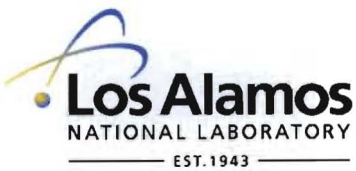
LA-UR- 10-07071

Approved for public release;
distribution is unlimited.

Title:	Detecting and Mitigating Abnormal Events in Large Scale Networks: Budget Constrained Placement On Smart Grids
Author(s):	Nandakishore Santhi and Feng Pan
Intended for:	Presentation at 44th Hawaii International Conference on System Sciences, (HICSS-44), to be held at Koloa, Kauai, HI from January 4-7, 2011.



Form 836 (7/06)

Detecting and Mitigating Abnormal Events in Large Scale Networks: Budget Constrained Placement On Smart Grids

Nandakishore Santhi
Computer and Computational Sciences Division
Los Alamos National Laboratory
Los Alamos, NM 87545
nsanthi@lanl.gov

Feng Pan
Decision Applications Division
Los Alamos National Laboratory
Los Alamos, NM 87545
fpan@lanl.gov

Abstract

Several scenarios exist in the modern inter-connected world which call for an efficient network interdiction algorithm. Applications are varied, including various monitoring and load shedding applications on large smart energy grids, computer network security, preventing the spread of Internet worms ad malware, policing international smuggling networks, and controlling the spread of diseases. In this paper we consider some natural network optimization questions related to the budget constrained interdiction problem over general graphs, specifically focusing on the sensor/switch placement problem for large-scale energy grids. Many of these questions turn out to be computationally hard to tackle. We present a particular form of the interdiction question which is practically relevant and which we show as computationally tractable. A polynomial-time algorithm will be presented for solving this problem.

1. Introduction

In today's inter-connected world, it is often necessary to maintain open energy, communication and transportation networks. However in the interest of fair use, it is also important to keep these networks safe while at the same time preventing catastrophic events and malicious attacks. This has to be achieved in the most non-intrusive manner possible and done using minimal additional infrastructure in a robust as well as distributed manner while simultaneously meeting budget constraints for the the cost of installation and operation.

Applications which require such *interdiction*, include future smart energy grids where dynamic load balancing will be crucial, computer network security applications where firewalls need to be setup to control the spread of Internet worms and malware, quarantine planning for controlling the spread of diseases [2], as well as policing drug [11] and nuclear smuggling networks [7].

A formal model for this practical problem is a network interdiction model, where interdiction is performed along the edges (or equivalently on the nodes) of a graph which represents the distribution, communication or transportation network in sufficient detail. In this paper, without loss of generality we will be considering an edge interdiction model on a directed network graph. The model we are about to introduce is primarily motivated by the problem of optimal sensor and switch placement for smart grid usage monitoring and control. An objective will be to optimally allocate resources to maximize the detection probability of abnormality in a network. We also show in Section 2, how the same model can be used to optimally place control switches so as to minimize the response time in the event of emergency load management on electricity grids.

The interdiction problem is an active research area in operations research and theoretical computer science. Several researchers have in the past considered interdiction in various forms [3, 6, 7, 9]. However many of these formulations are known to be computationally intractable for even modestly sized networks [12]. Most of the suggested solution methods involve some form of integer linear programming which is usually computationally costly. Cutting plane methods and sub-optimal linear programming relaxations have also been proposed in the literature [8].

In this paper, a network interdiction model, *budget constrained single edge interdiction*, is proposed. This model is closely related to the classical interdiction models, while being computationally tractable (as proved in a later section) unlike the classical models. In Section 3, we will formally define the single-edge interdiction model and two related, but so far intractable classical interdiction models. Our single-edge interdiction model is intuitively motivated by the following maxim: *The weakest link breaks the chain.* A polynomial-time algorithm is then developed based on an auxiliary graph

constructed from the original graph in Section 4.

