Annals of Operations Research
https://doi.org/10.1007/s10479-024-06022-4

ORIGINAL RESEARCH

The all-pairs vitality-maximization (VIMAX) problem

Alice Paul[1] · **Susan E. Martonosi**[2]

Received: 24 February 2023 / Accepted: 22 April 2024

Abstract
Traditional network interdiction problems focus on removing vertices or edges from a network so as to disconnect or lengthen paths in the network; network diversion problems seek to remove vertices or edges to reroute flow through a designated critical vertex or edge. We introduce the all-pairs vitality maximization problem (VIMAX), in which vertex deletion attempts to maximize the amount of flow passing through a critical vertex, measured as the all-pairs vitality of the vertex. The assumption in this problem is that in a network for which the structure is known but the physical locations of vertices may not be known (e.g., a social network), locating a person or asset of interest might require the ability to detect a sufficient amount of flow (e.g., communications or financial transactions) passing through the corresponding vertex in the network. We formulate VIMAX as a mixed integer program, and show that it is NP-Hard. We compare the performance of the MIP and a simulated annealing heuristic on both real and simulated data sets and highlight the potential increase in vitality of key vertices that can be attained by subset removal. We also present graph theoretic results that can be used to narrow the set of vertices to consider for removal.

Keywords Vitality · Network flow · Centrality · Network disruption

1 Introduction

Network disruption has important applications to infrastructure design (Brown et al., 2006; McMasters & Mustin, 1970; Church et al., 2004), energy transmission (Callaway et al., 2000; Holmgren, 2006), robust network design (Crucitti et al., 2004b; Dodds et al., 2003; Estrada, 2006), biological systems (Rasti & Vogiatzis, 2022), illicit trade networks (Anzoom et al., 2021), and counterterrorism (Basu, 2005; Sageman, 2004). Much of this work focuses on two primary problem types: network flow interdiction and network diversion. In *network interdiction*, an attacker is interdicting vertices or edges to maximize the minimum cost of

A. Paul and S. E. Martonosi have contributed equally to this work.

✉ Susan E. Martonosi
martonosi@g.hmc.edu

Alice Paul
alice_paul@brown.edu

1 Biostatistics Department, Brown University, 121, S Main St, 7th Floor, Box G-S121-7 , Providence, RI 02912, USA

2 Mathematics Department, Harvey Mudd College, 301 Platt Blvd., Claremont, CA 91711, USA

Published online: 10 May 2024

routing flow through a network (Alderson et al. 2015). This is commonly examined in the literature in the form of two special cases: selecting an interdiction set that minimizes the maximum flow between a source and sink (e.g., Altner et al. (2010), Balcioglu and Wood (2003), Bertsimas et al. (2016), Lei et al. (2018), Lim and Smith (2007), Royset and Wood (2007), Wood (1993), Enayaty-Ahangar et al. (2019)), or such that the shortest path between a source and sink is maximized (e.g., Israeli and Wood (2002), Pay et al. (2019), Zhang et al. (2018)). In *network diversion*, a minimum cost, minimal cutset of edges is identified such that when removed, any source-sink path in the network is forced to travel through a particular set of critical edges (e.g., Cintron-Arias et al. (2001), Cullenbine et al. (2013), Curet (2001)).

Of interest in this paper is the concept of vertex (equivalently, edge) *vitality*, which measures the reduction in the maximum flow between the source and sink when that vertex (or edge) is removed from the graph (Ausiello et al., 2019; Koschützki et al., 2005). A vertex having high vitality is needed to achieve a high volume of flow from source to sink, and as such, this vertex will have a high volume of flow passing through it when the maximum flow is achieved. We define the *all-pairs vitality* of a vertex v to be the summed reduction in the maximum flow between all pairs of nodes (themselves excluding vertex v), when vertex v is removed from the graph.

We present the following combinatorial optimization problem, the all-pairs vitality maximization problem (VIMAX): Given a connected, directed, general capacity graph $G = (V, E)$ with vertex set V, edge set E, and a key vertex of interest, k, identify a subset of vertices S, whose removal from the graph G maximizes the all-pairs vitality of k. This problem was first introduced in the second author's unpublished manuscript for the specific context of undirected, unit-capacity graphs, for which the maximum flow between a pair of vertices represents the number of edge-disjoint paths between that pair (Martonosi et al. 2011).

VIMAX can be considered a network disruption problem that is distinct from the three forms outlined above. Covert organizations, such as terrorist groups or drug cartels, tend to communicate along longer paths that are difficult to trace, suggesting a trade-off between efficiency and secrecy that could render path-length-based attacks ineffective (Anzoom et al., 2021; Freeman et al., 1991; Morselli et al., 2007). Moreover, we leverage the possibility that critical vertices in certain types of networks can become vulnerable if they are forced to become more active. (As an example, Osama bin Laden and, subsequently, Ayman al-Zawahiri were known to be leaders of the al-Qaeda terrorist network, yet they remained in hiding for many years before U.S. intelligence could pinpoint their geographic locations.) If we assume the volume of communication, money, or illicit substances passing through a vertex is a proxy for that corresponding member's visibility to intelligence officers, and communication between pairs of members in the organization is proportional to path capacity, then VIMAX can identify members of the organization whose removal will maximize communication through an important but clandestine leader. Unlike in network diversion problems, we do not require all flow in the remaining graph to be routed through this vertex (indeed in a network diversion problem, the volume of flow passing through the critical vertex might be quite small after vertex or edge removal); instead we seek to *maximize* the total flow routed through this vertex.

In this paper, we examine VIMAX from both computational and theoretical perspectives. In Sect. 2, we frame this work in the context of the existing literature. In Sect. 3, we define VIMAX, present it as a mixed integer linear program, and demonstrate that it is NP-Hard. Section 4 presents a simulated annealing heuristic for solving VIMAX. The computational performance of these two methods is compared in Sect. 5. Section 6 presents mathematical

properties of VIMAX that can be leveraged to streamline computations. Section 7 provides future extensions of this work and concludes.

2 Literature review

We first contrast the network interdiction and diversion problems commonly seen in the literature with the VIMAX problem we will present in this paper. We then discuss the relationship between vitality and other graph centrality metrics. Finally, we present research on optimization approaches that could be useful to the problem of vitality maximization.

2.1 Network interdiction and diversion

Network interdiction models address the logistical problem of removing edges or vertices from a graph to inhibit the flow of resources through a network. This has applications to military operations and combating drug or human trafficking (Konrad et al., 2017; Tezcan and Maass, 2023; Zhang et al., 2018). Analysis of complex network interdiction typically focuses on disconnecting the network, increasing the lengths of shortest paths, cutting overall flow capacity, or reducing the desirability of paths in the network (Albert et al., 2000; Flaxman et al., 2007; Cavallaro et al., 2004; Gallos et al., 2004, 2005; Gallos et al., 2006; Gierszewski et al., 2006; Grubesic et al., 2008; Holme et al., 2002; Holzmann & Smith, 2021; Memon et al., 2008; Paul et al., 2005; Pay et al., 2019; Sun et al., 2007; Tezcan & Maass, 2023; Wu et al., 2007; Zhang et al., 2018). The most well-known model involves maximum flow network interdiction and its variants (Altner et al., 2010; Bertsimas et al., 2016; Cormican et al., 1998; Lei et al., 2018; McMasters & Mustin, 1970; Phillips, 1993; Ratliff et al., 1975; Royset & Wood, 2007; Wood, 1993). Of note, Wood (1993) introduces the "dualize-and-combine" method that is commonly used in network interdiction literature, as well as in this paper. Smith and Song thoroughly survey the network interdiction literature, and demonstrate that the assumptions widely held across the papers they survey make interdiction problems a special case of Stackelberg games (Smith & Song, 2020).

A related problem to network interdiction is the *network diversion problem* in which an attacker seeks to interdict, at minimum cost, a set of edges (equivalently, vertices) such that all source-sink flow must be routed through at least one member of a pre-specified set of "diversion" edges or vertices. This problem was first posed by Curet (2001). Applications include military operations, in which it might be beneficial to force a foe to divert its resources through a target edge that is heavily armed; and information networks, in which communications are routed through a single edge that can more easily be monitored (Lee et al., 2019).

Cullenbine et al. also study the network diversion problem (Cullenbine et al., 2013). They present an NP-completeness proof for directed graphs, a polynomial-time solution algorithm for $s-t$ planar graphs, a mixed integer linear programming formulation that improves upon that given in Curet (2001), and valid inequalities to strengthen the formulation.

Lee et al. examine an extension of the network diversion problem known as the *multiple flows network diversion problem* in which there are many source-sink pairs being considered simultaneously (Lee et al., 2019). They define a set S of possible source nodes and T of possible sink nodes. They are interdicting a *minimum cost* set of edges such that all remaining flow in the network passes through the diversion edge. They formulate the problem as a mixed integer linear program, and compare its performance to standard combinatorial Benders

decomposition and a branch-and-cut combinatorial Benders decomposition. Without loss of generality, vertex interdiction can be formulated as arc interdiction in which each vertex v in the original graph is represented by two vertices v_i and v_o in a modified graph having a single arc between them, (v_i, v_o). Each arc (u, v) in the original graph is then transformed to a corresponding arc (u_o, v_i) in the modified graph. Interdicting this arc in the modified graph is equivalent to interdicting the vertex in the original graph. For undirected graphs, the graph is first transformed into a directed one before doing the transformation.

There are several aspects of Lee et al. (2019) worth noting as they connect to our work. First, after the interdiction set is removed from the graph, there is no guarantee that the total flow passing through the diversion edge is particularly large. In the vitality maximization problem that we present here, we are identifying an interdiction set of vertices such that the flow through the target vertex is maximized, thus ensuring that the target being surveilled has ample flow. Although our formulation does not associate a cost with each vertex that is interdicted, it is disadvantageous for the removal subset to be very large, as that would inherently cause the flow through the target vertex to drop. Second, we adopt their testing scheme of examining the performance of the algorithms we develop on grid networks (planar), as well as random $G_{n,m}$ graphs (Knuth 2014), and a drug trafficking network (Natarajan 2000).

