Minimizing Expected Attacking Cost in Networks

Anis Gharbi, Azaiez Mohamed Naceur, Mohamed Kharbeche

Minimizing Expected Attacking Cost in Networks

Anis Gharbi [1,2], Mohamed Naceur Azaiez [3]

PFARCAMT, Department of Industrial Engineering, College of Engineering
King Saud University, Riyadh, Saudi Arabia

Mohamed Kharbeche [4]

UMR CNRS 6599 Heudiasyc, Centre de Recherches de Royallieu
Université de Technologie de Compiègne, France

ROI - Combinatorial Optimization Research Group
Ecole Polytechnique de Tunisie, La Marsa, Tunisia

Abstract

A branch-and-bound algorithm is devised to determine the optimal attack strategy to disconnect a network where the objective is to minimize the expected attacking cost. The attacker cannot launch an attack if its cost is beyond his available budget or its probability of success falls below a threshold level. The proposed branch-and-bound algorithm includes, among other features, a dynamic programming-based lower bound as well as a preprocessing algorithm which aims at identifying unattackable links and removing irrelevant ones. Extensive use of the min-cut algorithm is made to derive valid upper bounds and to perform feasibility tests. Preliminary numerical implementation shows potential to provide exact solutions for medium-sized networks within reasonable time.

Keywords: network attack, branch-and-bound, minimum cut.

1 Introduction

Research on protecting infrastructure and targeted systems from intelligent threats is gaining increasing interest nowadays. Several models of defense-attack strategies have recently been suggested (see for instance Bier and Azaiez [2]). Most of these models arise in a game-theoretic context combined with reliability theory/risk analysis so that the best defensive strategies account for optimal attack strategies in order to be effective and to protect against the worst that could happen.

Network models represent an important area for application. For example, we may be concerned about preserving the functionality of electricity transmission and distribution systems, or about ensuring the existence of a viable transportation route from one major city to another, in situations where potential attackers may be able to observe some or all of our defenses and adapt their strategies accordingly. Other applications may include telecommunication, computer, underground transportation, and oil pipeline networks.

The current work falls in the area of defense-attack strategies of a network where some flow is to be sent from a source node to a destination node. The attacker is interested in disconnecting the network by fully disabling an entire cut of the network preventing the flow to reach its final destination. The defender problem would be to determine optimal strengthening policies of the network links (by increasing the reliability of the "most critical links") to enable the network to survive to potential attacks. Both attackers and defenders are submitted to resources constraints. In this paper, we address exclusively the attacker problem.

2 Statement of the problem

Consider an undirected network $N = (V, E)$ which is targeted by an attacker who wants to fully disconnect it from its source s to its destination t. To each link (i, j) in E are associated an attacking cost $c_{i,j}$, and a survival probability upon attack $p_{i,j}$. The attacker is deterred if the total attacking cost exceeds

[1] This research has been supported by the College of Engineering Research Center of King Saud University under the grant number 27-429. The authors are grateful to the Princess Fatimah Al Nijriss Research Chair for Advanced Manufacturing Technologies for providing partial support.
[2] Email: a.gharbi@ksu.edu.sa
[3] Email: mnazaiez@ksu.edu.sa
[4] Email: mohamed.kharbeche@hds.utc.fr

his available budget B (budget constraint), or if its probability of success is below a threshold value P (confidence constraint). It is assumed that at most one attack could be launched per link. The attack will be stopped as soon as a cut set is fully disabled (i.e., the network is disconnected) or if the attacker realizes that he cannot fully disable a cut set. The attacker problem could be stated as follows: Which among the links of the network to attack in order to disconnect the network under budget and confidence constraints so that the expected attacking cost is minimized? In addition, in which order the links of the selected cut set will be attacked?

Note that disabling an entire cut consists on disabling all the links of the cut set, which is equivalent to disabling a parallel system. In this context, the results of Azaiez and Bier [1] can be used to show that, given a cut set S, the optimal attack strategy is to sequentially attack the links $(i_1, j_1), (i_2, j_2), ..., (i_{|S|}, j_{|S|})$ in the non-decreasing order of their $c_{i,j}/p_{i,j}$. In that case, the least expected attack cost is given by

$$Z(S) = c_{i_1,j_1} + \sum_{k=2,...,|S|} \left(\prod_{h=1,...,k-1} (1 - p_{i_h,j_h}) \right) c_{i_k,j_k}$$

A link (i, j) of a cut set S is said to be *redundant* if $S\backslash\{(i, j)\}$ is a cut set. It should be clear that the optimal cut must be a non-redundant cut set (i.e. not including any redundant link). In this paper, a branch-and-bound algorithm is proposed to determine a non-redundant cut set S^* such that $Z(S^*) = \min_{S \subseteq E} Z(S)$. The problem is NP-hard since finding a feasible bi-criteria $s - t$ cut is already NP-complete [3]