2. Optimization Models for Effective Placement on Smart Grids

Before introducing interdiction models to detect and mitigate anomalous events in networks, we outline several potential applications in this section. The general framework for the interdiction model is as follows: In a network $G(V, E)$, on various edges $e \in E$ on the network graph, let us model the probability of an anomalous event evading detection from the sensor installed on that edge by a parameter called the *edge evasion probability*, π_e. The detection probability δ_e is the complement of the evasion probability as $\delta_e = 1 - \pi_e$. In most natural cases, the evasion probabilities on various edges can be modeled to be statistically independent, which means that on any path p on the network, the effective evasion probability π_p is given by the product, $\pi_p = \prod_{e \in p} \pi_e$. A limited number of sensors (switches) are installed on edges and the goal is to find a placement of the sensors (switches) such that the maximum detection probability (shortest response time) can be achieved.

The primary applications that we foresee for our interdiction model are in the optimal placement of sensors and switches for monitoring and control over new generation infrastructure for power grids. Sensor placement on power grids to detect patterns of anomalous spikes and excessive consumption presents several new challenges. Given an anomalous event happening on an edge, the probability of detecting such an event depends on various physical properties of the power line, the power flow at a given time on the line, and the environment surrounding the line. The detection rate will depend on the amount of data collected from all sensors within a period of time. In this case, the budgetary constraint on sensors is imposed by how much data can be stored/processed during a given detection period. We propose the use of the polynomial time interdiction algorithm introduced later in this paper to solve this sensor placement problem efficiently.

An application with a similar flavor is firewall placement on packet communication networks. These communication networks can be the Internet or a proprietary SCADA control network commonly encountered in the electrical energy sector. Worms and virus can propagate through the network along a set of paths $\mathcal{P}$. These abnormal activities can be detected by interdiction resources, e.g., specialized firewalls. Placement of a firewall on a node is equivalent to placement on an edge in terms of mathematical modeling – splitting a node into two nodes and adding an edge between the two nodes, and interdicting the newly created edge. Due to this equivalence hereafter, resources are assumed to be only allocated to edges. The *detection probability*, denoted as δ_e, of catching a malicious activity depends on a number of characteristics of the edge e, for example the load and hardware. We further assume that the detection event at a given edge is statistically independent of that on any other edge. The overall detection probability on a path $p \in \mathcal{P}$ is then $1 - \prod_{e \in p}(1 - \delta_e)$. The goal is to install a given number (representing the available budget) of firewalls on the network to achieve the maximum detection probability.

Yet another type of application exists in smart energy grid system design. This concerns the placement of load management (shedding/transfer) switches on a grid for emergency control of consumption. Fast, optimal power shedding is often crucial in preventing large scale and uncontrolled blackouts which can lead to massive financial and resource losses on the one hand, and unnecessary load tripping resulting in productivity loss on the other [10]. In some cases, critical loads which cannot tolerate extended power-cuts have to be switched over to alternate sources. In either scenario, the objective is to reduce the response time in effecting a safe load management given a certain budget.

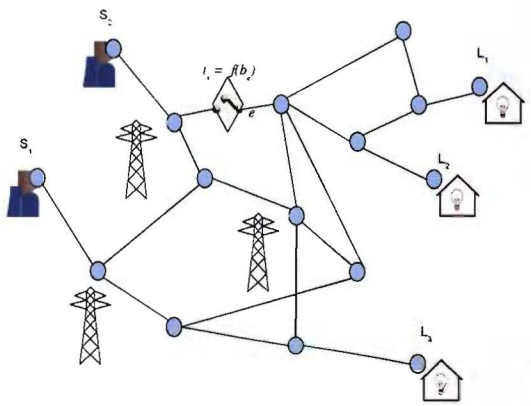


Figure 1. An illustrative power grid with two power generators and three loads. Switching equipment with switching times $t_s = f(b_e)$ are placed at a cost of b_e on each edge e.