A question conversely related to network interdiction and diversion is that of network resilience and detection of attacks. Sharkey et al. survey literature on four types of resilience: robustness, rebound, extensibility, and adaptability, with a primary focus on research addressing network robustness and the ability of a network to rebound following an attack (Sharkey et al., 2021). Dahan et al. study how to strategically locate sensors on a network to detect network attacks Dahan et al. (2022).

2.2 Vitality and other graph centrality measures

Vitality is one of several types of graph centrality metrics. Centrality metrics quantify the importance of a given vertex in a network. The book of Wasserman and Faust provides a detailed examination of social network analysis stemming from the field of sociology and includes discussion of many commonly known centrality metrics, including degree, betweenness, and closeness (Wasserman & Faust, 1994). The survey of Rasti and Vogiatzis presents centrality metrics commonly used in computational biology (Rasti & Vogiatzis, 2019).

The degree of a vertex is the number of neighbors it has. The betweenness of a vertex is the number of shortest paths between all pairs of vertices on which the vertex lies. Closeness measures the average shortest path length between the vertex and all other vertices in the graph. Vogiatzis et al. present mixed integer programming formulations for identifying groups of vertices having the largest degree, betweenness, or closeness centrality in a graph (Vogiatzis et al., 2015).

Stephenson and Zelen first proposed *information centrality* and applied it to a network of men infected with AIDS in the 1980s (Stephenson & Zelen, 1989). They are among the first to develop a centrality metric that does not require an assumption that information must flow along shortest paths. They use the theory of statistical estimation to define the information of a signal along the path to be the reciprocal of the variance in the signal. Assuming the noise induced along successive edges of a path is independent, the variance along each path is additive, and the total variance in the signal grows with the path length. They then use this assumption to evaluate the total information sent between any pair of vertices (s, t). From

here, they define the centrality of a vertex i to be the harmonic average of the sum of the inverses of the information sent from from vertex i to every other vertex. They point out that "information ... may be intentionally channeled through many intermediaries in order to 'hide' or 'shield' information in a way not captured by geodesic paths." This appears to be the case in terrorist and other covert networks as well (Carpenter et al., 2002).

Centrality metrics can be used to guide network disruption approaches. Cavallaro et al. show that targeting high betweenness vertices efficiently reduces the size of the largest connected component in a graph based on a Sicilian mafia network (Cavallaro et al. 2004). Grassi et al. find that betweenness and its variants can be used to identify leaders in criminal networks Grassi et al. (2019).

There also exist centrality measures related to network flows, as surveyed in Koschützki et al. (2005). In particular, for any real-valued function on a graph, Koschützki et al. define the *vitality* of a vertex (or edge) to be the difference in that function with or without the vertex (or edge). When the function represents the maximum flow between a pair of vertices, the *vitality* of a vertex k in a graph (equivalently, an edge u) with respect to an $s - t$ pair of vertices is defined to be the reduction in the maximum flow between s and t when vertex k (equivalently, edge u) is removed from the graph. Moreover, when one examines the same reduction in maximum flow in the network over all possible $s - t$ pairs with respect to a given vertex, we have what Freeman et al. define as *network flow centrality* (Freeman et al., 1991), or what we refer to as *all-pairs vitality* in this paper.

The *most-vital edge* or component is the one whose removal decreases the maximum flow through the network by the greatest amount. Identifying the most-vital edge in a network is a long-studied problem dating back to the work of Corley and Chang (1974), Wollmer (1963), and Ratliff et al. (1975). More recent examination includes the work of Alderson et al. (2013), who formulate a mathematical program to maximize resilience, using a defender-attacker-defender model. They additionally cite several applications for the most-vital edge problem including electric power systems, supply chain networks, telecommunication systems, and transportation. Ausiello et al. provide a method for calculating the vitality of all edges (with respect to a given s and t) with only $2(n - 1)$ maximum flow computations, rather than the m computations expected if one were to calculate the vitality of each edge individually (Ausiello et al., 2019). None of the found literature pertaining to vitality focuses on the problem presented here: that of identifying a set of removal vertices to maximize the vitality of a key vertex (VIMAX).

3 Optimization framework

We will show that VIMAX can be formulated as an integer linear program. We start by presenting terminology that will be used in the paper.

3.1 Definitions

We consider a connected, directed graph $G = (V, E)$ with vertex set V, edge set E, and a key vertex of interest, k. Each edge (i, j) has a capacity u_{ij} reflecting the maximum amount of flow that can be pushed along that edge. The graph has a *key vertex*, k, which could represent, for example, an important but elusive participant in an organization. The *vitality maximization problem (VIMAX)* seeks to identify a subset of vertices whose removal from the graph G maximizes the all-pairs vitality of k. Thus, the objective is to identify a set of

vertices to remove from the graph to make the key vertex k as "active" as possible by forcing flow to pass through that vertex.

For any source-sink s-t pair, let $z_{st}(G)$ be the value of the maximum s-t flow in graph G. We call $Z_k(G)$ the *flow capacity of graph G with respect to vertex k*, which is the all-pairs maximum flow in G that does not originate or end at k. Thus,

$$Z_k(G) = \sum_{\substack{s,t \in V \setminus \{k\} \\ s \neq t}} z_{st}(G). \tag{1}$$

The *all-pairs vitality of k*, $\mathcal{L}_k(G)$, equals the flow capacity of the graph with respect to k *minus* the flow capacity with respect to k of the subgraph $G \setminus \{k\}$ obtained when vertex k is deleted:

$$\mathcal{L}_k(G) = Z_k(G) - Z_k\left(G \setminus \{k\}\right). \tag{2}$$

It is worth noting that maximizing the all-pairs vitality of k does not imply an assumption that all $s-t$ pairs will communicate simultaneously. Rather, maximizing the sum over all pairs is equivalent to maximizing the average vitality of k when an $s-t$ pair is chosen uniformly at random from all pairs. Thus, we are maximizing the expected communication through the key vertex for a randomly chosen pair of vertices. The framework that follows can be easily extended to weighted pairs according to their likelihood of engaging in communication; we outline this in Sect. 3.3.

To measure how the removal of a subset of vertices impacts the vitality of the key vertex, we define the *vitality effect* of subset S on key vertex k to be the change in the key vertex k's vitality caused by removing subset S: $\mathcal{L}_k(G \setminus S) - \mathcal{L}_k(G)$. If the vitality effect of S on k is positive, then removing subset S from the graph has diverted more flow through k, a desired effect.

The goal of this research is to identify the subset of vertices S that maximizes the vitality effect, which is equivalent to maximizing the value of $\mathcal{L}_k(G \setminus S)$. We formally define the all-pairs vitality maximization problem (VIMAX) as

$$max_{S \subseteq V}\ \mathcal{L}_k(G \setminus S). \tag{3}$$

From expressions (1) and (2), we see that there is no guarantee that the vitality effect on k of removing any subset S need ever be positive. When subset S is removed from the graph, the overall flow capacity $Z_k(G \setminus S)$ generally decreases, and never increases, because S's contribution to the flow is removed. In order for subset S's removal to have a positive vitality effect on key vertex k, the remaining flow must be rerouted through k in sufficiently large quantities to overcome the overall decrease in flow through the network. However, as we will show in Sect. 5.3, identification of an optimal or near-optimal removal subset often dramatically increases the vitality of the key vertex.

3.2 Mixed integer linear programming formulation

To formulate VIMAX as an optimization problem, we first formulate a linear program to solve for the vitality of k in any graph G. Then we expand that formulation into a mixed integer programming formulation that seeks the optimal subset S of vertices to remove from the graph to maximize the vitality of k in the resulting graph.

3.2.1 Vitality max-flow subproblems.

Following the approach of Israeli and Wood (2002), we take the dual of problem $Z_k(G\backslash\{k\})$ to convert it into a minimum cut problem having the same optimal objective function value, and embed it in the formulation of $\mathcal{L}_k(G)$. Since the dual problem is a minimization problem, the objective function will correctly correspond to the vitality. Letting $V' = V\backslash\{k\}$, and letting E' be the set of edges that remain after removing vertex k and its incident edges, we obtain the following linear program for finding $\mathcal{L}_k(G)$:

$$
\begin{aligned}
&\text{Maximize} \sum_{\substack{s,t\in V'\\ s\neq t}} v_{s,t} - \sum_{\substack{s,t\in V'\\ s\neq t}} \sum_{(i,j)\in E'} u_{i,j}\alpha_{i,j,s,t}\\
&\text{subject to}\\
&\qquad \sum_{j:(i,j)\in E} x_{i,j,s,t} - \sum_{j':(j',i)\in E} x_{j',i,s,t} = \begin{cases} v_{s,t} & \text{if } i = s\\ -v_{s,t} & \text{if } i = t\\ 0 & \text{otherwise}\end{cases}\\
&\qquad\qquad \forall i \in V, \forall s,t \in V'\\
&\qquad x_{i,j,s,t} \le u_{i,j}, \quad \forall (i,j) \in E, \forall s,t \in V'\\
&\qquad \beta_{i,s,t} - \beta_{j,s,t} + \alpha_{i,j,s,t} \ge 0, \quad \forall (i,j) \in E', \forall s,t \in V'\\
&\qquad -\beta_{s,s,t} + \beta_{t,s,t} \ge 1, \quad \forall s,t \in V'\\
&\qquad v_{s,t} \ge 0, \quad \forall s,t \in V'\\
&\qquad x_{i,j,s,t} \ge 0, \quad \forall (i,j) \in E, \forall s,t \in V'\\
&\qquad \alpha_{i,j,s,t} \ge 0, \quad \forall (i,j) \in E', \forall s,t \in V'\\
&\qquad \beta_{i,s,t} \text{ unrestricted}, \quad \forall i,s,t \in V'.
\end{aligned}
\tag{4}
$$

Variables $x_{i,j,s,t}$ and $v_{s,t}$ are the primal variables from the maximum flow formulation of problem $Z_k(G)$. $x_{i,j,s,t}$ represent the optimal $s-t$ flow pushed along edge (i,j), and $v_{s,t}$ represent the optimal $s-t$ flow values. Variables $\alpha_{i,j,s,t}$ and $\beta_{i,s,t}$ are the dual variables from the minimum cut formulation of problem $Z_k(G\setminus\{k\})$. We can interpret $\beta_{i,s,t}$ as vertex potentials: For every edge (i,j), if $\beta_{i,s,t} < \beta_{j,s,t}$, meaning vertex i has lower potential than vertex j when computing the minimum $s-t$ cut, then edge (i,j) must cross the cut. In such a case, dual variable $\alpha_{i,j,s,t} = 1$, and edge capacity $u_{i,j}$ is counted in the objective function.

