Published in the Military Communications Conference (MILCOM) 2015, IEEE. http://ieeexplore.ieee.org/document/7357578/

On the Smallest Pseudo Target Set Identification Problem for Targeted Attack on Interdependent Power-Communication Networks

Arun Das, Chenyang Zhou, Joydeep Banerjee, Arunabha Sen
School of Computing, Informatics and Decision System Engineering
Arizona State University
Email: {arun.das, czhou24, joydeep.banerjee, asen}@asu.edu

Lloyd Greenwald
Internet and Cybersecurity Research Department
LGS Innovations
Email: lgreenwald@LGSInnovations.com

***Abstract*—Recognizing the need for a deeper understanding of the interdependence between critical infrastructures, such as the power grid and the communication network, a number of models have been proposed and analyzed in the last few years. However, most of these proposed models are over simplified and fail to capture complex interdependencies that may exist between these critical infrastructures. The recently proposed *Implicative Interdependency Model* is able to capture these complex interdependencies involving conjunctive and disjunctive relationships to overcome most of these limitations. Due to the existing interdependencies between the power and communication networks, a failure involving a small set of power and/or communication network entities can trigger a *cascading event*, resulting in the failure of a much larger set of entities through the cascading failure process. This implies that an adversary with an intent of destroying a specific set of entities E' (*real targets*), no longer needs to make an effort to destroy E' directly, but instead identify a set of smaller entities E'' (*pseudo targets*), whose destruction eventually leads to the destruction of the real target set E' due to the cascading failure process. A clever adversary will thus try to identify the smallest set of pseudo target entities E'', whose destruction eventually destroys E'. We refer to this problem as the *Smallest Pseudo Target Set Identification Problem* (SPTSIP). We divide the problem into four classes, and show that it is solvable in polynomial time for one class, and is NP-complete for others. We provide an *approximation algorithm* for the second class, and for the most general class, we provide an optimal solution using ILP, and a heuristic solution. We evaluate the efficacy of our heuristic using power and communication network data of Maricopa County, Arizona. The experiments show that our heuristic almost always produces near optimal results.**

I. Introduction

The last few years have seen a heightened awareness in the research community that the critical infrastructures of the nation do not operate in isolation. Instead, these infrastructures are closely coupled together and form a complex ecosystem of interdependent networks, where the well being of one infrastructure depends heavily on the well being of another. A case in point is the interdependent relationships between the electric power grid and the communication network. Power grid entities, such as the SCADA systems that control power stations and sub-stations, are reliant on the communication network to send and receive control signals. On the other spectrum, communication network entities, such as the routers and base stations are reliant on electric power. Understanding the impact of cascading failures in the power grid, a not so uncommon phenomena, becomes even more complex when the coupling between the power grid and communication network entities are considered. This coupling, or interdependence, allows not only entities in the power network, such as generators and transmission lines, to trigger power failure, but also communication network entities, such as routers and optical fiber lines, can potentially trigger failures in the power grid. Thus, it is imperative that the interdependencies in this complex network ecosystem be well understood, so that preventive measures can be undertaken to avoid catastrophic failures in *Interdependent Power-Communication Networks* (IPCN).

To address this need for a deeper understanding of interdependencies between multi-layered networks, in the last few years, the research community has made significant efforts [1-12], and accordingly proposed and analyzed a number of models. However, most of these proposed models are over simplified and fail to capture complex interdependencies that may involve a combination of conjunctive and disjunctive relationships. Suppose the power network entities such as *power generators, transmission lines* and *substations* are denoted by the set $A = \{a_1, a_2, \ldots, a_n\}$ and the entities of the communication network, such as *routers, fiber optic lines* and *base stations* are denoted by the set $B = \{b_1, b_2, \ldots, b_m\}$. Due to the topological design of the power-communication networks, it may so happen that an entity a_i may be *alive* (or operational) if (i) b_j *and* b_k are alive, *or* (ii) b_l *and* b_m *and* b_n are alive, *or* (iii) b_p is alive. Graph based interdependency modeling, such as in [1-7], [10] cannot capture such interdependencies involving conjunctive and disjunctive terms. The recently proposed *Implicative Interdependency Model* (IIM) [13] is a Boolean logic based model that overcomes these limitations and accommodates such complex interdependencies.

This interdependent relationship between the entities of the IPCN implies that a failure involving a small set of entities can trigger a *cascading event* that results in the failure of a much larger set of entities. This creates a potential scenario where an adversary with an intent to jeopardize a specific set of entities E', or *real targets*, now no longer needs to destroy E' directly.

This research was supported in part by the NSF grant 1441214. The data for the Maricopa county communication network used in this research was provided by GeoTel communications (www.geo-tel.com).

