A sequential algorithm for the bilevel network interdiction problem

Srinesh Chandrakesa[a], Varghese Kurian[a]

[a]*Department of Chemical Engineering, Indian Institute of Technology Madras, Chennai, India*

Abstract

Several attempts on network interdiction have been made by different authors in the past due to its application in military operations, supply chain management, prevention of illicit drug trade and other related areas. The problem is NP-complete and various techniques have been proposed for solving this problem. In this work we propose a greedy heuristic for the sequential disruption of edges in a network constrained by edge capacity. Because of a few shortcomings in this algorithm, we further include a stochastic component in the selection of edges. The solution was originally developed for the problem statement of the 8^{th} AIMMS - MOPTA Optimization Modeling Competition and therefore, the applicability of the technique is shown on the network provided for the competition.

Introduction

In the network interdiction problem, an 'interdictor' is provided with limited resources for disrupting a network through which the adversary operates its supply. The goal of the supplier is to maximize its profit using the existing network. This problem is an example of a Stackelberg game [14] where the interdictor first chooses its disruption policy followed by the supplier choosing its optimal supply policy.

One of the earliest works in this field was reported by Wollmer [12]. Here, an algorithm was proposed for the removal of arcs from a graph such that the maximum flow between a source and a sink was minimized. Later, Ratliff [9] extended this to the case of n arcs. Ghare *et al.* [6] proposed a branch and bound algorithm for a similar problem that is encountered by military commanders. Studying this problem in the context of illegal drug trade, Wood [13] showed that even a simpler version of the problem is NP-complete and developed an integer programming model for the problem. An extension of the model was also provided for continuous interdiction of the arcs. Parallel to these, Fulkerson & Harding [4] and Golden [7] worked on maximizing the length of the shortest s-t path in a network and showed this problem to be equivalent to a minimum cost flow problem. Further, the problem of a single evader travelling between two nodes and an interdictor setting up inspection points for detection of this was addressed by Washburn and Wood [11].

For a continuous interdiction problem, Lim and Smith [8] showed that the optimum would have a maximum of one partially disrupted edge as all other edges chosen for disruption would be completely disrupted. As the problem was still difficult to solve, a heuristic was proposed for solving this. Though it provided quick solution is certain cases, the technique was not always reliable. Meanwhile, Cormican *et al.* introduced uncertainties into arc capacities and developed a stochastic integer program for solving it [2]. A *connectivity interdiction* problem was also modeled and solved by Zenklusen [15]. In a more recent work, Bertsimas *et al.* have proposed a new modeling framework where the interdictor chooses one of the multiple pure strategies available based on probabilities assigned to them [1]. The authors have referred to this as the *randomized network interdiction problem*. In the mean time, the model has also seen applications on domains other than military such as multicommodity flow networks [8], communication networks and electric power grids [10].

The exact problem description is given in the next section followed by the solution technique. An alternate interpretation for the theorem given by Lim and Smith [8] is provided and a novel solution technique using a sequential greedy algorithm is discussed. The results obtained while applying this technique on a network is given in the continuing section. Few variations of the parameters used in the algorithm are tried and their performances are reported.

Problem description

In this section, a resource transportation network with a supplier agent that generates profit by transporting resources from production centers to demand nodes is described. The network used for this paper is adopted from 8th AIMMS-MOPTA competition held at Lehigh university in 2016. The dataset consisting of individual demands for each nodes, edge capacities and cost per unit for edge traversal can be dowloaded from the competition website http://coral.ie.lehigh.edu/∼mopta /competition.

All the nodes are demand points except nodes 15,16 and 20 which are supplier nodes as shown in figure (1). The supplier agent tries to maximize the transport of resources from the supply points to the demand nodes though the edges. For every unit of goods transported through any edge, the supplier has to pay a transportation cost (which is different for different edges). In addition, the edges are confined by a capacity. Unlimited goods can be generated from the supply points, therefore the traffic in the network is constrained by the maximum capacity on these edges. The goal of the supplier agent is to maximize its profit earned by transporting the goods from the supply nodes to the demand nodes. The interdictor on the other hand tries to minimize the supplier's profit. For this purpose, the interdictor is allowed to make reductions in the edge capacities of the network, but with a limited