3.2.2 VIMAX: choosing an optimal removal subset.

Now that we have expressed the vitality of k in G as a linear program, we can return to VIMAX, which finds a subset S of vertices whose removal maximizes the vitality of k. Given a set S, the linear program in Eq. 4 applied to graph $G\backslash S$ solves for $\mathcal{L}_k(G\backslash S)$. We must modify the LP above to choose a subset S that maximizes the objective function $\mathcal{L}_k(G\setminus S)$.

We can formalize this by creating binary variables q_i for each vertex i such that $q_i = 1$ if vertex i remains in the graph, and $q_i = 0$ if vertex i is removed from the graph (that is, i is included in subset S). We also define variables $w_{i,j}$ for each edge that indicate whether or not edge (i,j) remains in the graph following the removal of S and/or k. We define linking constraints so that whenever both vertices i and j remain in the graph (that is, $q_i = q_j = 1$), then $w_{i,j}$ must equal 1, and whenever either vertex i or j is selected for deletion (that is, $q_i = 0$ or $q_j = 0$ or both) then $w_{i,j}$ must equal 0. (Due to this relationship between $w_{i,j}$ and

the binary q_i, the $w_{i,j}$ are effectively constrained to be binary variables without explicitly declaring them as such.)

To Eq. 4, we make the following adjustments to the original primal and dual constraints. We constrain the primal flow variables $x_{i,j,s,t} \leq u_{i,j} w_{i,j}$, reflecting whether or not edge (i, j) remains in the graph. We also modify the dual potential constraints so that $\alpha_{i,j,s,t} = 0$ whenever vertices i and j are at the same potential (as before) or edge (i, j) no longer exists in the graph.

Introducing the variables q_i and $w_{i,j}$ and the modifications on our vitality constraints, we can now write the full mixed-integer linear program. Given a graph $G = (V, E)$, a key vertex k, and a maximum size, m, of the removal set, the following mixed-integer linear program solves VIMAX.

$$\text{Maximize} \sum_{\substack{s,t \in V' \\ s \neq t}} v_{s,t} - \sum_{\substack{s,t \in V' \\ s \neq t}} \sum_{(i,j) \in E'} u_{i,j} \alpha_{i,j,s,t}$$

subject to

$$\begin{aligned}
&\sum_{i \in V} q_i \geq n - m \\
&q_k = 1 \\
&w_{i,j} \leq q_i, \quad \forall (i, j) \in E \\
&w_{i,j} \leq q_j, \quad \forall (i, j) \in E \\
&w_{i,j} \geq q_i + q_j - 1, \quad \forall (i, j) \in E
\end{aligned}$$

$$\sum_{j:(i,j) \in E} x_{i,j,s,t} - \sum_{j':(j',i) \in E} x_{j',i,s,t} = \begin{cases} v_{s,t} & \text{if } i = s \\ -v_{s,t} & \text{if } i = t \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in V, \forall s, t \in V' \tag{5}$$

$$\begin{aligned}
&x_{i,j,s,t} \leq u_{i,j} w_{i,j}, \quad \forall (i, j) \in E, \forall s, t \in V' \\
&\beta_{i,s,t} - \beta_{j,s,t} + \alpha_{i,j,s,t} \geq -(1 - w_{i,j}), \quad \forall (i, j) \in E', \forall s, t \in V' \\
&-\beta_{s,s,t} + \beta_{t,s,t} \geq 1, \quad \forall s, t \in V'
\end{aligned}$$

$$\begin{aligned}
&q_i \text{ binary}, \quad \forall i \in V \\
&w_{i,j} \geq 0, \quad \forall (i, j) \in E \\
&v_{s,t} \geq 0, \quad \forall s, t \in V' \\
&x_{i,j,s,t} \geq 0, \quad \forall (i, j) \in E, \forall s, t \in V' \\
&\alpha_{i,j,s,t} \geq 0, \quad \forall (i, j) \in E', \forall s, t \in V' \\
&\beta_{i,s,t} \text{ unrestricted}, \quad \forall i, s, t \in V'
\end{aligned}$$

Extending the approach of Ovadia (2010) to general capacity, directed graphs, we can show that VIMAX is NP-Hard. In the case that $m = 1$ and we can remove at most one vertex, we can do brute-force and solve the above MIP setting $q_i = 0$ and all other $q_j = 1$ for all $i \in V'$.

Theorem 1 *The all-pairs vitality maximization problem is NP-Hard.*

Proof The proof of this can be found in Appendix A. □

3.3 Extension to pairwise weights

Maximizing the sum over all pairs is equivalent to maximizing the average vitality of k when an $s - t$ pair is chosen uniformly at random from all pairs. Thus, we are maximizing the expected communication through the key vertex for a randomly chosen pair of vertices. However, not all pairs may be equally likely to communicate. The objective function above can be easily extended to weight each pair according to their likelihood of engaging in communication. Let $\mathrm{P}_{s,t}$ be a weight corresponding to the frequency that pair $s - t$ communicates. For example, traffic matrix estimation can be used to estimate the pairwise demands for an internet network Zhang et al. (2003); Medina et al. (2002). The objective function in Eq. 5 can be modified to include these weights to maximize the expected communication through the key vertex.

$$\text{Maximize} \sum_{\substack{s,t \in V' \\ s \neq t}} \mathrm{P}_{s,t} v_{s,t} - \sum_{\substack{s,t \in V' \\ s \neq t}} \sum_{(i,j) \in E'} \mathrm{P}_{s,t} u_{i,j} \alpha_{i,j,s,t}$$

4 Simulated annealing heuristic

As an alternative to solving VIMAX exactly with a MIP, we develop a simulated annealing heuristic. Each iteration of simulated annealing begins with a candidate removal subset. In the first iteration, this is the empty set, and in subsequent iterations the initial solution is the best solution found at the conclusion of the previous iteration. The objective function value of each solution is computed as the vitality of the key vertex when this subset is removed from the graph. Each call to the algorithm consists of an *annealing phase* and a *local search phase*.

During the annealing phase, neighboring solutions of the current solution are obtained by toggling a single vertex's, or a pair of vertices', inclusion or exclusion from the candidate removal subset, subject to the constraint that $|S| \leq m$. If the neighboring solution improves the objective function value, it is automatically accepted for consideration. If the neighboring solution has a worse objective function value, it will be accepted to replace the current solution with an *acceptance probability* governed by a temperature function, T. When the temperature is high (in early iterations), there is a high probability of accepting a neighboring solution even if its objective function value is worse than that of the incumbent solution. This permits wide *exploration* of the solution space. In later iterations, the temperature function cools, reducing the likelihood that lower objective function value solutions will be considered. This permits *exploitation* of promising regions of the solution space.

Given temperature T, the probability of accepting a solution having objective function value e_0 when the best objective function value found so far is $e_{max} > e_0$ is given by $P = e^{-(e_{max} - e_0)/T}$. The initial temperature, T, is chosen so that the acceptance probability of a solution having at least 90% of the initial objective function value is at least 95%. In subsequent iterations, T is cooled by a multiplicative factor of 0.95.

After a set number of annealing iterations, a single iteration of local search is conducted on the best solution found so far by toggling each vertex sequentially to determine if its inclusion or exclusion improves the objective function value. The best solution found is returned.

We use a Gomory-Hu tree implementation of the all-pairs maximum flow problem to rapidly calculate the vitality of the key vertex on each modified graph encountered by the

heuristic (Gomory & Hu, 1961; Gusfield, 1990). For mathematical reasons that are discussed in Sect. 6, we can exclude leaves from consideration in any removal subset. These two enhancements permit the simulated annealing heuristic to run very fast on even large instances, as we discuss in Sect. 5.3.

5 Computational analysis

We now present performance comparisons on a variety of datasets of the MIP formulation and the simulated annealing heuristic. Following the approach of Lee et al. (2019), we generate grid networks, which are planar. We also test the performance of the methods on random networks (Knuth, 2014) and on a real drug trafficking network (Natarajan, 2000). We first describe these data sets and the computational platform used, and then we present the results. Code and data files are available at our Github repository: https://github.com/alicepaul/network_interdiction.

5.1 Data

5.1.1 Grid networks

We generate grid networks in a similar fashion as Lee et al. (2019). We generate square $M \times M$ grids with M varying from five to eight. Such graphs have an edge density of $\frac{4}{M(M+1)}$, which ranges from 13.3% for $M = 5$ to 5.6% for $M = 8$. On each grid, we generate edge capacities independently and uniformly at random from the integers from 1 to M. For each case, we likewise consider two scenarios, testing a maximum removal subset size of $m = 1$ or $m = M$ (that is, $\sqrt{|V|}$). For each grid size and removal subset size combination, we generate three trial graphs. For each trial, a key vertex is selected uniformly at random over the vertices.