Instead, the adversary can take advantage of the cascading failure process by identifying a smaller set of entities E'', or *pseudo targets*, whose failure eventually leads to the failure of E' due to the cascade. Thus, the objective of the adversary is to identify the smallest set of pseudo targets E'' whose failure eventually causes E' to fail. In this paper we refer to this problem as the *Smallest Pseudo Target Set Identification Problem* (SPTSIP), and in the IIM setting, categorize the problem in four classes. We show that one class of the problem is solvable in polynomial time, whereas for others it is NP-complete. For the second class of the problem we provide an *approximation algorithm*, and for the most general form of the problem we provide an optimal solution using ILP, and a heuristic solution. Finally, we evaluate the efficacy of our heuristic using power and communication network data of Maricopa County, Arizona. The experiments show that our heuristic almost always produces near optimal results.

II. Implicative Interdependency Model (IIM)

The *Implicative Interdependency Model* (IIM) proposed in [13], is an entity based model that allows representation of complex dependency relations between entities of multi-layer network systems. The dependent relationships between the network entities are represented using Boolean Logic and are termed as *Implicative Interdependency Relations* (IDRs). Table I outlines a set of IDRs representing a sample IPCN where the power network and communication network entities are represented by the sets $A = \{a_1, a_2, a_3, a_4\}$ and $B = \{b_1, b_2, b_3\}$ respectively. The IDRs represent a set of Boolean conditions that need to be satisfied for an entity to be operational. In Table I, entity b_1 is operational if either a_2 is operational, or both a_1 and a_3 is operational. It may be noted that although in the IDRs of this example, A (B) type entities appear on either the left hand side or the right hand side of an IDR, the IIM does not require that A (B) type entities appear only on one side of an IDR. In other words, an IDR can also be of the form $a_i \leftarrow a_q b_j b_k b_l + a_r b_m b_n + b_p + a_s$. The conjunction of entities, such as $a_r b_m b_n$, is also referred to as a *minterm*.

Power Network	**Comm. Network**
$a_1 \leftarrow b_1 b_2$	$b_1 \leftarrow a_1 a_3 + a_2$
$a_2 \leftarrow b_1 + b_2$	$b_2 \leftarrow a_1 a_2 a_3$
$a_3 \leftarrow b_1 + b_2 + b_3$	$b_3 \leftarrow a_1 + a_2 + a_3$
$a_4 \leftarrow b_1 + b_3$	$--$

TABLE I: A sample Interdependent Power-Communication Network

Entities	Time Steps (t)						
	0	1	2	3	4	5	6
a_1	0	0	1	1	1	1	1
a_2	1	1	1	1	1	1	1
a_3	0	0	0	0	1	1	1
a_4	0	0	0	0	1	1	1
b_1	0	0	0	1	1	1	1
b_2	0	1	1	1	1	1	1
b_3	1	1	1	1	1	1	1

TABLE II: Time Stepped Failure Propagation in a Multilayer Network. A value of 1 denotes entity failure.

The interdependencies expressed through IDRs govern the failure cascade process in IIM, where the failure of a set of entities can trigger further failures due to the interdependencies shared between the entities. We illustrate this cascading failure process with the help of an example. For the IPCN system of Table I, Table II shows a time-stepped cascading failure of entities triggered by the failure of $\{a_2, b_3\}$ at time step 0.

The dependency (IDR) formulation in the IIM setting can either be done by careful analysis of the underlying system as was done in [11], or by consultation with subject matter experts of these complex systems. In this paper we utilize IIM to model the underlying IPCN system and proceed to formulate and analyze the SPTSIP in this setting.

III. Problem Formulation and Computational Complexity Analysis

In this section we formally state the *Smallest Pseudo Target Set Identification Problem* (SPTSIP) in the IIM setting, and analyze its complexity for different types of dependency relations. We formulate the SPTSIP as follows:

The Smallest Pseudo Target Set Identification Problem
INSTANCE: Given:
(i) the set A and B representing the entities of the power and communication networks respectively with $n = |A|$, $m = |B|$
(ii) a set of dependency relations or IDRs, between A and B type entities
(iii) the set of *real targets* $E' \subseteq A \cup B$
(iv) positive integer K

QUESTION: Is there a subset $E'' \subseteq A \cup B$ of *pseudo targets*, with $|E''| \leq K$, whose failure at time step 0, triggers a cascade of failures resulting in failure of the real target set E' by time step $p = n + m - 1$?

We outline some assumptions for the SPTSIP: First, we assume that an entity $e_i \in A \cup B$ can fail by itself and not due to its dependencies, only at time step 0. Any failures after time step 0 occur due to the cascade effect of entities that failed at time step 0. Second, we assume that dependent entities immediately fail in the next time step, i.e. if $e_i \leftarrow e_j e_k$, and e_k fails at time step $p-1$, then e_i fails at p. Third, time step $p = n + m - 1$ is a sure end of any failure cascade that begins at time step 0 as there are at most $n+m$ entities and we assume that entities cannot become operational once they fail. Finally, the pseudo target set E'' does not have to be unique.