Load management is usually achieved by a remotely operated switch with a design parameter, *switching time*, t_s which is a factor contributing to the overall response time. Another factor contributing to the response time is the delay in operating the switch which is inherent to the automatic or manual control system in

use. The response time of a switch also changes with the technology used, the type of switch (make-before-break, break-then-make etc) and constraints to be met such as phase/frequency synchronization. In this context, in what follows in the next section, b_e represents an edge budget used to install a load switching equipment and the cost incurred (which also includes loss of revenue and cost of equipment failures) in achieving a response time of $f_e(b_e)$. In general, the higher the allowed budget, the lower is the achievable response time. In this case, the objective is to switch loads as fast as possible, i.e., achieving a minimum response time on any generator-load path while staying within a maximum budget constraint: $\sum_e b_e \leq B$. A typical network with load switching elements and multiple power sources and sinks is shown in Figure 1. The resulting optimization problem is formally stated as Question 3 in section 3.

These seemingly unrelated problems can be formulated in quite general terms as network interdiction problems. A common objective in interdiction problems is to determine an optimal allocation of budgets for installation of interdiction apparatus on individual edges such that the effective evasion probability is minimized while simultaneously satisfying some total budget constraints. In the next section, we give a formal definition of the budget constrained network interdiction problem.

3. Budget Constrained Single Edge Interdiction

In this section we consider a few most commonly encountered versions of the network interdiction problem. We then consider a particular model (Question 3 below), which is most relevant in the smart-grids context, deriving an efficient polynomial-time optimization algorithm for it.

Definition 1 (BC-INT, BC-AV-INT, BC-SE-INT)
Instance: *A directed network graph $\mathcal{G}(\mathcal{V},\mathcal{E})$; a set of efficiently computable monotonic non-increasing local budget-evasion-probability functions $f_e : \mathbb{R}^+ \mapsto [0,1]$ associated with each directed edge $e \in \mathcal{E}$; two non-empty subsets of $\mathcal{V}$, the source nodes $\mathcal{S}$ and the destination nodes $\mathcal{D}$, such that $\mathcal{S} \cap \mathcal{D} = \emptyset$; and a total interdiction budget B.*
Question 1 (BC-INT): *Find a budget assignment to each edge, b_e which satisfies the total budget constraint $\sum_{e\in\mathcal{E}} b_e \leq B$, and minimizes,*

$$\pi_{MAX} \stackrel{\text{def}}{=} \max_{p^i_{(s,d)} \in \mathcal{P}(\mathcal{S},\mathcal{D})} \prod_{e_j \in p^i_{(s,d)}} f_{e_j}(b_{e_j})$$

Question 2 (BC-AV-INT): *Find a budget assignment to each edge, b_e which satisfies the total budget constraint $\sum_{e\in\mathcal{E}} b_e \leq B$, and minimizes,*

$$\pi_{AV} \stackrel{\text{def}}{=} \sum_{p^i_{(s,d)} \in \mathcal{P}(\mathcal{S},\mathcal{D})} w_{p^i_{(s,d)}} \cdot \prod_{e_j \in p^i_{(s,d)}} f_{e_j}(b_{e_j})$$

Question 3 (BC-SE-INT): *Find a budget assignment to each edge, b_e which satisfies the total budget constraint $\sum_{e\in\mathcal{E}} b_e \leq B$, and minimizes,*

$$\pi \stackrel{\text{def}}{=} \max_{p^i_{(s,d)} \in \mathcal{P}(\mathcal{S},\mathcal{D})} \min_{e_j \in p^i_{(s,d)}} f_{e_j}(b_{e_j})$$

where $\mathcal{P}(\mathcal{S},\mathcal{D})$ is the set of all directed paths $p^i_{(s,d)}$ from some node in $\mathcal{S}$ to some node in $\mathcal{D}$, $w_{p^i_{(s,d)}}$ are positive weights associated with these paths such that $\sum_{p^i_{(s,d)}} w_{p^i_{(s,d)}} = 1$, and e_j represents a directed edge in the directed path $p^i_{(s,d)}$.