5.1.2 Random $G_{n,m}$ Networks.

Random $G_{n,m}$ graphs are parametrized by a number of vertices, $n = |V|$, and a number of edges, $m = |E|$ (Knuth 2014). Each graph is sampled by finding a random graph from the set of all connected graphs with n nodes and m edges. We test our methods on graphs having the same number of vertices and same number of edges as the grid networks above: $|V| = \{25, 36, 49, 64\}$ vertices, with $|E| = \{40, 60, 84, 112\}$, respectively. On each graph, we generate edge capacities independently and uniformly at random from the integers from 1 to $\sqrt{|V|}$. For each case, we likewise consider two scenarios, testing a maximum removal subset size of $m = 1$ or $m = \sqrt{|V|}$. For each graph size and removal subset size combination, we generate three trial graphs. For each trial, the key vertex is selected to have the highest betweenness centrality.

5.1.3 Drug trafficking network

Lastly, we test our models on a real-world covert cocaine trafficking group, prosecuted in New York City in 1996 (Natarajan, 2000). This network consists of 28 people between whom 151 phone conversations were intercepted over wiretap over a period of two months. An edge exists between persons i and j if at least one conversation between them appears in the data set. There are 40 edges in this graph, corresponding to an edge density of 10.6%. We can

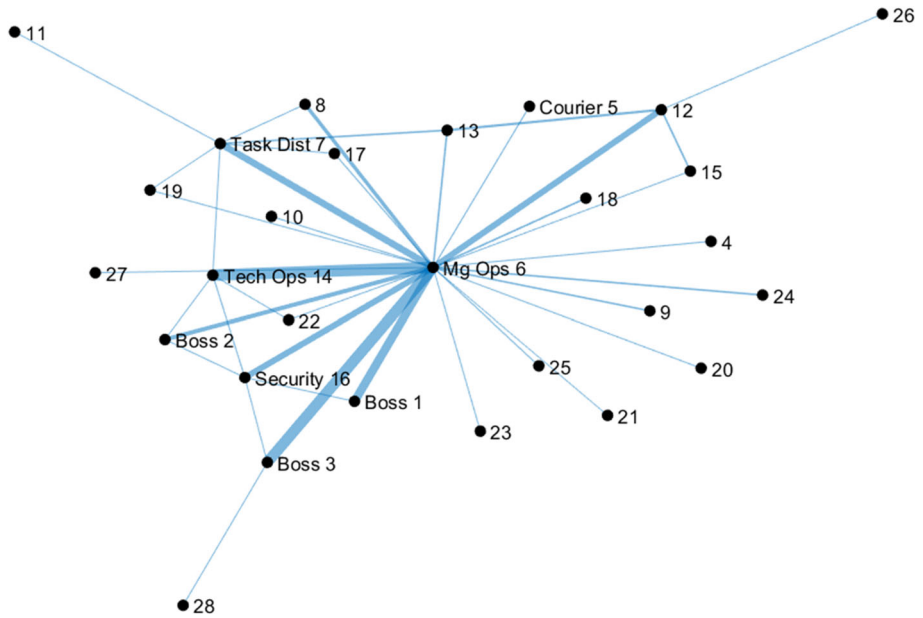

Fig. 1 Cocaine trafficking network of Natarajan et al. Line width is proportional to number of wiretapped calls made between pairs of operatives (Natarajan, 2000)

Table 1 Roles of notable vertices in the cocaine trafficking network of Natarajan et al. (Natarajan, 2000)

Vertex	Role
1, 2, 3	Colombian Bosses
5	Courier
6	Managing Operations
7	Task Distribution
14	Technical Operations
16	Security

consider a unit capacity version of the network, as well as a general capacity version in which the capacity on edge $i - j$ is equal to the number of conversations between them appearing in the data. The weighted network is shown in Fig. 1, where line width is proportional to the number of wiretapped calls occurring between two operatives. According to Natarajan et al., some individuals in the network are known to have the roles described in Table 1. We test a maximum removal subset size of $m = 1$ or $m = 5 \approx \sqrt{|V|}$. Because the Colombian bosses (vertices 1, 2, and 3) are high-level leaders important to the functioning of the organization, we treat these vertices as the key vertices on which we attempt to maximize vitality.

5.2 Computational framework

The performance of the MIP and the simulated annealing heuristic was tested on a computer with a 3 GHz 6-Core Intel Core i5 processor and 16 GB of memory. The Single-VIMAX and VIMAX MIP instances were run in python 3.9.6 calling the CPLEX solver through

the CPLEX python API, and were each limited to two hours of computation time. The simulated annealing heuristic was also coded in python and limited to 10,000 iterations on each trial instance. Initial results were collected using the Extreme Science and Engineering Discovery Environment (XSEDE) supercomputers (Towns et al., 2014) and up to five hours of computation time but did not show substantially different results. In addition to the general VIMAX MIP, a single vertex removal MIP (Single VIMAX) was also tested. Single vertex removal simulated annealing results are not reported, as they are effectively equivalent to brute force search.

5.3 Results

Table 2 presents the results of all completed trials. The first five columns explain the graph type, number of vertices ($|V|$), number of edges ($|E|$) and for the general VIMAX problem allowing multiple removals, the maximum allowed size, m, of the removal subset. Column six gives the initial vitality of the key vertex in the original graph with no vertices removed. Columns seven through ten provide results on the performance of the single vertex removal MIP (Single VIMAX); columns eleven through fifteen provide results from the multi-removal MIP (VIMAX); and columns sixteen through nineteen provide results from the multi-removal simulated annealing heuristic. (There is no need to use simulated annealing for Single VIMAX because it can be solved by sequentially testing the removal of each vertex.) For the three methods, the best vitality found within the time or iteration limit, the MIP gap if available, the percentage increase of the best vitality found by the method over the original vitality of the key vertex in the full graph, and the running time in seconds are given. For the multi-removal methods, the size of the best found removal subset ($|S|$) is also given. MIP instances that terminated due to time limit have Time reported as $'-'$. Figures 2, 3, and 4 plot, for each graph type, vitality averaged over the three trials by removal method (original vitality, single-removal MIP, multi-removal MIP, and multi-removal simulated annealing).

First we note that Table 2 and Figs. 2–4 provide a proof-of-concept demonstrating that it is possible to increase (sometimes dramatically) the vitality of the key vertex through subset removal. Removing a single vertex increased the vitality by 42%-200% in all grid network instances for which the MIP solved to optimality within the time limit, and by up to 82% in the random graph instances; single vertex removal was not able to increase the vitality of the key vertex in the drug network. When allowing multiple removals, simulated annealing was able to identify removal subsets that increased the vitality on the key vertex by as much as 1,373%.

Unsurprisingly, the full VIMAX MIP allowing multiple removals is substantially harder to solve than the single removal MIP. On grid and random networks, the MIP failed to terminate within the two-hour time limit on all instances with at least $n = 36$ nodes. On the $n = 36$ random and the 7×7 and 8×8 grid network instances, the single removal MIP also did not terminate within the time limit, but an improving solution was returned in more cases. The large MIP gaps on the multi-removal MIP indicate a failure to find improving integer solutions.

For multiple vertex removal, the simulated annealing heuristic yielded excellent solutions in a fraction of the time required by even the single removal MIP. On the large instances for which the multiple removal MIP reached the time limit, the simulated annealing heuristic found substantially better solutions than the MIP incumbents. For those instances in which the multiple removal MIP solved to optimality, the solutions found by simulated annealing are often optimal and always near-optimal.

Table 2 Computational results of solving VIMAX via mixed integer program (single VIMAX and multi-removal VIMAX) and simulated annealing (multi-removal VIMAX). For the grid and random networks, each row represents a randomly generated instance with randomly selected key vertex. MIP trials that reached the two-hour time limit show 'Time' reported as '-'

						Single-Removal MIP				Multi-Removal MIP					Multi-Removal Simulated Annealing			
Graph Type		$\|V\|$	$\|E\|$	m	Orig Vitality	Best Found Vitality	MIP Gap (%)	% Incr	Time (s)	Best Found Vitality	MIP Gap (%)	% Incr	Time (s)	$\|S\|$	Best Found Vitality	% Incr	Time (s)	$\|S\|$
Drug Net.	unit cap.	28	40	5	3	3	0.00	0.00	15.78	8	0.00	166.67	173.38	3	8	166.67	182.10	3
		28	40	5	5	5	0.00	0.00	11.32	8	0.00	60.00	249.92	3	8	60.00	185.09	3
		28	40	5	29	29	0.00	0.00	9.41	31	0.00	6.90	94.11	3	31	6.90	210.50	3
	gen. cap.	28	40	5	5	5	0.00	0.00	13.91	5	0.00	0.00	378.17	0	5	0.00	196.87	3
		28	40	5	7	7	0.00	0.00	11.23	7	0.00	0.00	329.90	0	7	0.00	225.23	1
		28	40	5	31	31	0.00	0.00	9.58	31	0.00	0.00	177.60	0	31	0.00	264.67	0
Random	$n = 25$	25	40	5	0	0	0.00	0.00	32.70	0	0.00	0.00	1819.28	0	0	0.00	244.41	0
		25	40	5	64	90	0.00	40.63	25.91	135	0.00	110.94	1020.51	5	115	79.69	103.65	4
		25	40	5	56	73	0.00	30.36	22.04	149	0.00	166.07	144.94	3	149	166.07	168.73	3
	$n = 36$	36	60	6	34	34	0.00	0.00	526.01	34	4548.07	0.00	–	0	34	0.00	543.61	0
		36	60	6	230	368	0.00	60.00	533.77	693	169.68	201.30	–	6	859	273.48	129.36	6
		36	60	6	167	304	0.00	82.04	435.19	938	74.89	461.68	–	5	980	486.83	282.99	5
	$n = 49$	49	84	7	287	335	0.00	16.72	6007.47	287	1900.56	0.00	–	0	651	126.83	229.95	7
		49	84	7	387	581	0.00	50.13	4559.45	387	1632.44	0.00	–	0	957	147.29	452.74	6
		49	84	7	969	1,254	0.00	29.41	3519.72	969	622.02	0.00	–	0	2574	165.63	346.76	6
	$n = 64$	64	112	7	151	151	2377.86	0.00	–	151	9144.91	0.00	–	0	2225	1373.51	427.85	8
		64	112	7	581	581	563.49	0.00	–	581	1971.25	0.00	–	0	2780	378.49	897.63	8
		64	112	7	602	602	492.67	0.00	–	602	1887.84	0.00	–	0	1619	168.94	368.37	8