It may be noted that the SPTSIP and the Root Cause of Failure (RCF) problem of [14] are considerably different. In the RCF problem, given a failure set $F, F \subseteq A \cup B$ the objective is to find the minimum number of entities $F', F' \subseteq F$ such that when F' entities fail at time step 0, F fails at time step p. It may be noted that for solving the RCF problem it is sufficient to look at the entities in F and their corresponding IDRs to find a solution F'. The set of entities in $(A \cup B) \setminus F$ can completely be ignored for computing F'. However, in the case of the SPTSIP, for the real target set E' that must fail at time step p there is no requirement that $E'' \subseteq E'$ and E'' can be any subset of $A \cup B$. This modification to the problem considerably changes the techniques required for tackling the SPTSIP.

To analyze the complexity of the SPTSIP, we categorize the type of IDRs encountered in IPCNs in terms of the number of minterms they contain, and the size of each minterm. We analyze the complexity of each of these categories as follows:

A. Case I: Problem Instance with One Minterm of Size One

For Case I the IDR's are represented as: $x_i \leftarrow y_j$, where x_i and y_j are elements of the set A (B) and B (A) respectively. In the example $a_i \leftarrow b_j$, $x_i = a_i$, $y_1 = b_j$. As noted in [13], a conjunctive implication of the form $a_i \leftarrow b_j b_k$ can be written as two separate IDRs $a_i \leftarrow b_j$ and $a_i \leftarrow b_k$. However, this case is considered in Case II and not in Case I. This exclusion implies that the entities that appear on the left hand side of an IDR in Case I are unique. For Case I, Algorithm 1 presents a polynomial time algorithm for the solution of the SPTSIP.

Algorithm 1: Case I Optimal Algorithm for SPTSIP

Data:
1. Set of network entities $A \cup B$, with $n = |A|$ and $m = |B|$
2. A set S of IDRs of the form $y \leftarrow x$, where $x, y \in A \cup B$
3. A set of *real targets* E'

Result: The smallest set of *pseudo targets* E'' such that if E'' fails at time step 0, the real target set E' fails by time step $p = n + m - 1$

```
1  begin
2      Construct a directed graph G = (V, E), where V = A ∪ B.
         For each IDR y ← x in S, where x, y ∈ A ∪ B, introduce
         a directed edge (x, y) ∈ E;
3      For each node x_i ∈ V, construct a transitive closure set
         C_{x_i} as follows: If there is a path from x_i to some node
         y_i ∈ V in G, then include y_i in C_{x_i}. As
         |A| + |B| = n + m, we have n + m transitive closure sets
         C_{x_i}, 1 ≤ i ≤ (n + m). Each x_i is termed as the seed
         entity for the transitive closure set C_{x_i};
4      Remove all the transitive closure sets which are proper
         subsets of some other transitive closure set;
5      E'' ← ∅;
6      while E' ≠ ∅ do
7          For entity e_j ∈ E', find set C_{x_i} such that e_j ∈ C_{x_i};
8          Include seed entity x_i in E'';
9          E' ← E' \ C_{x_i};
10     return E''
```

Algorithm 1 Time Complexity: Since $|A| + |B| = n + m$. Step 2 takes $O(n + m + |S|)$ time. Step 3 can be executed in $O((n+m)^3)$ time. Step 4 takes at most $O((n+m)^3)$ time. The while loop in Step 6 takes at most $O(n(n + m))$. Therefore the overall complexity of Algorithm 1 is $O((n + m)^3)$.

Theorem 1. *For each pair of transitive closure sets C_{x_i} and C_{x_j} produced in Step 3 of Algorithm 1, either $C_{x_i} \cap C_{x_j} = \emptyset$ or $C_{x_i} \cap C_{x_j} = C_{x_i}$ or $C_{x_i} \cap C_{x_j} = C_{x_j}$, where $x_i \neq x_j$.*

Proof. We give a proof by contradiction, assume there exists a pair of transitive closure sets C_{x_i} and C_{x_j} such that $C_{x_i} \cap C_{x_j} \neq \emptyset$, $C_{x_i} \cap C_{x_j} \neq C_{x_i}$ and $C_{x_i} \cap C_{x_j} \neq C_{x_j}$. Let $x_k \in C_{x_i} \cap C_{x_j}$, this implies that there exists a path $P1$ from x_i to x_k, as well as a path $P2$ from x_j to x_k. Thus there exists some x_l such that $x_l \in P1$ and $x_l \in P2$. W.l.o.g. assume that x_l is the first node in $P1$, also, as $x_l \in P2$, x_l has an in-degree greater than 1. This implies that there is more than one IDR in set S such that x_l appears on the left hand side of these IDRs. This is a contradiction as it violates the definition of Case I type IDRs and hence the theorem is proved. □

