems can, in turn, push our technological horizon forward.

VI. Conclusions

Weapons development often comes with a large price tag, however, cubesats have a much lower barrier to entry than larger missions [10]. While numbers can vary wildly for putting a satellite into orbit, there is, on average, several orders of magnitude in difference between the price of a satellite and a cubesat. The average cost to put a cubesat into orbit is on the order of 10s to 100s of thousands of dollars ($US), while the average cost to put up a larger satellite is 10s to 100s of millions of dollars ($US). As an additional comparison, space-based-interceptor missile systems were estimated to have costs in the billions [11]. However, cubesats by nature have both low development and production costs. Their relative size and low mass constrains both the physical resource requirement per cube and the complexity of systems that can be included due to available space. The use of standardized design and commercially available off-the-shelf parts can allow for rapid mass production. This means that procurement of this weapon system would be both highly scalable and economically viable.

The physical and simulation demonstrations of our project also show that the TrojanCube is a realistic option for spacecraft sabotage. The project set out to explore the concept of deliberately impairing the attitude and orbit of a target spacecraft via the direct attachment and thrust output of a Trojan cubesat. Our physical model produced thruster commands based on attitude sensor feedback, repeatedly disturbing orientation, which demonstrated the overall concept and a potential software solution in a neutral buoyancy test environment. It also showed that this method can be executed autonomously. Additionally, we successfully modeled simulations of attitude perturbations of a target Iridium NEXT model as a result of a TrojanCube's thrust output. We found that appreciable torques and subsequent momentum build up can be applied to a target by a range of commercially available small-form thrusters in a short period of time. The overall project demonstrated that a TrojanCube can quite easily disrupt the primary mission function of a target spacecraft due to the precision pointing required for many instruments, high gain antennas, and solar panel facings.

Although scalable, economically viable, and physically realistic, the operation of this potential weapon system also raises a set of considerations. First, deployment of the system is dependent on delivery to orbit and subsequent maneuvering by the cube to the target. There are several options to examine when considering how to employ the TrojanCube method of attack. For example, one option is to deploy multiple TrojanCubes as a primary payload, delivering them in batches to multiple orbits where they lay dormant, awaiting activation from the ground; while another option is for a singular TrojanCube to hitch a ride as a secretive secondary to an alternative primary mission that, when activated, separates and redirects towards its intended target. Another consideration that must be addressed is that international space and arms laws may present a barrier to development and deployment of this system and would require thought before real development work could proceed. Finally, one must consider the issue of how to conduct functional in situ system testing economically. Because there is no shortage of defunct materials and spacecraft in orbit, assessment of intended effects of the weapon system could be conducted against any of these potential targets or a friendly asset that has reached, or is approaching, end of life.

While deployment of the TrojanCube system and in situ assessment raise unique considerations, the provided solutions address those concerns and furnish unique opportunities for creative development. Additionally, given the low individual cost and high scalability potential, a TrojanCube should be considered an extremely likely and economical weapon system. The demonstration model and simulated momentum build-up potential also showcased the potential effectiveness and feasibility of the TrojanCube in the space domain and found that TrojanCube is an effective method to sabotage spacecraft functions and operation.

References

[1] "MEMORANDUM FOR THE VICE PRESIDENT THE SECRETARY OF STATE THE SECRETARY OF DEFENSE THE SECRETARY OF COMMERCE THE SECRETARY OF TRANSPORTATION THE SECRETARY OF HOMELAND SECURITY Space Policy Directive-3, National Space Traffic Management Policy," 2018. Accessed: Mar.01, 2021. [Online].

Available:https://aerospace.org/sites/default/files/policy_archives/Space%20Policy%20Directive%203%20-%20STM%2018Jun18.pdf.

Copyright © AJSE Publishers

[2] “Space Capstone Publication, Spacepower (SCP),” spaceforce.mil, Jun. 2020.

[3] S. Kan, “Chinese Test Anti-Satellite Weapon,” 2007. Accessed: Mar. 01, 2021. [Online]. Available: https://fas.org/sgp/crs/row/RS22652.pdf.

[4] “Dedicated rideshare Falcon 9 launch raises satellite tracking concerns,” SpaceNews, Nov. 30, 2018. https://spacenews.com/dedicated-rideshare-falcon-9-launch-raises-satellite-tracking-concerns/ (accessed Mar. 01, 2021).

[5] “NASA Technical Reports Server (NTRS),” ntrs.nasa.gov. https://ntrs.nasa.gov/citations/20100002023.

[6] “Micrometeoroids and Orbital Debris (MMOD),” NASA, 2011. https://www.nasa.gov/centers/wstf/site_tour/remote_hypervelocity_test_laboratory/micrometeoroid_and_orbital_debris.html.

[7] D. V. Lebedev and A. I. Tkachenko, “High-Precision Attitude Control of Remote Sensing Satellite,” IFAC Proceedings Volumes, vol. 37, no. 6, pp. 735–740, Jun. 2004, doi: 10.1016/s1474-6670(17)32264-4.

[8] “Iridium-NEXT – Spacecraft & Satellites.” https://spaceflight101.com/spacecraft/iridium-next/ (accessed Mar. 01, 2021).

[9] “Thruster Principles,.” Accessed: Mar. 01, 2021. [Online]. Available: https://descanso.jpl.nasa.gov/SciTechBook/series1/Goebel_02_Chap2_thruster.pdf.

[10] M. N. Sweeting, “Modern Small Satellites-Changing the Economics of Space,” Proceedings of the IEEE, vol. 106, no. 3, pp. 343–361, Mar. 2018, doi: 10.1109/jproc.2018.2806218.

[11] J. Judson, “US agency forecasts cost for missile defense plans over next decade,” Defense News, Jan. 19, 2021. https://www.defensenews.com/pentagon/2021/01/19/cbo-missile-defense-plans-on-path-to-cost-176-billion-over-next-decade/#:~:text=While%20space%2Dbased%20interceptors%20do (accessed Mar. 03, 2021).

Copyright © AJSE Publishers