Table 2 continued

Graph Type		$\|V\|$	$\|E\|$	m	Orig Vitality	Single-Removal MIP Best Found Vitality	MIP Gap (%)	% Incr	Time (s)	Multi-Removal MIP Best Found Vitality	MIP Gap (%)	% Incr	Time (s)	$\|S\|$	Multi-Removal Simulated Annealing Best Found Vitality	% Incr	Time (s)	$\|S\|$
Grid	5 × 5	25	40	5	271	387	0.00	42.80	75.21	559	0.00	106.27	576.68	3	559	106.27	186.38	3
		25	40	5	126	379	0.00	200.79	54.97	472	0.00	274.60	528.42	2	472	274.60	195.55	2
		25	40	5	207	377	0.00	82.13	58.27	432	0.00	108.70	503.27	2	432	108.70	202.67	2
	6 × 6	36	60	6	336	603	0.00	79.46	1376.30	683	288.43	103.27	–	2	526	56.55	486.04	3
		36	60	6	79	180	0.00	127.85	1219.06	293	1016.24	270.89	–	6	422	434.18	513.14	4
		36	60	6	222	587	0.00	164.41	1071.95	1,066	195.83	380.18	–	6	1178	430.63	493.66	4
	7 × 7	49	84	7	845	1303	142.50	54.20	–	845	817.34	0.00	–	0	2435	188.17	740.19	6
		49	84	7	651	651	439.40	0.00	–	651	1145.64	0.00	–	0	2363	262.98	743.13	5
		49	84	7	653	1446	205.77	121.44	–	653	1348.54	0.00	–	0	3352	413.32	1224.05	3
	8 × 8	64	112	7	1410	2212	239.08	56.88	–	1,410	1258.79	0.00	–	0	6001	325.60	1592.45	7
		64	112	7	542	542	1008.47	0.00	–	542	2922.76	0.00	–	0	1827	237.08	442.94	8
		64	112	7	380	649	886.21	70.79	–	380	4729.71	0.00	–	0	5497	1346.58	2426.68	6

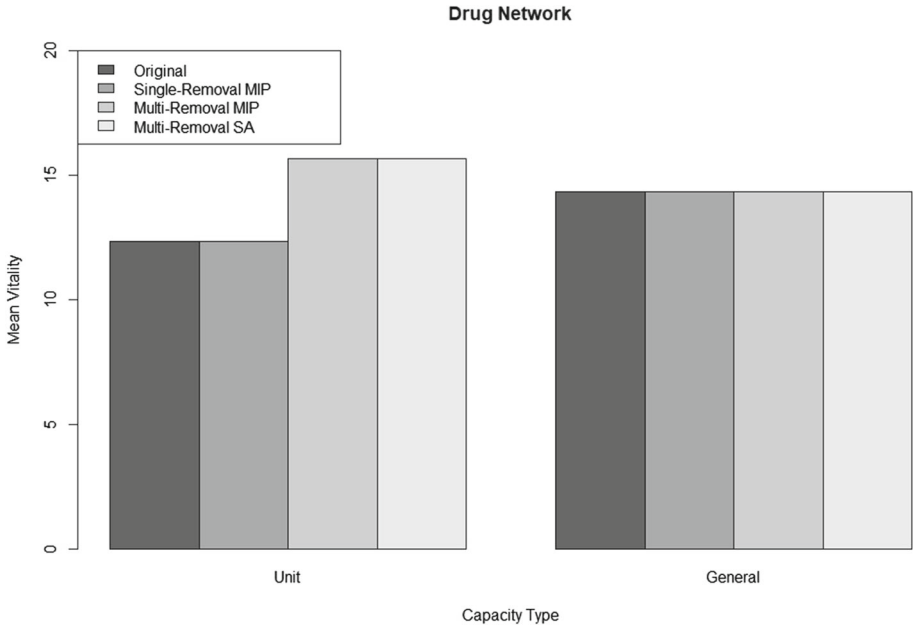

Fig. 2 Mean vitality across three trials for the unit capacity and general capacity instances of the drug network, by removal type (original graph, single-removal MIP, multi-removal MIP, and multi-removal simulated annealing)

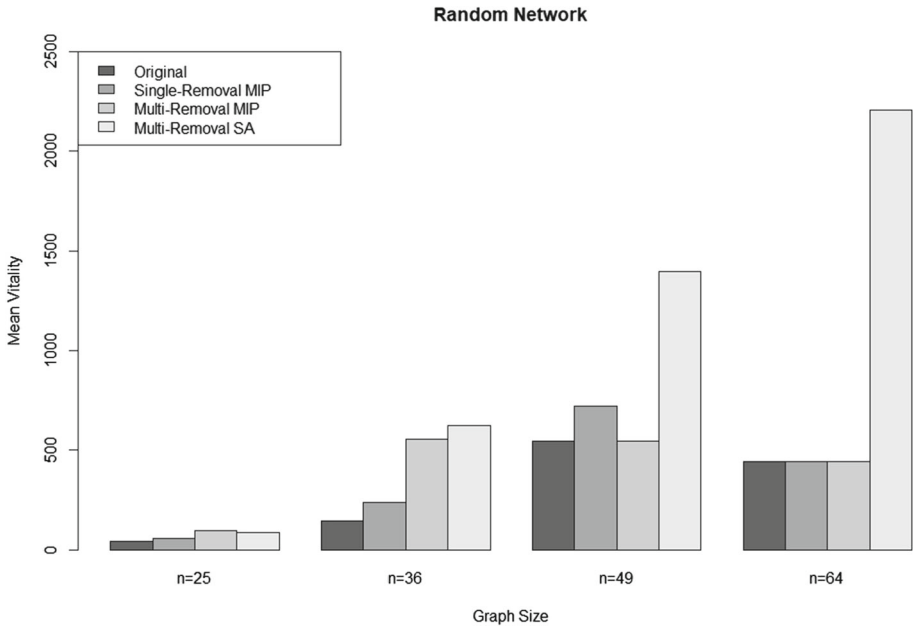

Fig. 3 Mean vitality across three trials for each size of the random network, by removal type (original graph, single-removal MIP, multi-removal MIP, and multi-removal simulated annealing)

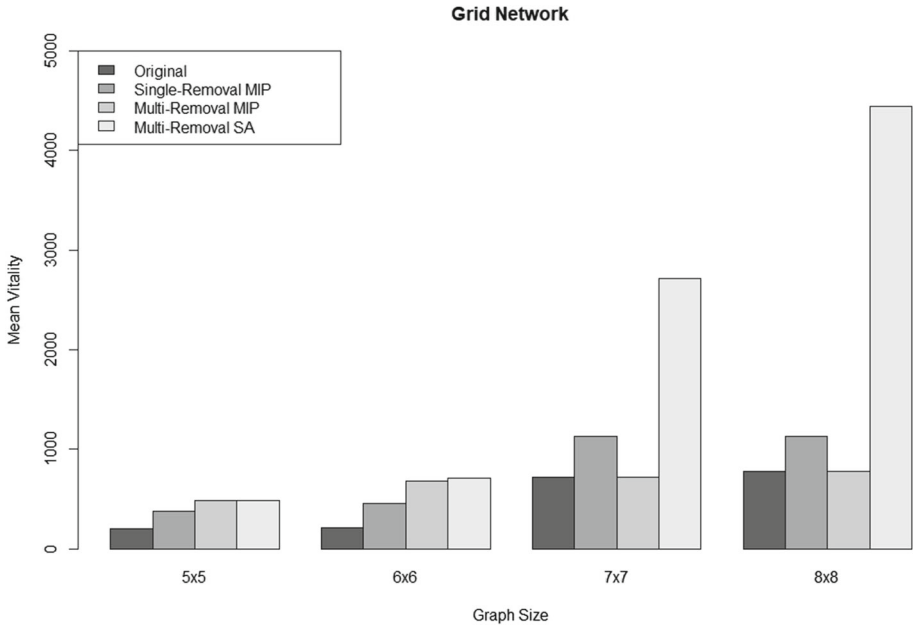

Fig. 4 Mean vitality across three trials for each size of the grid network, by removal type (original graph, single-removal MIP, multi-removal MIP, and multi-removal simulated annealing)

The effectiveness of vertex removal to maximize vitality appears to depend on the network structure and choice of key vertices. While the drug network has approximately the same number of vertices and edges as the 25-node instances of the random and grid networks, the key vertices (corresponding to vertices Boss 1, Boss 2, and Boss 3 in Fig. 1) chosen in these trials are less amenable to vitality maximization. The drug network has a large number of leaves, whereas the grid networks do not. As we will see in Sect. 6, vertices, such as leaves, that do not have at least two vertex-disjoint paths to the key vertex will never appear in an optimal removal subset.