Theorem 2. *Algorithm 1 gives an optimal solution for the SPTSIP in a multi-layer network for Case I type IDRs.*

Proof. Theorem 1 proves that every pair of the transitive closure sets created in Step 3 of Algorithm 1 are either disjoint or is a proper subset of the other, in Step 4 of the algorithm all transitive closure sets that are proper subsets of some other transitive closure set are removed. This implies that the remaining sets are all necessarily disjoint, and for every $e_i \in E'$, e_i belongs to exactly one transitive closure set. This necessitates that the seed entity x_k of the transitive closure set C_{x_k} that e_i belongs to, must be included in the solution. This is done in the while loop of Step 6. To prove the optimality claim we need to show that the number of seed entities chosen by the algorithm is minimum. If we assume that the number of seeds chosen is not minimum, then some C_{x_i} chosen by the algorithm must necessarily be a proper subset of another closure. This contradicts Theorem 1, and hence Algorithm 1 always returns the optimal solution. □

B. Case II: Problem Instance with One Minterm of Arbitrary Size

For Case II the IDR's are represented as:
$x_i \leftarrow \prod_{k_1=1}^{l} y_{k_1} \prod_{k_2=1}^{q} x_{k_2}$ (with $x_i \neq x_{k_2} \forall x_{k_2}, 1 \leq k_2 \leq q$), where x_i, x_{k_2} are elements of set A (B) and y_{k_2} is an element of set B (A). The size of the minterm is given as $l + q$. In the example $a_r \leftarrow b_u b_v a_s$. $l + q = 3$, $x_i = a_r, y_1 = b_u, y_2 = b_v$ and $x_1 = a_s$.

Theorem 3. *The SPTSIP for Case II is NP Complete*

Proof. We prove that the SPTSIP for Case II is NP-complete by giving a transformation for the Set Cover (SC) problem [15]. An instance of the set cover problem is specified by a universal set $S = \{s_1, \ldots, s_{n+m}\}$ and a set of subsets S', $S' = \{S_1, \ldots, S_q\}$, where $S_i \subseteq S, \forall i, 1 \leq i \leq q$. In the set cover problem one wants to know whether there exists a subset of $S'' \subseteq S'$ such that $\bigcup_{S_i \subseteq S''} S_i = S$ and $|S''| \leq Q$, for some specified integer Q. From an instance of the SC problem we create an instance of the SPTSIP in the following way: For every $s_i \in S$, we create an IDR of the form $s_i \leftarrow \prod_{s_i \in S_j} S_j$ $\forall S_j \in S'$. We set the real target set $E' = S$ and $K = Q$. It can now easily be verified that the instance of the SC problem has a set cover of size Q, iff in the created instance of SPTSIP the failure of K entities at time step 0 triggers a cascade of failures resulting in the failure of all entities in the set E' by time step $p = n + m - 1$. □

We now define the following:

Definition: *Kill Set of a set of Entities $\mathcal{P}$*: The *Kill Set* of a set of entities $\mathcal{P}$, denoted by $KillSet(\mathcal{P})$, is the set of all entities in the multilayer network (including $\mathcal{P}$) that fail by

$p = n + m - 1$ time steps as a consequence of: (i) the failure of $\mathcal{P}$ entities at time step 0, and (ii) the interdependency relationships (IDRs) shared between the entities of the network.

In Algorithm 2 we present an approximation algorithm for the SPTSIP with Case II type IDRs.

Algorithm 2: Case II Approx. Algorithm for SPTSIP

Data:
1. Set of network entities $A \cup B$, with $n = |A|$ and $m = |B|$
2. A set of IDRs of the form $y \leftarrow \prod_{i=1}^{q} x_i$, where $x_i, y \in A \cup B, \forall 1 \le i \le q$
3. A set of *real targets* E', with $M = |E'|$

Result: Set of entities $E'' \subseteq A \cup B$ such that failure of E'' entities in time step 0 results in failure of E' entities by time step $p = n + m - 1$.

```
1  begin
2      U ← ∅;
3      DEP_i ← ∅, S_i ← ∅,   ∀i = 1, ..., M;
4      KillSet_j ← ∅,   ∀j = 1, ..., n + m;
5      foreach e_i ∈ E' do
6          foreach entity e_j ∈ IDR e_i ← ∏_{j=1}^{q} e_j do
7              DEP_i ← DEP_i ∪ {e_j};
8          U ← U ∪ {i};
9          S_i ← U ∪ {i};
10     foreach e_i ∈ A ∪ B do
11         KillSet_i ← KillSet(e_i);
12         for d = 1 to M do
13             if KillSet_i ∩ DEP_d ≠ ∅ then
14                 S_i ← S_i ∪ {d};
15     E'' ← ∅;
16     while U ≠ ∅ do
17         Select S_i, i = 1, ..., M that maximizes |S_i ∩ U|;
18         E'' ← E'' ∪ {e_i};
19         U ← U \ S_i;
20     return E''
```