3 Solution Procedure

The procedure consists of a sequence of decisions on the different links taken in the non-decreasing order of $c_{i,j}/p_{i,j}$ in consistency with the above result. At each stage of the search tree, a decision will be made on whether or not a particular link is selected in the generation of the cut set (to be potentially attacked). If the decision is to "select", then that link will be removed from a set called S_0, which initially includes all the links of the network, and will be placed in a set called S_1. Otherwise, the link will be considered as "unattackable". In this case, the attacking cost of this link will be set to infinity and its survival probability will be set to 1. Moreover, it will be moved to another set called S_2. At each stage, a cut-finding test is made to verify whether the links generated in S_1 already form a cut set. Clearly, if a cut is obtained then there is no need to proceed further from that stage. In addition, feasibility

tests are performed at every stage. These tests deal with budget and confidence constraints in addition to the possibility of proving that no cuts could be found. They make extensive use of the min-cut algorithm with varying assignments of values. For the solution procedure to gain more in computational efficiency, a preprocessing algorithm is devised in order to eliminate at each stage links that are guaranteed not to belong to an optimal cut. Three upper bounding schemes are proposed in order to derive non-redundant feasible cut sets. Three lower bounds (including a dynamic programming-based one) with varying degrees of complexity and effectiveness are suggested. The depth-first search strategy is adopted. In the sequel, we provide the details of each component of the proposed branch-and-bound algorithm. Due to lack of available space, the outline of the branch-and-bound algorithm has been omitted.

Preliminary identification of unattackable links: Obviously, any link $(i_0, j_0) \in S_0$ which satisfies $\sum_{(i,j)\in S_1} c_{i,j} + c_{i_0,j_0} > B$ or $(1 - p_{i_0,j_0}) \prod_{(i,j)\in S_1}(1 - p_{i,j}) < P$ cannot be selected and will move to S_2. This simple test substantially reduces the size of S_0 and improves the performance of the computed lower bounds, yielding a faster convergence of the whole optimization procedure.

Sufficient conditions for identifying irrelevant arcs: In this section, it is assumed that each link (i, j) is replaced by two directed arcs (i, j) and (j, i). It is worth noting that the flow value on some particular arcs of $S_0 \cup S_2$ can be equal to zero for any feasible $s - t$ flow. That is, such arcs will never be used by the defender to send any flow from s to t. These arcs are referred to as *irrelevant* arcs and can be completely removed from the network. The following sufficient conditions for an arc to be irrelevant clearly hold: (a) if a node $j \neq t$ has no successors (except perhaps node i), or all arcs emanating from node j (except perhaps (j, i)) are irrelevant, then (i, j) is irrelevant; (b) if a node $j \neq s$ has no predecessors (except perhaps node i), or all arcs entering node j (except perhaps (i, j)) are irrelevant, then (j, i) is irrelevant.

In order to identify arcs which satisfy one of the above irrelevance conditions, we devised the following linear program where $x_{ij} = 0$ if (i, j) is irrelevant, and $x_{ij} = 1$ otherwise.

$$Maximize \sum_{(i,j)\in S_0\cup S_2} x_{ij} \tag{1}$$

$$x_{ij} \leq \sum_{k \in succ(j),\ k \neq i} x_{jk} \ \forall (i,j) \in S_0 \cup S_2 \text{ with } j \neq t \tag{2}$$

$$x_{ij} \leq \sum_{k \in pred(i),\ k \neq j} x_{ki} \ \forall (i,j) \in S_0 \cup S_2 \text{ with } j \neq s \tag{3}$$

$$0 \leq x_{ij} \leq 1, \ \forall \ (i,j) \in S_0 \cup S_2 \tag{4}$$

Constraints (2) and (3) ensure that conditions (a) and (b) are satisfied. Note that although Constraints (3) seem to relax the binary nature of x_{ij}, the objective function forces the optimal value of x_{ij} to be equal to 0 or 1.

Actually, an arc (i_0, j_0) is irrelevant if and only if it does not belong to any acyclic $s-t$ path. In other words, if one cannot find any two node-disjoint $s-i_0$ and $j_0 - t$ paths, then (i_0, j_0) is irrelevant. In order to check this condition, we consider the network derived from $S_2 \cup S_0 \backslash \{(i_0, j_0), (j_0, i_0)\}$ by duplicating each node j into two nodes j' and j" linked with an arc $(j', j")$; replacing each arc (i, j) by an arc $(i", j')$; adding two dummy nodes s_0 and t_0 together with arcs (s_0, s), (s_0, j_0), (i_0, t_0) and (t, t_0); and setting all the arc values to 1. Now, if the minimum $s_0 - t_0$ cut value is less than 2, then there are no node-disjoint $s - i_0$ and $j_0 - t$ paths.