Lastly, in these trials, we chose to restrict the removal subset size to at most m vertices. The reason to restrict the removal subset size is to reduce the solution space, and thus the complexity, of the problem. This decision is justifiable because we know removing too many vertices will cause overall flow in the network to drop such that the vitality on the key vertex cannot increase. Thus, an important question is what should be an appropriate value of m to effectively reduce the solution space without compromising the quality of solutions found? We do not have a definitive answer to this question. However, we see that in many of the trials, the best removal subset identified by any method has a size strictly less than $m \approx \sqrt{|V|}$, suggesting that this choice of m is reasonable for the sizes and types of graphs considered here.

6 Leveraging structural properties of vitality

Thus far, we have established that subset removal can dramatically increase the vitality of a key vertex. However, solving this problem exactly as a MIP is computationally intractable for

even modestly sized graphs. Fortunately, simulated annealing is an appealing alternative that yields very good solutions in dramatically less time than the MIP. In this section, we explore mathematical properties that characterize vertices that can be ignored by subset removal optimization approaches. We demonstrate how these properties can be leveraged to simplify the graph on which VIMAX is run.

6.1 Identifying vitality-reducing vertices

To reduce the complexity of the optimization formulation, we turn to identifying conditions that cause a vertex to have a vitality-*reducing* effect on the key vertex. This allows us to ignore such vertices in any candidate removal subset and reduce the solution space of the VIMAX problem.

Our first observation is that the presence of a cycle is necessary for the removal of a vertex to increase the vitality of a key vertex. The vitality of a leaf is always equal to 0, so the removal of any subset that results in k becoming a leaf also cannot increase the vitality of k. As a corollary, if k has neighbor set $N(k)$ and more than $|N(k)| - 2$ of k's neighbors are removed, the vitality effect on k will be nonpositive.

We can generalize this further. When there are not at least two vertex-disjoint paths from i to k, any removal subset including i will have a vitality effect on k no greater than the same subset excluding i, as stated by the following theorem.[1]:

Theorem 2 *Let G be a graph with key vertex k, and let i be a vertex such that there do not exist at least two vertex-disjoint paths starting at i and ending at k. Let S be any vertex subset containing i, and let $T = S \setminus \{i\}$. Then, $\mathcal{L}_k(G \backslash S) \leq \mathcal{L}_k(G \backslash T)$. Therefore, T will have at least as large a vitality effect on k as S.*

Proof The proof of this can be found in Appendix B. □

Put simply, the existence of only one vertex-disjoint path between i and k means that i and k do not lie on a cycle together. Therefore when i is removed, any $s - t$ paths that previously passed through i cannot be rerouted through any alternate path passing through k.

Note that identifying vertices that do not have at least two vertex-disjoint paths to k is computationally straightforward. We can solve an all $u - k$ pairs maximum flow problem on a related graph $\hat{G}$ in which every vertex u is replaced with a pair of vertices connected by a unit capacity edge: (u, u'). For every directed edge $i - j$ in the original graph, we include directed edge (i', j) in the modified graph. Through the use of a Gomory-Hu tree, we can solve this in $O(|V|^3\sqrt{|E|})$ time (Gomory & Hu, 1961; Gusfield, 1990). Any vertex u corresponding to vertex u' in $\hat{G}$ that has a maximum $u' - k$ flow of one in $\hat{G}$ does not have at least two vertex-disjoint paths to k in the original graph and can be ignored by any removal subset. We call the set of such vertices, $\mathcal{Q}$. Every vertex in $\mathcal{Q}$ should be maintained in the graph and not be considered for removal.

These properties show that when seeking a vitality-maximizing subset for removal, we can ignore all subsets that include:

[1] A more general cut theorem holds for the specific case of an undirected graph in which all edges in the graph have unit capacity (Martonosi et al., 2011) In such a graph, the value of the maximum $s - t$ flow equals the number of edge disjoint paths between s and t in the graph. In this case, the relationship between the size of the cut between the key vertex k and a candidate for removal, i, and the connectivity between vertices along the boundaries of that cut conveys information about the vitality effect on k of removing i. The reader is also referred to Paul (2012) for an overview of how this theorem might be implemented in practice for unit capacity, undirected graphs.

Table 3 Improvement in key VIMAX instance size parameters by identifying vitality-reducing vertices and using graph-simplification. $|\mathcal{Q}|$ is the number of vertices that do not have at least two vertex-disjoint paths to k; vertices in $\mathcal{Q}$ can be ignored by VIMAX (see Sect. 6.1). $|\hat{V}|$ and $|\hat{E}|$ are the numbers of vertices and edges, respectively, in the reduced graph after applying the graph simplification method of Sect. 6.2. The last two columns report the percentage decrease in time and percentage increase in best objective function value of the graph simplification method compared to the Multi-Removal MIP results reported in Table 2. Entries denoted by '-' indicate instances in which the MIP did not terminate within two hours

Graph Type		$\lvert V\rvert$	$\lvert E\rvert$	$\lvert\mathcal{Q}\rvert$	$\lvert\hat{V}\rvert$	$\lvert\hat{E}\rvert$	% Decr Time	% Inc Obj
Drug Network	unit cap.	28	40	14	18	30	93.19	0.00
		28	40	14	18	30	94.68	0.00
		28	40	13	18	30	91.83	0.00
	gen. cap.	28	40	14	20	32	89.40	0.00
		28	40	14	20	32	92.91	0.00
		28	40	13	20	32	90.98	0.00
Random	$n = 25$	25	40	23	12	11	99.98	0.00
		25	40	1	25	40	62.92	0.00
		25	40	6	24	38	23.98	0.00
	$n = 36$	36	60	33	5	4	99.99	0.00
		36	60	4	35	59	–	13.56
		36	60	8	35	59	–	0.00
	$n = 49$	49	84	6	48	83	–	0.00
		49	84	7	47	82	–	0.00
		49	84	7	48	83	–	0.00
	$n = 64$	64	112	8	64	112	–	0.00
		64	112	10	63	111	–	0.00
		64	112	10	63	111	–	0.00

- vertices in $\mathcal{Q}$ (i.e. they do not share a cycle with k);
- more than $|N(k)| - 2$ of k's neighbors.

After performing preprocessing on the graph to identify $N(k)$ and $\mathcal{Q}$, we can add the following constraints to the MIP formulation:

$$\begin{aligned} q_i &= 1, \forall i \in \mathcal{Q} \\ \sum_{i \in N(k)} q_i &\geq 2 \end{aligned} \tag{6}$$

Although the above constraints provide a tighter formulation for VIMAX, the anticipated benefits of these constraints are likely to be modest. Table 3 shows $|\mathcal{Q}|$ (the number of vertices that do not have at least two vertex-disjoint paths to k) for each graph used for testing in Sect. 5.

Unsurprisingly given their structure, all the vertices in the grid networks have at least two vertex-disjoint paths to k; thus none of these vertices can be eliminated from consideration and are omitted from Table 3. By contrast, the sparse drug trafficking network has nearly half of its vertices that do not have at least two vertex-disjoint paths to the key vertex; this is a significant reduction in the number of candidate vertices for removal, but VIMAX was

readily tractable on this already-small network. Thus, this criterion alone is unlikely to render previously intractable MIP instances tractable.

6.2 Simplifying the graph

Because VIMAX grows rapidly in the number of vertices, we can improve the computational tractability of VIMAX by simplifying our original graph into a vitality-preserving graph having fewer vertices. We rely heavily on Theorem 2 to do this.

Suppose that a vertex v disconnects the graph into two components T_1 and T_2 such that $k \in T_1$. Then, by Theorem 2, an optimal solution will not contain any vertex in T_2. Further, the maximum flows between pairs of vertices within T_2 do not contribute to the vitality effect on k. Therefore, all that is needed to preserve the vitality effect on k in the simplified graph is to preserve information about the maximum flow between all pairs of vertices s, t such that $s \in T_1$ and $t \in T_2$.

For all vertices $t \in T_2$ we create a single edge between t and v with capacity equal to the maximum flow between t and v. This replaces all previous edges between vertices in T_2. This affects the value of the all-pairs maximum flow problem but does not affect the vitality effect on k for any subset $S \subset T_1$. Further, if any subset of vertices $T' \subseteq T_2$ all have the same new capacity value, we combine T' into a single vertex with weight $|T'|$. When calculating the maximum flow between any pair of vertices s and t in the graph, we multiply the flow by the product of the weights of the vertices to account for this simplification.

Using the process described in the previous section, we can identify the subset of vertices $\mathcal{Q} \subseteq V \setminus \{k\}$ that do not have at least two vertex-disjoint paths to k. Given a vertex $i \in \mathcal{Q}$, we find a path from i to k and find the first vertex v along i's path to k such that v has at least two vertex-disjoint paths to k. Removing the vertex v disconnects the graph. Therefore, we follow the simplification process above and mark all vertices in the corresponding T_2, including i, as processed. We then repeatedly identify any unprocessed vertex in $\mathcal{Q}$ to further simplify the graph. After all vertices in $\mathcal{Q}$ have been processed, all these vertices will be weighted leaves in the new simplified graph where the weight depends on how many vertices have been combined. All other vertices will retain a weight of one.

Figure 5 shows an example of this simplification process in which there are two components that have been simplified. Note that vertices 4, 6, and 7 have been combined together into a vertex with weight three. Further, vertices 5 and 8 have been combined together into a vertex with weight two.

As argued above, the maximum flow between all pairs of vertices that were in the same simplified component never contribute to the vitality effect on k. Therefore, we ignore these pairs in the optimization problem by removing the appropriate variables and constraints. We therefore just need to check that we have preserved the maximum flow between all pairs of vertices that were not in the same component. This is true by nature of the weights which are multiplied. For example, in Fig. 5, we multiply by weight 4 for the maximum flow between vertex 4 and vertex 1, accounting for all the paths between vertices 4, 6, and 7 and vertex 1. Thus, our optimization problem still finds an optimal subset to remove on the simplified graph that is optimal in the original graph. The number of pairs of vertices decreases from 45 to 19 since the number of vertices excluding k decreases from 10 to 7 and we can ignore the flow between vertices 9 and 10 and between vertices 4 and 8 in the simplified graph.