Theorem 4. *The approximation solution produced by Algorithm 2 for Case II type IDRs is at most $O(ln(M))$ times the optimal, where $M = |E'|$*

Proof. Algorithm 2 implements a a greedy approach for solving a set cover problem. We set up the set cover problem the following way: First, in Steps 5-9, for each entity $e_i \in E'$ we construct dependency DEP_i as the set of entities out of which at least one entity must fail for e_i to fail, thus "unsatisfying" the dependency. In Step 9 we also account for dependency DEP_i getting unsatisfied due to failure of e_i itself. The universe U contains the indexes of each of these M dependencies. Next, in Steps 10-14, for each entity $e_i \in A \cup B$ we compute $KillSet(e_i)$ and construct set S_i that contains the index of the dependencies in $DEP_j, j = 1, \dots, M$ that has a non-empty intersection with $KillSet(e_i)$. This implies that with the failure of e_i and the ensuing failure propagation, DEP_j gets unsatisfied. With the universe set of U and the subsets $S_i, i = 1, \dots, M$, the greedy technique for set cover is used in Steps 16-19 that yields a known approximation factor of $O(ln(M))$ times the optimal solution [16]. □

Algorithm 2 Time Complexity: To construct the dependencies in Steps 5-9 at most M IDRs will be traversed each with at most $n + m$ entities hence these steps take $O(M(n + m))$ time. In Steps 10-14 computing the kill set of each of the $n+m$ entities and comparing it to each of the M dependencies of maximum size $n + m$ requires $O(M(n + m)^3)$ time. And finally, the greedy set cover in Steps 16-19 takes $O(M\ log(n+m))$. Overall the complexity of Algorithm 2 is $O(M(n+m)^3)$.

C. Case III: Problem Instance with an Arbitrary Number of Minterms of Size One

For Case III an IDR has the following form: $x_i \leftarrow \sum_{k_1=1}^{l} y_{k_1} + \sum_{k_2=1}^{q} x_{k_2}$ (with $x_i \neq x_{k_2} \forall x_{k_2}, 1 \le k_2 \le q$), where x_i, x_{k_2} are elements of set A (B) and y_{k_2} is an element of set B (A). The size of the minterm is given as $l + q$. In the example $a_r \leftarrow b_u + b_v + a_s$. $l + q = 3$, $x_i = a_r, y_1 = b_u, y_2 = b_v$ and $x_1 = a_s$.

Theorem 5. *The SPTSIP for Case III is NP Complete*

Proof. We prove that the SPTSIP for Case III is NP-complete by giving a transformation for the Vertex Cover (VC) problem [15]. An instance of the vertex cover problem is specified by an undirected graph $G = (V, E)$ and an integer R. In the vertex cover problem, one wants to know whether there is a subset $V' \subseteq V$ such that $|V'| \le R$, and for every edge $e \in E$, at least one end vertex of e is in V'. From an instance of the VC problem we create an instance of the SPTSIP in the following way: From the graph $G = (V, E)$ for each vertex $v_i \in V$ that has adjacent nodes (say) v_j, v_k and v_l, we create an IDR $v_i \leftarrow v_j + v_k + v_l$. We set the real target set $E' = V$ and $K = R$. It can now be verified that the instance of the VC problem has a vertex cover of size R, iff in the created instance of SPTSIP the failure of K entities at time step 0 triggers a cascade of failures resulting in the failure of all entities in the set E' by time step $p = |V| - 1$. □

D. Case IV: Problem Instance with an Arbitrary Number of Minterms of Arbitrary Size

This is the general case, where IDRs have arbitrary number of minterms of arbitrary size.

Theorem 6. *The SPTSIP for Case IV is NP Complete*

Proof. As both Case II and Case III are special cases of Case IV, the SPTSIP for Case IV is NP-Complete as well. □

IV. Algorithms for the SPTSIP

In this section we propose an optimal solution for the SPTSIP using Integer Linear Programming (ILP), and a polynomial time heuristic solution.

A. Optimal Solutions for the SPTSIP problem

We formulate an optimal solution for the SPTSIP with an ILP that uses two variables x_{it} and y_{jt}. Where $x_{it} = 1$, when entity $a_i \in A$ is in a failed state at time step t, and 0 otherwise. And, $y_{jt} = 1$, when entity $b_j \in B$ is in a failed state at time step t, and 0 otherwise.