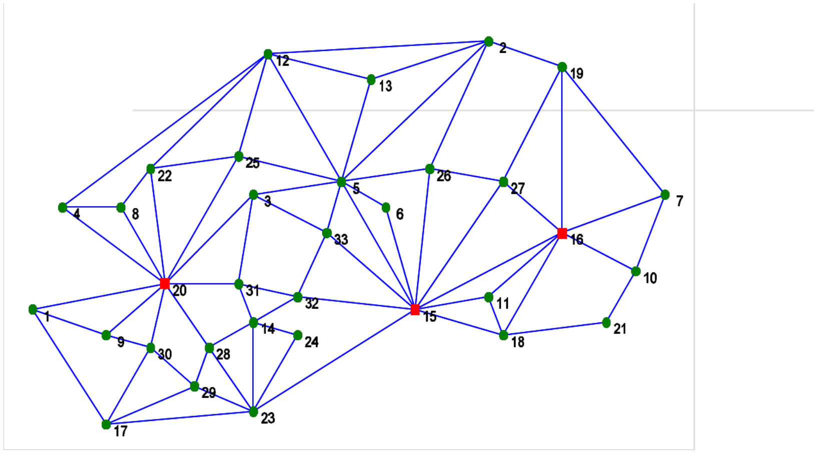

Figure 1: Transportation network

budget constraints. This means that the total amount of edge capacity reductions the interdictor can make to the network is bounded.

Solution methodology

The interdictor's problem is to determine the capacity reduction that has to be caused on various edges in the network and this is a combonatorially difficult problem to solve. As discussed earlier, even a simpler version of this problem is shown to be NP complete [13]. Therefore, a greedy heuristic that sequentially chooses the edges for interdiction is used for solving the problem. After each step, the heuristic solves the suppliers problem before choosing the next edge for interdiction. The implementation of this technique is fairly simple due to the relative ease of solving the suppliers problem.

The supplier agent in the network tries to maximize its profit by directing resources to the demand centers. This profit maximization problem can be formulated as a linear program ($P1$). Here the objective function (profit) is defined as the difference between the revenue earned from the nodes and the transportation cost.

The constraints are similar to that of the popular max-flow problem. It includes the flow conservation at nodes, edge capacity constraints and the limits on the maximum demand that can be handled by the sink nodes. It has to be noted that the edge capacity constraints allow flow though the edges in either direction subject to the limit on maximum flow allowable.

$$
\begin{aligned}
(P1) \qquad & \qquad \max_{x,y} \; p^T x - b^T |y| \\
& s.t \\
\text{Flow balance constraint :} \quad & Ay - x = 0 \qquad & A : \text{Incidence matrix} \\
\text{Maximum demand constraint :} \quad & x_i \leq d_i \qquad & i \in \mathbb{N} \\
\text{Non negative demand constraint :} \quad & x_i \geq 0 \qquad & i \in \mathbb{N} \backslash \{15, 16, 20\} \\
\text{Edge capacity constraints :} \quad & -w_j \leq y_j \leq w_j \qquad & j \in \mathbb{E}
\end{aligned}
$$

Though the problem necessitates that the flows have to be integral, such a constraint is not explicitly made. The reason being that even in the absence of the constraint, the solution to $P1$ would still remain integral as long as the maximum demand at the nodes and the edge capacities are integral.

An intuitive proof for the claim given above can be made by adding a super sink to the graph. All demand nodes of the original graph has to be connected to this super sink by edges with capacity equal to the demand of the respective node. The supplier's problem can now be reformulated for the modified graph with all the demand constraints dropped, but with the capacity constraints for the new edges added. The constraints for this problem are exactly same as that of the max-flow problem which is already proven to have an integral solution. Therefore, the original problem would also have an integral solution.

Here it is assumed that the supplier agent has access to complete information of the available network and shows a rational behavior by trying to maximize its profit according to the solution of $P1$. The interdictor now has to reduce or disable the capacities of individual edges in the graph. Accordingly, the supplier will reroute (or reduce) the flows. The problem can be viewed as a Stackelberg game with the interdictor (leader) reducing and/or disabling the routes according to the total budget and the suppliers (follower) trying to maximize its profit under new available graph.