Table 3 shows the number of vertices ($|\hat{V}|$) and edges ($|\hat{E}|$) in each test graph after applying the graph simplification algorithm. The only graph types experiencing an appreciable reduction in size after simplification are the drug trafficking network and the smaller random

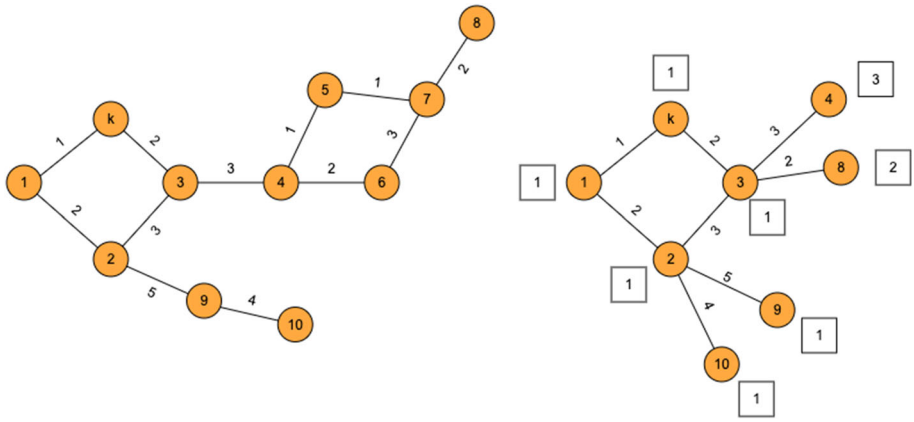

Fig. 5 An example of a graph (left) and its simplified version (right) with vertex weights. Vertices 4, 6, and 7 have been combined together into a vertex with weight three. Further, vertices 5 and 8 have been combined together into a vertex with weight two

graphs. We posit that highly connected graphs such as the grid networks are less amenable to the simplification method than sparser networks. In Table 3 we also include the percentage decrease in time and percentage increase in the best objective function value found via graph simplification to the Multi-Removal MIP removal results reported in Table 2. The time includes the time to perform the graph simplification, which is very efficient. For graphs with a significant reduction in the number of nodes and edges, we see a corresponding decrease in the runtime for the MIP. For the larger networks that did not terminate within the time limit, we only see the best vitality found improve in one instance.

7 Future work and conclusions

In this paper we have presented the VIMAX optimization problem that identifies a subset of vertices whose removal maximizes the volume of flow passing through a key vertex in the network. VIMAX is NP-Hard. We have used the dualize-and-combine method of Wood (1993) to formulate VIMAX as a mixed integer linear program, and we compared its performance to that of a simulated annealing heuristic. We also demonstrated how identifying vertices not having at least two vertex-disjoint paths to the key vertex can be used to simplify the graph and reduce computation time on certain graph types. Key limitations to the work presented are the computational bottlenecks. Future work could focus on two areas highlighted in this paper - graph simplification and the Bender's Decomposition.

- Graph Simplification: Additional properties of vitality-reducing vertices, such as those outlined in Paul (2012) for the unit capacity case, could be derived for the general capacity case and used to preprocess or simplify the graph to reduce the solution space of VIMAX. In particular, it would be beneficial to identify small cuts in the graph such that all vertices on the other side of the cut as k can be ignored from consideration.
- Bender's Decomposition: Because the number of constraints in the VIMAX MIP grows on the order of $O(|E||V|^2)$, we can use Bender's decomposition algorithm to solve our problem for large graphs. The decomposition is presented in Appendix C, but preliminary testing did not improve the MIP performance. The survey of Smith and Song illustrates a

variety of approaches that could be applied to improve the performance of the Bender's decomposition of VIMAX (Smith and Song 2020).

Additionally, this paper opens up a rich area of future research on extensions of this problem.

- Optimization: In this paper, we have focused on identifying vertices having high vitality effect on the key vertex without considering the cost or difficulty of removing them from the graph. An enhancement to VIMAX could include a budget constraint restricting the choice of subsets based on the difficulty of their removal.
- Dynamic response: The disruption technique described in this paper focuses on the network at one snapshot in time and assumes that any subset removal occurs simultaneously and that the network remains static. This might be a reasonable assumption for networks that evolve slowly over time, such as transportation supply chains, or for disruption interventions that occur over a short time scale, such as military maneuvers. On the other hand, social networks might be able to more rapidly reconfigure following a disruption, counter to our assumption. Extensions to VIMAX might explore cascading effects of sequential vertex removal, similar to the literature on multi-period interdiction (Enayaty-Ahangar et al., 2019), cascading failures (Crucitti et al., 2004a; Motter & Lai, 2002; Zhao et al., 2005), and agent-based models for counter-interdiction responses Magliocca et al., 2019).
- Imperfect information: The VIMAX formulation presented here assumes complete and perfect knowledge of the network's structure. However, the complete structure of a covert network is typically not known to enforcement agencies, and can evolve rapidly (Konrad et al., 2017). Future work could address applying VIMAX to networks with uncertain or unknown structure and capacities. For example, one could examine the robustness of the vitality measure and vitality-maximizing subset to graph perturbations over an uncertainty set.
- Robust network design: We can use the results of this research to design networks, such as telecommunication and other infrastructure networks, to be robust to vitality-diverting attacks (Crucitti et al., 2004b).
- Multiple key vertices: In the case that we want to maximize the flow through a subset S of key vertices, we can extend the definition of vitality maximization to maximize the all-pairs vitality of S. The MIP and simulated annealing algorithm can be updated accordingly.

VIMAX has broad applicability to problems including disrupting organized crime rings, such as those used in terrorism, drug smuggling and human trafficking; disrupting telecommunications networks and power networks; as well as robust network design.

Appendix A Proof of Theorem 1

In this section we prove Theorem 1 stating that the all-pairs vitality maximization problem is NP-Hard. Our proof extends the proof of Ovadia (2010) for the special case of undirected, unit-capacity edges. We first restate VIMAX as a decision problem: For a fixed value C, does there exist a subset S such that $\mathcal{L}_k(G \setminus S) \geq C$?

Theorem 1 *The all-pairs vitality maximization problem is NP-Hard.*

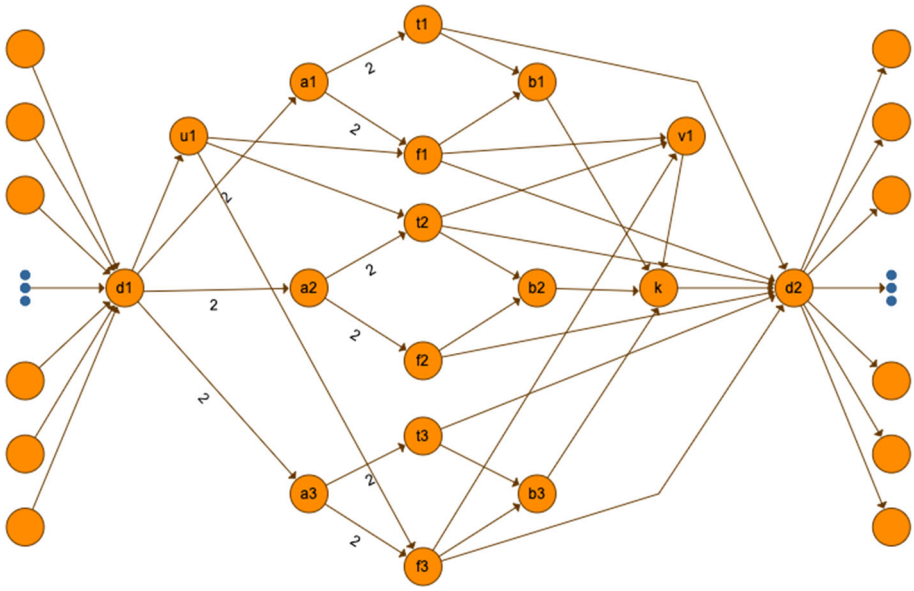

Fig. 6 Graph representation of a single clause 3SAT problem with three variables and the clause ($\overline{x_1}$ or x_2 or $\overline{x_3}$). All edge capacities equal one except where indicated otherwise

Proof We use a reduction from the 3-Satisfiability problem (3SAT). Given an instance of 3SAT with n boolean variables $x_1, x_2, \ldots, x_n$ and m clauses in 3-conjunctive normal form $c_1, c_2, \ldots, c_m$, the 3SAT decision problem is whether there is an assignment of variables to true/false values such that all clauses are satisfied. As an example with three variables, any assignment with x_3 set to false would satisfy the two clauses (x_1 or $\overline{x_2}$ or $\overline{x_3}$) and ($\overline{x_1}$ or x_2 or $\overline{x_3}$).

Given an instance of 3SAT, we construct a corresponding instance of VIMAX. We start building our directed graph G with three vertices d_1, k (the key vertex), and d_2 with an edge from k to d_2 with capacity $n + m$. Further, for each variable x_i we create four vertices $\{a_i, b_i, t_i, f_i\}$ and add edges (d_1, a_i), (a_i, t_i), and (a_i, f_i) each with capacity two and edges (t_i, b_i), (f_i, b_i), (t_i, d_2), (f_i, d_2), and (b_i, k) each with capacity one.

Then, for each clause c_j, we create two vertices u_j and v_j and add unit capacity edges (d_1, u_j) and (v_j, k). To encode this clause, for each variable x_i in clause c_j we add unit edges (u_j, t_i) and (t_i, v_j); for each variable $\overline{x_i}$ in clause c_j we add unit edges (u_j, f_i) and (f_i, v_j). Last, we create $M = 8 \cdot (m + n + n \cdot m)$ leaves with unit edges to d_1 and M leaves with unit edges from d_2 and set $C = (M + 1)^2(n + m)$. An example graph of a single-clause, three-variable, 3SAT problem having clause ($\overline{x_1}$ or x_2 or $\overline{x_3}$) is given in Fig. 6.