The objective function can now be formulated as follows:

$$min\sum_{i=1}^{n}x_{i0}+\sum_{j=1}^{m}y_{j0} \quad (1)$$

Where $n = |A|$ and $m = |B|$. The constraints are as follows:

Failure Consistency Constraints: $x_{it} \geq x_{i(t-1)}, \forall t, 1 \leq t \leq p$, these constraints ensure that if an entity a_i fails at time step t, it continues to remain in a failed state for all subsequent time steps. A similar constraint applies for y_{it} variables [13].

Failure Propagation Constraints: These constraints govern the failure cascade process caused by the dependencies shared between the network entities. The correctness of these constraints is established in [13], we outline an overview of these constraints here for consistency. For any Case IV type IDRs of the form $a_i \leftarrow b_jb_kb_l + b_vb_u + b_q$ the subsequent steps are followed to model the failure propagation:

Step 1: Transform the IDR to a disjunctive form of size one minterms, i.e. $a_i \leftarrow c_1 + c_2 + b_q$.

Step 2: For each of the c type minterms create constraints to model the failure cascade for individual c type minterms, i.e. for $c_1 \leftarrow b_jb_kb_l$ introduce $c_{1t} \leq y_{j(t-1)} + y_{k(t-1)} + y_{l(t-1)}, \forall t, 1 \leq t \leq p$.

Step 3: For each transformed IDR from Step 1, for example $a_i \leftarrow c_1+c_2+b_q$, introduce a constraint of the form $N \times x_{it} \leq c_{1(t-1)} + c_{2(t-1)} + b_q, \forall t, 1 \leq t \leq p$, where N is the number of minterms in the transformed IDR, in this example $N = 3$.

Prior to the transformation of Step 1 if an IDR does not contain any disjunctions (Case II), then Step 3 is skipped, or if it does not contain any conjunctions (Case III), then Step 2 is skipped.

Real Target Set Failure Constraints: $x_{ip} = 1, \forall a_i \in E'$, and $y_{ip} = 1, \forall b_i \in E'$, these constraints ensure that all entities of the real target set E' are in a failed state at time step p.

Adhering to the above constraints, the objective in (1) minimizes the total number of entities that need to fail at time step 0 so that E' entities fail by time step p.

B. Heuristic Solution

We first outline the following definition:

Definition: *Kill Impact of a set of Entities* $\mathcal{P}$: The Kill Impact of a set of entities $\mathcal{P}$, denoted by $KillImpact(\mathcal{P})$, is defined as the contribution of $\mathcal{P}$ entities in causing the failure of entities in E'. It may be noted that any entity $e_i \in E'$ can fail due to two reasons: (i) when e_i itself fails at time step 0, or (ii) when at least one entity in all the minterms of e_i's IDR fail in some time step. $KillImpact(\mathcal{P})$ captures these two aspects by computing the impact of failure of $\mathcal{P}$ entities on E' based on: (i) the number of entities that fail in E' at time step p when $\mathcal{P}$ entities fail at time step 0, and (ii) the number of minterms in the IDR of each entity $e_i \in E'$ that get affected at time step p when $\mathcal{P}$ entities fail at time step 0. For a given set of $\mathcal{P}$ entities, and the set of minterms $MT_i = \{mt_1, mt_2, \ldots, mt_{|MT_i|}\}, mt_j \subseteq A \cup B$, for each entity $e_i \in E'$, to compute $KillImpact(\mathcal{P})$ we first compute $impact_i$ as the impact of failure of $\mathcal{P}$ on e_i as follows:

If $e_i \in KillSet(\mathcal{P})$, $impact_i = 1$, else if $e_i \notin KillSet(\mathcal{P})$:

$$impact_i = \frac{\left|\bigcup_{mt_j \cap KillSet(\mathcal{P}) \neq \emptyset} mt_j\right|}{|MT_i|}, \quad \forall mt_j \in MT_i$$

We then compute $KillImpact(\mathcal{P})$ as follows:

$$KillImpact(\mathcal{P}) = \frac{\sum_{i=1}^{|E'|} impact_i}{|\mathcal{P}|}$$

In Algorithm 3, we present a heuristic technique to solve the SPTSIP for the general case of the problem. The general approach for Algorithm 3 is to greedily select a set of entities that provide the maximum benefit towards reaching the objective of failing E'. In Steps 5-7, for each entity $e_j \in A \cup B$ we compute how frequently e_j appears in all minterms in the set of IDRs, we also compute $KillImpact(e_j)$. Next, in Steps 8-14, for each entity $e_i \in E'$, we examine each of the minterms of e_i's IDR and select the highest frequency entity of each minterm to construct set k_i and compute $KillImpact(k_i)$. In Step 15 we choose the most impactful set of entities from the total $KImpact$ sets constructed. Intuitively, this selection of a higher kill impact set for inclusion implies more failures in the target set. Also, since our objective is to minimize the size of the entities selected, a set with the largest impact to size ratio is preferred. Finally, the algorithm proceeds to update E'' and $failed$ set of entities, and prunes the IDR set and minterm set in Steps 16-19. This greedy selection process repeats until $E' \subseteq failed$. The heuristic ensures that for every iteration of the while loop in Step 3 the E'' set increases in such a way that at least one additional entity in E' fails than the previous iteration, thus moving closer to the objective. Algorithm 3 runs in polynomial time, specifically it runs in $O(M(n+m)^4)$ time, where $M = |E'|$. In Section V, our experiments show that Algorithm 3 almost always produces the optimal result.

V. Experimental Results

We now present experimental results for the SPTSIP and compare the optimal solution computed using an ILP, with the proposed heuristic algorithm. The experiments were conducted on power and communication network data of Maricopa County, Arizona. The power network data was obtained from Platts (www.platts.com), and the communication network data obtained from GeoTel (www.geo-tel.com). This data consisted of 70 power plants, 470 transmission lines, $2,690$ cell towers, $7,100$ fiber-lit buildings and $42,723$ fiber links. We identified five non-intersecting geographical regions, and from the consolidated power and communication network data of each region, we set up interdependencies between the network entities using the rules outlined in [13]. For continuity, we briefly outline an overview of these rules here: For each generator to be operational, either (i) the nearest cell tower must be operational, or (ii) the nearest fiber-lit building and the fiber link connecting the generator to the fiber-lit building must be operational. For each fiber-lit building and cell tower to be operational, at least one of the two nearest generators and the connecting transmission lines must be operational. The transmission lines and the fiber links have no dependencies.

Algorithm 3: Case IV Heuristic for SPTSIP

Data:
1. Set of network entities $A \cup B$, with $n = |A|$ and $m = |B|$
2. A set S of IDRs of type Case IV (general case)
3. A set of *real targets* E'

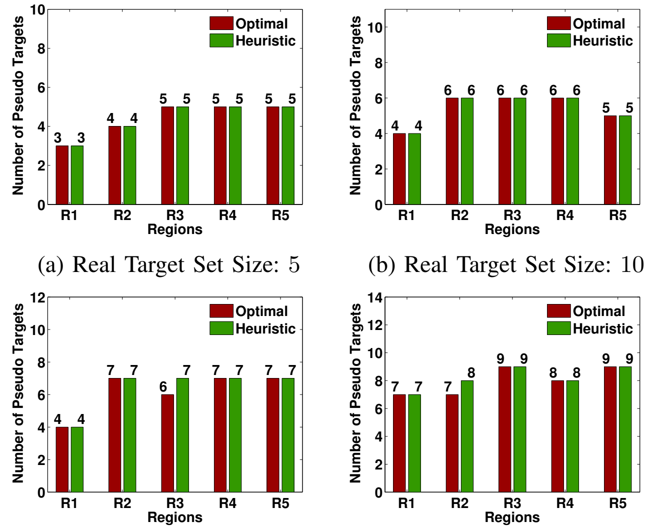
Result: A set of *pseudo targets* E'' such that when E'' fails at time step 0, the real target set E' fails by time step $p = n + m - 1$

begin
  $E'' \leftarrow \emptyset$, $failed \leftarrow \emptyset$;
  while $E' \not\subseteq failed$ **do**
    $KImpact \leftarrow \emptyset$;
    foreach *entity* $e_j \in A \cup B$ *and* $e_j \notin failed$ **do**
      Compute $frequency_j$ as the number of times e_j appears in a minterm for all IDRs in S;
      $KImpact \leftarrow KImpact \cup (e_j, KillImpact(e_j))$;
    foreach *entity* $e_i \in E'$ *and* $e_i \notin failed$ **do**
      Let idr in S be the IDR of entity e_i;
      $k_i \leftarrow \emptyset$;
      foreach *minterm* MT *in* idr **do**
        Select entity $e_j \in MT$ with largest $frequency_j$ from all entities in MT;
        $k_i \leftarrow k_i \cup e_j$;
      $KImpact \leftarrow KImpact \cup (k_i, KillImpact(k_i))$;
    Select tuple $(failSet, failVal) \in KImpact$ where $failVal \geq val, \forall (set, val) \in KImpact$;
    $E'' \leftarrow E'' \cup failSet$;
    $failed \leftarrow KillSet(E'')$;
    Remove IDR of entity e_k from S, $\forall e_k \in failed$;
    For each IDR in S remove all minterms that contain entity e_k, $\forall e_k \in failed$;
  return E''

The optimal solutions were obtained by solving Integer Linear Programs using the IBM CPLEX Optimizer 12.5. For each of the five regions $R1$ through $R5$, real target sets of entities of sizes 5, 10, 15 and 20 were chosen from the set of all power and communication entities of that region. For each real target set the optimal and heuristic solutions were computed, and these results are presented in Fig. 1. Our experiments showed that for the five regions considered, in the worst case the heuristic solution differed from the optimal by a factor of 0.16, in the best case was equal to the optimal, and on an average was within a factor of 0.02 of the optimal solution.