The interdictor's problem can now be formulated as a min-max problem ($P2$). Here the objective is to minimize the maximum profit that can be achieved by the supplier agent.

$$
\begin{array}{llr}
(P2) & \underset{c}{\text{Min}} \quad \underset{x,y}{\text{Max}}\ p^T x - b^T |y| & \\
& s.t\ \ Ay - x = 0 & \\
& x_i \le d_i & i \in \mathbb{N} \\
& x_i \ge 0 & i \in \mathbb{N} \backslash \{15, 16, 20\} \\
& -(w_j - c_j) \le y_j \le w_j - c_j & j \in \mathbb{E} \\
& w_j - c_j \ge 0 & \\
& c_j \ge 0 & \\
\text{Budget constraints :} & \sum_j c_j \le \text{budget} &
\end{array}
$$

Among constraints, in comparison to $P1$, the only addition in $P2$ is the limit on maximum disruption that can be caused by the interdictor. The variable c_j is used to denote the extend by which capacity of edge j is reduced by the interdictor.

The huge solution space available for simultaneous disruptions limits the possibility of an exhaustive search to find the best solution to ($P2$), even for a small network. Prior to discussing a heuristic to solve this problem, we establish a special property of the system that significantly decreases the search space and allows us to find faster solutions. This has already been proved by Lim and Smith [8] based on the results of Dantzig [3]. Here we provide two propositions that further elucidates the claim.

Proposition 1.

The marginal damages caused by disruption is a non-increasing function of edge capacity $w_j, j \in E$.

Proof. From Theorem IV-3 [5], the objective function of P1 is a concave and a piecewise affine function of capacity, $w_j, j \in E$. Additionally, as the profit never reduces with increase in capacity of an edge, the function is non-decreasing. Hence the marginal reduction in profit is a non-increasing, piece wise constant function of edge capacity.

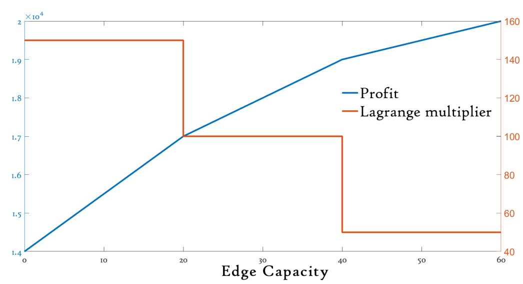

Figure 2: Illustration for proposition 1.

Proposition 2.

*At most one edge is partially cut in the solution to P*2*, i.e., given a disrupting budget l, in the optimal solution to P*2*, amongst all edges with $c > 0$, at most one edge has $c < w_j$.*

Proof. Let us assume to the contrary, i.e., that when $P2$ is solved, there are two least edges i and j that are partially disabled, i.e., $0 < c_j < w_j$ and $0 < c_i < w_i$. Let λ_i and λ_j be the Lagrange multiplies corresponding to the respective capacity constraints. Wlog, we may assume $\lambda_i > \lambda_j > 0$. Hence, it is possible to reduce the capacity of edge i by ε, $\varepsilon > 0$ and increase the capacity of edge j by ε with no change in the budget. Since $\lambda_i > \lambda_j$, this results in overall reduction in profit. Since λ_i and λ_j are non-increasing functions of capacity (Proposition 1), this can be continued further till edge i is completely disrupted, i.e., $c_i = w_i$ or edge j is not disrupted at all, i.e., $c_j = 0$. This can be continued until there exists at most one edge with $c > 0$ amongst the edges which are disrupted.

Proposition 2 greatly reduces the search space for the given problem. Now the candidate solutions are the ones that have at most one edge partially disrupted (and all other edges chosen for disruption to be completely disabled). However, the total number of possible network disruptions are still large. In the following section, we propose a greedy algorithm that sequentially disables the edges in the network. A meta heuristic is introduced that make use of the dual variables of the edge capacity constraints. Initially, a deterministic edge selection strategy is discussed. This is followed by a variant of this algorithm that has a stochastic component included.

Deterministic algorithm

Let z_1 and z_2 be the dual variables corresponding to the edge capacity constraints in $P1$. These are indicators of the importance of the corresponding constraints to the LP. Since the flow is considered non directional, $z = z_1 + z_2$ denotes the limiters to the LP. The algorithm proceeds by the following :

Problem $P1$ is executed and the corresponding dual variables (z) are obtained. The edge with the highest dual variable is chosen for disruption and this is completely disrupted subject to availability of budget. With the updated capacities, $P1$ is executed and the dual variables are obtained. The process continues until the budget constraints are met. If capacity of a certain edge cannot be fully disabled, maximum reduction is caused to the edge subject to budget constraint. The pseudo code is given below as Algorithm 1.