In the above definition, the local budget-evasion-probability functions $f_e(\cdot)$ can be interpreted as follows: given a local edge budget of b_e for edge e, we can achieve an evasion probability (or equivalently, the overall response time for switches) of $f_e(b_e)$ at that edge. Very often in practice, the local functions f_e could be made to subsume other more complex characteristics on the network too.

For example, if in a network with a single source and destination, there are already in place other interdiction apparatus, which ensures evasion probabilities less than 1 on certain edges. Then, we may wish to calculate the residual evasion probability before installing any new apparatus by first running a Dijkstra type shortest path algorithm. Let each edge $e = (i,j)$ have a prior evasion probability of α_e. Also let us assume for example that by installing N_e apparatus of unit cost, the post-installation edge evasion probability can be reduced to $\alpha_e \cdot \beta_e^{N_e}$. Then we may wish to set as a first order approximation, $f_e(N_e) = \alpha_e \cdot \beta_e^{N_e} \cdot \prod_{e'_s \in p(s,i)} \alpha_{e'_s} \cdot \prod_{e'_d \in p(j,d)} \alpha_{e'_d}$. Here, $p(s,i)$ is the shortest path from source node s to node i when the edges e' are labeled with non-negative edge weights of $(-\log \alpha_{e'})$. Similarly $p(j,d)$ is the shortest path from node j to the destination node d.

All the three forms of interdiction problems can be seen to be practically relevant in various contexts. However, even for the simplest local functions f_e, the problems posed in questions 1 and 2 above are known to be NP-complete even to approximate within a constant factor, by a polynomial time reduction from the relatively well known VERTEX-COVER and CLIQUE problems [5]. For a simple proof of this reduction, see [12].

In this paper therefore, we will focus solely on Question 3. Since the local functions f_e can be heavily non-linear, it is not apriori clear that the problem in Question 3 admits a polynomial time solution. We present one such solution in the next section.

One may justify posing Question 3 in favor of the other two versions in many situations. In the switch placement problem minimizing the response time for load management given a limited budget, Question 3 is definitely the appropriate model. This is because the response time for any source–load pair is given by the maximum response time on any source–load path, while for a fixed source–load path, the response time is given by the fastest response among all links in that path.

In problems where non-zero evasion probabilities have to be avoided at all costs (for example in the case of preventing nuclear smuggling or protecting a crucial SCADA system), interdiction apparatus at edge e can be reasonably modeled as requiring a cost of b_e to ensure $\pi_e = 0$. In this case, solving Question 3 is equivalent to solving Question 1, whereas Question 2 is perhaps not practically relevant (since it is the worst case evasion probability that matters, not the average case). In many other instances, it is usually the case that the evasion probability that can be achieved is so small that a solution for Question 3 is practically very close to that of Question 1. Moreover, the availability of an efficient algorithm is clearly a factor to be considered. Typical solutions to interdiction problems would otherwise rely on the solution of cumbersome integer-linear-programs, which are often computationally intractable even for medium scale networks.

4. A polynomial time algorithm for BC-SE-INT

In order to derive an algorithm for BC-SE-INT, we will assume that the local functions are efficiently invertible - that is, $f_e^{-1}(\cdot)$ can be computed in polynomial-time. There is no loss in generality due to this assumption, since in virtually all practical scenarios this is true - moreover in the event of there being no analytical form for the inverse function, a table look-up based approach can be easily implemented. A pseudo-code for the proposed algorithm BC-SE-INT-ALGO is listed as Algorithm 1.

5. Correctness and complexity of BC-SE-INT-ALGO

To see that the algorithm BC-SE-INT-ALGO produces the correct result to an accuracy of better that an additive factor of ϵ, we can note the following. Since $\mathcal{S} \cap \mathcal{D} = \emptyset$, any $(s,d)-$path should contain at least one interdictable edge. Moreover, since the local functions are monotonic non-decreasing, an increased local budget will not increase the edge's evasion probability.