VI. Conclusion

In this paper we presented the *Smallest Pseudo Target Set Identification Problem* (SPTSIP) for targeted attack on IPCN. We used the IIM to model interdependencies between the two networks and classified the problem into four classes. We showed that the problem is solvable in polynomial time for the first class, whereas for others it is NP-complete. We provided an *approximation algorithm* for the second class, and for the general class we provided an optimal solution using ILP, and a heuristic technique. Finally, we evaluated the efficacy of our heuristic using power and communication network data of Maricopa County, Arizona. Our experiments showed that our heuristic almost always produced near optimal results.



(a) Real Target Set Size: 5 (b) Real Target Set Size: 10 (c) Real Target Set Size: 15 (d) Real Target Set Size: 20

Fig. 1: Comparison of optimal and heuristic approaches for computing pseudo targets (E''), for given real targets (E') of sizes 5, 10, 15 and 20, on five geographical regions of Maricopa County, Arizona.

References

[1] S. V. Buldyrev, R. Parshani, G. Paul, H. E. Stanley, and S. Havlin, "Catastrophic cascade of failures in interdependent networks," *Nature*, vol. 464, no. 7291, pp. 1025–1028, 2010.

[2] J. Gao, S. V. Buldyrev, H. E. Stanley, and S. Havlin, "Networks formed from interdependent networks," *Nature Physics*, vol. 8, no. 1, 2011.

[3] J. Shao, S. V. Buldyrev, S. Havlin, and H. E. Stanley, "Cascade of failures in coupled network systems with multiple support-dependence relations," *Physical Review E*, vol. 83, no. 3, p. 036116, 2011.

[4] V. Rosato, L. Issacharoff, F. Tiriticco, S. Meloni, S. Porcellinis, and R. Setola, "Modelling interdependent infrastructures using interacting dynamical models," *International Journal of Critical Infrastructures*, vol. 4, no. 1, pp. 63–79, 2008.

[5] P. Zhang, S. Peeta, and T. Friesz, "Dynamic game theoretic model of multi-layer infrastructure networks," *Networks and Spatial Economics*, vol. 5, no. 2, pp. 147–178, 2005.

[6] J.-F. Castet and J. H. Saleh, "Interdependent multi-layer networks: Modeling and survivability analysis with applications to space-based networks," *PloS one*, vol. 8, no. 4, p. e60402, 2013.

[7] M. Parandehgheibi and E. Modiano, "Robustness of interdependent networks: The case of communication networks and the power grid," in *Global Communications Conference (GLOBECOM)*. IEEE, 2013.

[8] M. Parandehgheibi, E. Modiano, and D. Hay, "Mitigating cascading failures in interdependent power grids and communication networks," in *Smart Grid Communications (SmartGridComm), 2014 IEEE International Conference on*. IEEE, 2014, pp. 242–247.

[9] C.-C. Liu, A. Stefanov, J. Hong, and P. Panciatici, "Intruders in the grid," *Power and Energy Magazine, IEEE*, vol. 10, no. 1, pp. 58–66, 2012.

[10] D. T. Nguyen, Y. Shen, and M. T. Thai, "Detecting critical nodes in interdependent power networks for vulnerability assessment," 2013.

[11] A. Bernstein, D. Bienstock, D. Hay, M. Uzunoglu, and G. Zussman, "Power grid vulnerability to geographically correlated failures-analysis and control implications," *arXiv preprint arXiv:1206.1099*, 2012.

[12] M. Korkali, J. G. Veneman, B. F. Tivnan, and P. D. Hines, "Reducing cascading failure risk by increasing infrastructure network interdependency," *arXiv preprint arXiv:1410.6836*, 2014.

[13] 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," in *Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2014, pp. 831–836.

[14] A. Das, J. Banerjee, and A. Sen, "Root cause analysis of failures in interdependent power-communication networks," in *Military Communications Conference (MILCOM), 2014 IEEE*. IEEE, 2014, pp. 910–915.

[15] M. R. Garey and D. S. Johnson, "Computer and intractability," *A Guide to the NP-Completeness. New York, NY: WH Freeman and Co.*, 1979.

[16] J. Kleinberg and É. Tardos, *Algorithm design*. Pearson Education, 2006.