Algorithm 1:

$w_{cur} = 0$ $//w_{cur}$ = sum of all current capacity reductions

while $w_{cur} <$ Budget **do**

  Solve P1

  Select edge $f \mid$ z(f) is the highest $\forall f \in E$

  Update $w(f)$ as max(0,w_{cur} - Budget) and w_{cur} appropriately

end while

The disruption strategy discussed above need not be optimal as the dual variables indicates the best edge only for an ε-cut. For a complete edge disruption, the one chosen using the dual variables need not be the best. Even if the edge chosen turns out to be the best available, this need not remain valid as the disruption progresses. To overcome this shortcoming, we introduce a stochastic component in the sequential selection of edges for disruption. This is discussed in the next section.

Randomized algorithm

Due to the shortcomings of the deterministic algorithm discussed earlier, we now introduce a stochastic component for the selection of edges for disruption. Instead of selecting edges strictly on the basis of the value of the dual variables, an edge is selected with a probability proportional to the value of its dual variable. A non zero probability (1- p) for picking an edge with its dual variable zero is also included. Choosing edges for disruption continues until the capacity constraints are met. This complete disruption process is repeated multiple times and the best solution amongst them is chosen. The pseudo code for the algorithm is given as Algorithm 2.

Algorithm 2:

$w_{cur} = 0$ $//w_{cur}$ = sum of all current capacity reductions

while $w_{cur} <$ Budget **do**

Solve P1

Group the set of edges into $L_1 \in z(f) \neq 0$ and $L_2 \in z(f) = 0$

Sort the edges f by the decreasing order of $z(f)$ $f \in E$

Compute $q(f) := z(f)/\Sigma_f z(f)$

Select randomly an edge f from set L_1 with a probability $\propto qp$ or an edge f from set L_2 with a probability $\propto p$

Update $w(f)$ as max(0,w_{cur} - Budget) and w_{cur} appropriately

end while

Results

The performance of the deterministic and randomized algorithms are tested for the given network for different disruption capacities. The problem was modeled using the AIIMS platform and CPLEX solver was used. A comparison between the solution obtained by both the algorithms are shown in figure (3).

The randomized algorithm takes about 34.4 seconds for 1000 iterations in an Intel (R) Core (TM) i5-4570 CPU @ 3.20 GHz system. Figure (4) shows the

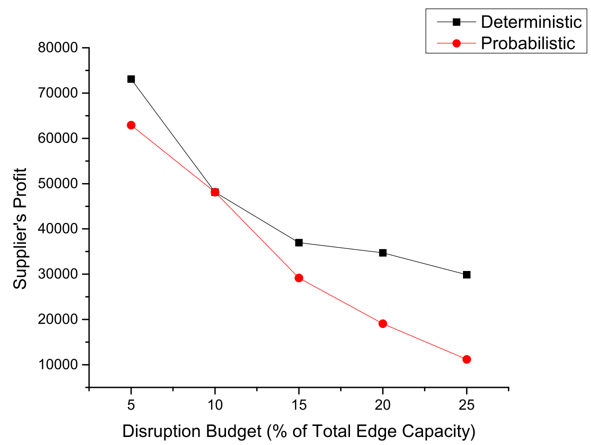

Figure 3: Comparison of deterministic and randomized algorithm for the test network

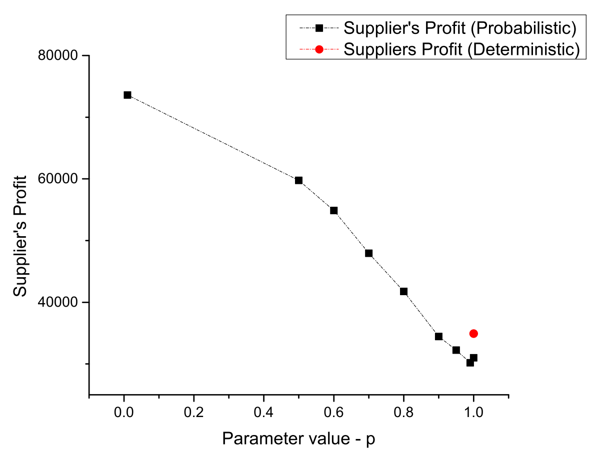

Figure 4: Variation with parameter p

variation of the mean profit generated by the supplier for different values of parameter p. It should be noted here that the mean supplier profit for the randomized algorithm is smaller than the deterministic one for values of p above 0.9. The proposed method is computationally less intensive owing to its sequential nature with solution quality comparable with those obtained from other techniques such as Bender's decomposition. The figures (5) and (6) shows the best cut for 15 % and 20 % budget. It has to be noted that the solution was obtained multiple times even in a single run of 1000 iterations.