The linear program in step 5 is well known to have an integral polyhedron, so that at the solution, $x_e \in \{0,1\}$. This can be easily seen considering the following probabilistic argument: If $\bar{y}_i$ is a fractional point in the solution, let us use the following randomized procedure - generate a uniform random variable u, then set

Algorithm 1 Budget Constrained Single Edge Interdiction Algorithm (BC-SE-INT-ALGO)

INPUT:

A network graph $\mathcal{G}(\mathcal{V},\mathcal{E})$ to be interdicted along with the local functions f_e for all $e \in \mathcal{E}$, a total budget B and a tolerance value $\epsilon > 0$.

STEPS:

1. Set $\pi^1 \leftarrow 1$ and $\pi^0 \leftarrow 0$.
   Augment the original network graph to obtain a new graph $\mathcal{G}'(\mathcal{V}',\mathcal{E}')$ in the following way:
   Create a new node s and connect it to all nodes $s_i \in \mathcal{S}$ with new directed edges $e_{(s,s_i)}$. Similarly, create a new node d and connect it to all nodes $d_i \in \mathcal{D}$ with new directed edges $e_{(d_i,d)}$. All newly created edges e are marked non-interdictable, that is $f_e^{-1}(x) \mapsto \infty$ for $x \in [0,1)$.
2. while $(\pi^1 - \pi^0 > \epsilon)$ do {
3. Set $\pi' \leftarrow \frac{\pi^1+\pi^0}{2}$.
4. for all $e \in \mathcal{E}'$, compute $b'_e = f_e^{-1}(\pi')$.
5. Solve the linear program:
   Minimize, $\sum_{e\in\mathcal{E}} b'_e \cdot x_e$ subject to,

$$x_e \geq (y_i - y_j); \qquad x_e \geq (y_j - y_i)$$
$$y_s = 1, \quad y_d = 0; \qquad 0 \leq y_i, x_e \leq 1$$

   Let B' be the minimum attained.
   Let $\mathcal{E}' \supseteq \mathcal{C} \stackrel{\text{def}}{=} \{e : x_e = 1\}$.
6. if $(B' > B)$ set $\pi^0 \leftarrow \pi'$.
7. else if $(B' < B)$ set $\pi^1 \leftarrow \pi'$.
8. }
9. Set $\pi \leftarrow \pi^1$.
   For all $e \in \mathcal{C}$, set $b_e \leftarrow b'_e$ and for all $e \in \mathcal{E}\backslash\mathcal{C}$, set $b_e \leftarrow 0$.

OUTPUT:

The solution π and an associated set of edge budgets $\{b_e : e \in \mathcal{E}\}$

$y_i \leftarrow 0$ if $\bar{y}_i < u$ and set $y_i \leftarrow 1$ otherwise. Now,

$$\begin{aligned} B' &\leq \mathbb{E}\left(\sum_{e\in\mathcal{E}} b'_e \cdot x_e\right) \\ &= \sum_{e=(i,j)} b'_e \cdot \Pr\left(u \in [\min\{\bar{y}_i, \bar{y}_j\},\ \max\{\bar{y}_i, \bar{y}_j\}]\right) \\ &= \sum_{e=(i,j)} b'_e \cdot |\bar{y}_i - \bar{y}_j| = \sum_e b'_e \cdot \bar{x}_e = B' \end{aligned}$$

Therefore step 5 finds a minimum budget interdiction cut on the original network graph such that on any $(s,d)-$path, at least one edge has evasion probability less that π'. Moreover the interdiction cut cannot involve any of the fictitious non-interdictable edges introduced in step 1. Furthermore, the monotonous property of the local functions f_e implies that an optimal interdiction cut resulting in a higher budget B', cannot have a higher evasion probability π'. Therefore each iteration of the loop from step 2 to step 8 reduces the search region for π by half at either of the steps 6 or 7, while satisfying the budget constraint and will therefore terminate with the correct solution in $\mathcal{O}(\log 1/\epsilon)$ iterations.