Upper bounds on the maximum number of selected links: Let $c_{[k]}$ and $p_{[k]}$ denote the k^{th} smallest attacking cost, and the k^{th} smallest attack survivability in S_0. Denote by K_1 the maximum number satisfying $\sum_{k=1,\ldots,K_1} c_{[k]} \leq B$ and $\prod_{k=1,\ldots,K_1}(1 - p_{[k]}) \geq P$. Now, let K_2 denote the maximum number of selected links in S_0 yielding a lower bound (e.g. LB_3 described below) on the minimum expected cost which is less than the best found upper bound. Clearly, an upper bound on the maximum number of selected links in S_0 is $K = \min(K_1, K_2)$.

Feasibility tests: Given the sets S_1 and S_2, a node of the branch-and-bound tree may be pruned if one of the following conditions is satisfied: $(C1)$ an $s - t$ cut has been obtained; $(C2)$ there is no way to disconnect s and t; $(C3)$ the attacking cost will exceed B; $(C4)$ the probability of a successful attack will fall below P; $(C5)$ the number of selected links will exceed K.

Checking (C1) and (C2): Let ψ denote the minimum $s - t$ cut value obtained on the network defined by setting values equal to $|S_1| + 1$, 1, and

$(|S_0|+1)(|S_1|+1)$ for all links belonging to S_0, S_1 and S_2, respectively. Three cases are to be considered:

(i) $\psi \leq |S_1|$: condition (C1) is satisfied since attacking a subset of links from S_1 suffices to disconnect s and t.

(ii) $|S_1|+1 \leq \psi < (|S_0|+1)(|S_1|+1)$: attacking all links of S_1 is not sufficient to disconnect s and t. On the other hand, the obtained minimal cut provides a way of disconnecting s and t by completing the links of S_1 by additional selected links from S_0. Therefore neither (C1) nor (C2) is satisfied.

(iii) $\psi \geq (|S_0|+1)(|S_1|+1)$: condition (C2) is satisfied since there is no way to disconnect s and t except by attacking at least one unattackable link from S_2.

Checking (C3)-(C5): Recall that all links of S_1 are removed from the network, and that the attacking cost of any link belonging to S_2 is equal to ∞. Now, let α, β, and γ denote the minimum $s-t$ cut values obtained after setting the value of each link (i,j) in S_0 to $c_{i,j}$, $-\ln(1-p_{i,j})$, and 1, respectively. Clearly, Condition (C3) is satisfied if $\sum_{(i,j)\in S_1} c_{i,j} + \alpha > B$, Condition (C4) is satisfied if $e^{-\beta}\prod_{(i,j)\in S_1}(1-p_{i,j}) < P$, and Condition (C5) is satisfied if $|S_1|+\gamma > K$.

Lower bounds: Given the set S_1 and a lower bound LB on $Z^*(S_0)$, a lower bound on $Z^*(E)$ is $Z(S_1)+\prod_{(i,j)\in S_1}(1-p_{i,j})LB$. A simple way to compute LB is to set the cost and the attack survivability of all links in S_0 to $c_{\min} = \min_{(i,j)\in S_0} c_{i,j}$ and $p_{\max} = \max_{(i,j)\in S_0} p_{i,j}$, respectively. Since the minimum number of links of S_0 to be selected is γ, then a valid lower bound on $Z^*(S_0)$ is $LB_1 = [1-(1-p_{\max})^{\gamma}]c_{\min}/p_{\max}$. A second lower bound on $Z^*(S_0)$ which dominates LB_1 is $LB_2 = \sum_{k=1,...,\gamma} w_k c_{[k]}$, where $w_k = \prod_{h=0,...,k-1}(1-p_{[h]})$ (with $p_{[0]}=0$). For presorted links, LB_1 and LB_2 can be updated in $O(1)$ time if one link is moved from S_0 to S_1 or S_2.