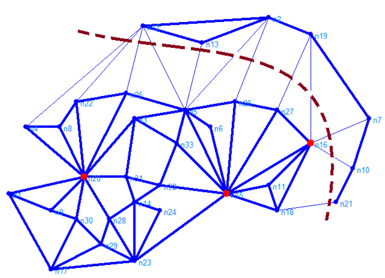

Figure 5: Best cut for 15 % budget

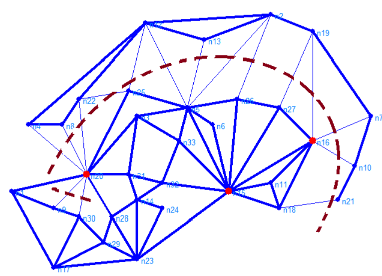

Figure 6: Best cut for 20 % budget

Conclusion

In this work, a greedy sequential algorithm is proposed for a bilevel network interdiction problem. Initially, a heuristic based on the dual variables corresponding to the edge capacity constraints of the supplier's problem is introduced which is later extended to include a stochastic selection of edges. The technique was tested on a standard network and promising results were obtained without much computational effort.

References

[1] Bertsimas, D., Nasrabadi, E., Orlin, J.B.: On the power of randomization in network interdiction. Operations Research Letters 44(1), 114–120 (2016)

[2] Cormican, K.J., Morton, D.P., Wood, R.K.: Stochastic Network Interdiction. Operations Research 46(2), 184–197 (1998)

[3] Dantzig, B.Y.G.B.: Upper Bounds , Secondary Constraints , and Block Triangularity in Linear Programming. Econometrica 23(2), 174–183 (1955)

[4] Fulkerson, D.R., Harding, G.C.: Maximizing the minimum source-sink path subject to a budget constraint. Mathematical Programming 13(1), 116–118 (1977)

[5] Gal, T.: Postoptimal Analyses, Parametric Programming, and Related Topics: degeneracy, multicriteria decision making, redundancy, vol. 13. http://www.degruyter.com/view/product/143242, Berlin,Boston (1994)

[6] Ghare, P.M., Montgomery, D.C., Turner, W.C.: Optimal interdiction policy for a flow network. Naval Research Logistics Quarterly 18(1), 37–45 (1971), `http://doi.wiley.com/10.1002/nav.3800180103`

[7] Golden, B.: A problem in network interdiction. Naval Research Logistics Quarterly 25(4), 711–713 (1978), `http://doi.wiley.com/10.1002/nav.3800250412`

[8] Lim, C., Smith, J.C.: Algorithms for discrete and continuous multicommodity flow network interdiction problems. IIE Transactions 39(1), 15–26 (2007)

[9] Ratliff, H.D., Sicilia, G.T., Lubore, S.H.: Finding the n most vital links in

flow networks. Management Science 21(5), 531–539 (1975), http://dx.doi.org/10.1287/mnsc.21.5.531

[10] Salmeron, J., Wood, K., Baldick, R.: Worst-case interdiction analysis of large-scale electric power grids. IEEE Transactions on power systems 24(1), 96–104 (2009)

[11] Washburn, A., Wood, K.: Two-Person Zero-Sum Games for Network Interdiction. Operations Research 43(November 2015), 243–251 (1995)

[12] Wollmer, R.: Removing Arcs from a Network. Operations Research 12(6), 934–940 (1964), http://dx.doi.org/10.1287/opre.12.6.934

[13] Wood, R.K.: Deterministic network interdiction. Mathematical and Computer Modelling 17(2), 1–18 (1993)

[14] Wood, R.K.: Bilevel network interdiction models: Formulations and solutions. Wiley Encyclopedia of Operations Research and Management Science pp. 1–11 (2011)

[15] Zenklusen, R.: Connectivity interdiction. Operations Research Letters 42(6), 450–454 (2014)