To estimate the complexity of BC-SE-INT-ALGO, for a precision as required by the constant ϵ, the loop from step 2 to step 8 is executed $\mathcal{O}(\log 1/\epsilon)$ times, which is again a constant. We can further improve the algorithm by substituting for the linear program in step 5 any well known algorithm for max-flow, since max-flow and min-cut are related by linear programming duality [4]. Each iteration of this loop requires a polynomial amount of time, which depends on the $(s-d)-$min-cut algorithm employed. Using an efficient max-flow algorithm as in [1], which has a complexity of $\mathcal{O}(|\mathcal{V}| \cdot |\mathcal{E}| + |\mathcal{V}|^2 \log |\mathcal{V}|)$, each iteration takes $\mathcal{O}(r|\mathcal{E}| + \mathcal{O}(|\mathcal{V}| \cdot |\mathcal{E}| + |\mathcal{V}|^2 \log |\mathcal{V}|))$ time, where r denotes the time required for computing the inverse function $f_e^{-1}(\cdot)$ to the required precision.

6. Conclusion

We considered the important system design problem of budget constrained interdiction which has a variety of applications in diverse areas such as smart power grids, sensor networks, law enforcement and surveillance. We focused on the efficient placement problem for load control equipment and monitoring sensors on smart grids; formulating an equivalent optimization problem covering such scenarios. We showed that this optimization problem is tractable – unlike other common variations of the interdiction problem which are typically computationally hard. We derived an algorithm which finds an optimal solution (up to any given small constant, ϵ) to the problem we consider. Simulation results using a C implementation of our algorithm were very promising - large power networks which were typically not amenable to brute force integer programming approaches have yielded meaningful solutions while using up only reasonable computation times.

Problems of future interest include scenarios where simultaneous optimization is required over several cost functions and under multiple budget constraints. Also of interest are networks where multiple commodities are transacted. Further improvements in running time are of definite interest, as are faster approximation algorithms for use with extremely large networks. Algorithms which adapt to dynamic changes in evasion probabilities as well as models which consider statistical dependence and other stochastic variables are also of interest.

7. References

[1] R. K. Ahuja, and J. B. Orlin, "A Fast and Simple Algorithm for the Maximum Flow Problem," *Operations Research*, 37(5), 1989, pp. 748–759.

[2] N. Assimakopoulos, "A network interdiction model for hospital infection control," *Comput Biol Med*, 1987, 17(6), pp. 413–22.

[3] E. P. Durbin, "An Interdiction Model of Highway Transportation, RM-4945-PR" *Rand Corporation*, Santa Monica, CA, May, 1966.

[4] P. Elias, A. Feinstein, and C. E. Shannon, "A Note on maximum flow through a network," *IEEE Trans. on Information Theory*, IT(2), 1956, pp. 117–119.

[5] M. R. Garey, and D. S. Johnson, *Computers and Intractability*, W. H. Freeman and Co., San Francisco, 1979.

[6] A. W. McMasters, and T. M. Mustin, "Optimal Interdiction of a Supply Network," *Naval Research Logistics Quarterly*, 17(3), 1970, pp. 261–268.

[7] D. P. Morton, F. Pan, and K. J. Saeger, "Models for Nuclear Smuggling Interdiction," *IIE Transactions*, Jan. 2006.

[8] F. Pan, and D. P. Morton, "Minimizing a stochastic maximum-reliability path," *Networks*, 52(3), 2008, pp. 111–119.

[9] F. Pan, W. Charlton, and D. P. Morton, "Interdicting smuggled nuclear material," *Network Interdiction and Stochastic Integer Programming*, D. L. Woodruff (ed.), Kluwer, Boston, MA, 2003.

[10] S. Shokooh, K. R. Wood, T. Khandelwal, F. Shokooh, J. Tastet, and J.J. Dai, "Intelligent Load Shedding Need for a Fast and Optimal Solution," *IEEE PCIC Europe*, 2005.

[11] A. Washburn, and K. R. Wood, "Two-person zero-sum games for network interdiction," *Operations Research*, 43(2), 1995, pp. 243–251.

[12] R. K. Wood, "Deterministic Network Interdiction," *Mathematical and Computer Modeling*, 17(2), 1993, pp. 1–18.