A more effective lower bound on $Z(S_0)$, denoted herafter by LB_3, can be derived by computing the subset $\overline{S}$ of the γ links of S_0 that yield the minimum expected cost (Note that $\overline{S}$ does not necessarily constitute a cut set.) Assume that the links $(i_1,j_1),(i_2,j_2),...,(i_{|S_0|},j_{|S_0|})$ of S_0 are sorted according to the nondecreasing order of $c_{i,j}/p_{i,j}$. According to that order, let $f(k,h)$ denote the minimum expected cost of the subset of the $\gamma-h+1$ last links in $\overline{S}$ if (i_k,j_k) is the h^{th} link in $\overline{S}$. Clearly, we have $f(k,\gamma) = c_{i_k,j_k}$; $f(k,h) = c_{i_k,j_k} + (1-p_{i_k,j_k}) \min_{l=k+1,...,|S_0|}\{f(l,h+1)\}$ for $h = 1,...,\gamma-1$; and

$LB_3 = \min_{k=1,\ldots,|S_0|}\{f(k,1)\}$. Using a dynamic programming algorithm together with some dominance rules, LB_3 can be computed in $O(|S_0|^2)$ time. In our implementation, $LB_k (k = 2, 3)$ is computed only if $LB_{k-1} < UB$.

Upper bounds: Each time an $s-t$ cut is obtained (e.g. in the computation of $\psi, \alpha, \beta, \gamma$), three attempts to derive a valid upper bound on the optimal solution are performed. Let S denote the subset of (non-redundant) links that are selected in order to complement those links of S_1. First, a non-redundant cut set which satisfies the budget constraint is attempted to be found. For that purpose, we consider the minimum cut obtained on the network defined by removing all links of S, setting a value equal to $c_{i,j}$ for all links belonging to S_1, and a value equal to ∞ for all the remaining links. The subset of links $S_1^{'} \subseteq S_1$ that have to be cut constitutes the set of non-redundant links with minimum attacking cost. Note that if $\sum_{(i,j)\in S_1' \cup S} c_{i,j} > B$, then no feasible cut could be derived. Otherwise, the subset $S_1^{'} \cup S$ provides a feasible cut set if $\prod_{(i,j)\in S_1' \cup S}(1 - p_{i,j}) \geq P$, and $|S_1^{'} \cup S| \leq K$. In this case, the expected cost of the obtained cut set constitutes a valid upper bound on the optimal solution.

Two additional attempts to derive non-redundant cut sets which satisfy the confidence (resp. the cardinality) constraint are performed in a similar way by setting a value equal to $-\ln(1 - p_{i,j})$ (resp. equal to 1) for all links belonging to S_1 in the network described above.

4 Preliminary computational results

The proposed branch-and-bound algorithm has been implemented and tested on a set of instances generated in the following way. The attacking costs are drawn from the discrete uniform distribution on $[20, 60]$ for s and t related links, and on $[1, 40]$ for the remaining ones. The survival probabilities upon attack are generated between 0.1 and 0.35 for s and t related links, and between 0.05 and 0.3 for the remaining links. The number of nodes n is taken equal to 10, 15, 20, 25, and 30. The degree of each node is randomly generated between 2 and d_{max} where $d_{max} \in \{4, 5, 6, 7\}$. For each (n, d_{max}) combination, 10 instances were randomly generated. All the computational experiments were carried out on a Quad 2.8 GHz Personal Computer with 4 GB RAM. The linear program has been solved using CPLEX 11.1. Table 1 displays the performance of the proposed procedure with respect to the variation of n and $d_{\max}$, where we provide: the number of instances for which optimality was not proved after reaching a 30 minutes time limit (US), the average CPU

time in seconds ($Time$), and the average number of explored nodes (NN). We observe that the preliminary implementation of the proposed algorithm exhibits promising performance on medium-sized networks. Enhanced results for larger networks are expected to be displayed during the talk. In particular, the impact of each component of the branch-and-bound algorithm will be investigated in order to derive the most efficient variant.

		US	*Time*	*NN*
n	*10*	0	28.39	87267.4
	15	4	253.12	391504.3
	20	14	975.29	982298.8
	25	25	1366.46	1308521.6
	30	32	1523.05	1193070.7
d_{max}	*4*	10	509.55	493768.3
	5	17	794.87	740335.5
	6	23	968.86	871149.5
	7	25	1043.77	1064877.0

Table 1
Performance of the Branch-and-Bound algorithm

References

[1] M.N. Azaiez and Vicki M. Bier, *Optimal resource allocation for security in reliability systems*, European Journal of Operational Research **181** (2007), pp. 773–786.

[2] Bier V.M. and M.N. Azaiez, *Game Theoretic Risk Analysis of Security Threats*, Springer: The International Series of Operations Research and Management Science, **Vol. 128** (2009), ISBN 978-0-387-87766-2.

[3] C. H. Papadimitriou and M.Yannakakis, *On the approximability of trade-offs and optimal access of web sources*, In Proceedings of the IEEE Symposium on Foundations of Computer Science (2000), pp. 86–92.