Note that the leaves adjacent to d_1 and d_2 essentially increase the weight of the flow between d_1 and d_2. In particular, if we define

$$\mathcal{L}_k^{s,t}(G \setminus S) := z_{st}(G \setminus S) - z_{st}(G \setminus (S \cup \{k\}))$$

and let V' be all vertices excluding these leaves as well as d_1, d_2, and k, then we can rewrite the all-pairs vitality as

$$\begin{aligned}\mathcal{L}_k(G \setminus S) = &\sum_{\substack{s,t \in V \setminus (S \cup \{k\}) \\ s \neq t}} \mathcal{L}_k^{s,t}(G \setminus S) \\ =&(M+1)^2 \mathcal{L}_k^{d_1,d_2}(G \setminus S) + \sum_{\substack{s,t \in V' \setminus S \\ s \neq t}} \mathcal{L}_k^{s,t}(G \setminus S) + \\ &(M+1) \sum_{s \in V' \setminus S} \left[\mathcal{L}_k^{d_1,s}(G \setminus S) + \mathcal{L}^{s,d_2}(G \setminus S) \right] \\ =&(M+1)^2 \mathcal{L}_k^{d_1,d_2}(G \setminus S) + (M+1) \sum_{s \in V' \setminus S} \mathcal{L}^{s,d_2}(G \setminus S).\end{aligned}$$

The last line holds since paths from d_1 to $s \in V' \setminus S$ or between s and $t \in V' \setminus S$ cannot travel through k. Further, we can bound the second half of the sum above by bounding the vitality by the capacity out of the starting node for each maximum flow.

$$(M+1) \sum_{s \in V' \setminus S} \mathcal{L}^{s,d_2}(G \setminus S) \leq (M+1)(4m + 9n + 2m \cdot n) \leq \frac{1}{2}(M+1)^2.$$

This shows that the maximum flow from pairs that are not $\{d_1, d_2\}$ contributes a trivial amount to the overall vitality. Therefore, finding a subset such that $\mathcal{L}_k(G \setminus S) \geq C = (M+1)^2(n+m)$ is equivalent to finding a subset S such that $\mathcal{L}_k^{d_1,d_2}(G \setminus S) \geq n + m$.

We now show that given an assignment of variables to boolean values that satisfy all clauses, we can find an equivalent subset S such that $\mathcal{L}_k^{d_1,d_2}(G \setminus S) \geq n + m$. Let S contain t_i for all i such that x_i is set to false and f_i for all i such that x_i is set to true.

Consider the maximum flow between d_1 and d_2 in $G \setminus S$. For each variable x_i such that $t_i \in S$, we send two units of flow: one along the path (d_1–a_i–f_i–b_i–k–d_2) and one along the path (d_1–a_i–f_i–d_2). If, instead, $f_i \in S$, then the paths change to use t_i instead of f_i. Further for each clause j, since this clause is satisfied, there exists at least one vertex t_i or f_i adjacent to u_j that is not in S. Without loss of generality, let this vertex be t_i. We send one unit of flow along the path (d_1–u_j–t_i–v_j–k–d_2). The overall flow has value $2n + m$. Since all edges adjacent to d_1 are saturated, this is a maximum flow.

Now consider the maximum flow between d_1 and d_2 in $G \setminus (S \cup \{k\})$. For each variable x_i such that $t_i \in S$, we send one unit of flow along the path (d_1–a_i–f_i–d_2). If, instead, $f_i \in S$, then the path changes to use t_i instead of f_i. The overall flow has value n. Since all edges adjacent to d_2 are saturated in $G \setminus (S \cup \{k\})$ this is a maximum flow. This shows that $\mathcal{L}_k^{d_1,d_2}(G \setminus S) \geq n + m$.

We must now show the reverse direction to complete the proof. Suppose that we have found a subset S such that $\mathcal{L}_k(G \setminus S) \geq C$. Then, given that all pairs except d_1 and d_2 contribute at most $\frac{1}{2}(M+1)^2$ to the vitality, it must be the case that $\mathcal{L}_k^{d_1,d_2}(G \setminus S) \geq n + m$. We decompose the flow into unit flow paths from d_1 to d_2. Let $f(s,t)$ be the number of these paths that go from s to t in the maximum flow from d_1 to d_2 in $G \setminus S$ and $f'(s,t)$ be the number of paths from s to t in the maximum flow between d_1 and d_2 in $G \setminus (S \cup \{k\})$. Then,

$$\mathcal{L}_k^{d_1,d_2}(G \setminus S) = \sum_{i=1}^{n} \left[f(a_i, d_2) - f'(a_i, d_2) \right] + \sum_{j=1}^{m} \left[f\left(u_j, d_2\right) - f'\left(u_j, d_2\right) \right]. \quad \text{(A1)}$$

For the first term in Eq. A1, we can verify that $\left[f(a_i, d_2) - f'(a_i, d_2)\right] \leq 1$ if exactly one of t_i and f_i is in S and $\{a_i, b_i\} \cap S = \emptyset$ and at most zero otherwise. In particular, if t_i and f_i are both in S then $f(a_i, d_2) = f'(a_i, d_2) = 0$. If both t_i and f_i are not in S, then at most two units of flow can go from a_i to d_2 in both graphs and both t_i and f_i can avoid using vertex k. Only when exactly one of t_i or f_i has been chosen will at least one path be forced to go through vertex k. For the second term, each term is also at most one given the unit capacity of the edge from d_1 into u_j. Therefore,

$$\mathcal{L}_k^{d_1,d_2}(G \setminus S) = \sum_{i=1}^{n} \left[f(a_i, d_2) - f'(a_i, d_2)\right] + \sum_{j=1}^{m} \left[f\left(c_j, d_2\right) - f'\left(c_j, d_2\right)\right] \leq n + m. \quad \text{(A2)}$$

Since $\mathcal{L}_k^{d_1,d_2}(G \setminus S) \geq n + m$ this implies equality throughout and that $|\{t_i, f_i\} \cap S| = 1$ for all $i = 1, 2, \ldots, n$. For each variable for which t_i is in S, we set that variable to false. Otherwise, we set the variable to true. Last, in order for every clause to contribute at least one to the overall vitality, u_j must be adjacent to some t_i or f_i not in S. Given the design of our network, this indicates that the assignment satisfies that clause.

Overall, this shows that every 3SAT decision problem can be reduced to a VIMAX decision problem and that VIMAX is NP-Hard. □

Appendix B Proof of Theorem 2

Here we prove Theorem 2 stating that the removal of any vertex not having at least two vertex-disjoint paths to the key vertex k can never increase the vitality of k.

Theorem 2 *Let G be a graph with key vertex k, and let i be a vertex such that there do not exist at least two vertex-disjoint paths starting at i and ending at k. Let S be any vertex subset containing i, and let $T = S \setminus \{i\}$. Then, $\mathcal{L}_k(G \backslash S) \leq \mathcal{L}_k(G \backslash T)$. Therefore, T will have at least as large a vitality effect on k as S.*

Proof Let G be a graph with key vertex k and let i be a vertex such that there do not exist at least two vertex-disjoint paths starting at i and ending at k. Then there exists a cut vertex v whose removal would disconnect the graph into at least two components. We consider two cases, $v \neq i$ and $v = i$.

When $v \neq i$, then v separates a component G_k that includes k from a component G_i that includes i. Consider the maximum flow between an $s - t$ pair $(s, t \neq k)$.

- If both s and t are in G_i, the flow between them is unaffected by the removal of vertex k, whether or not vertex i is removed from the graph. This is because any optimal flow path that passes through vertex k must first go into and out of vertex v, creating a flow cycle, $s - \ldots - v - \ldots - k - \ldots - v - \ldots - t$, and thus is equivalent to a flow path that avoids G_k entirely, $s - \ldots - v - \ldots - t$.
- If both s and t are in G_k, their contribution to the vitality of k is unaffected by the removal of i by the same logic as above: any optimal flow path that passes through vertex i must go into and out of vertex v, creating a flow cycle, and thus is equivalent to a flow path that avoids G_i entirely.
- If, without loss of generality, $s \in G_i$ and $t \in G_k$, then the removal of vertex i may reduce the flow between $s - \ldots - v$, but the remainder of the path $v - \ldots - t$ is unaffected. Thus no additional flow can be routed through k when i is removed than when i is present.

When $v = i$, then i separates a component G_k that includes k from the remainder of the graph, G_i. In this case, the removal of i will eliminate all $s - t$ flow between $s \in G_i$ and

$t \in G_k$, regardless of whether or not k is in the graph. Thus, no additional flow can be routed through k when i is removed from the graph than when i is present. □

Appendix C Benders decomposition

Because the number of constraints in the VIMAX MIP grows on the order of $O(|E||V|^2)$, we can use Benders decomposition algorithm to solve our problem for large graphs. In our case, the integer master problem chooses the subset of vertices to remove; this problem has relatively few variables and constraints. Given a fixed removal subset, we are left with a large linear network flow subproblem that is guaranteed to have an integer optimal solution.

We see in Eq. 5 constraints that couple $w_{i,j}$, $x_{i,j,s,t}$, $\alpha_{i,j,s,t}$ and $\beta_{i,s,t}$. We let the q_i's and $w_{i,j}$'s be the variables in our master problem. Our initial master problem contains only the constraints related to the $w_{i,j}$'s and q_i's, representing the choice of subset to remove. Thus the master problem is

$$\begin{aligned}
&\text{Maximize } \mathcal{L}_k \\
&\text{subject to} \\
&\qquad \textstyle\sum_{i \in V} q_i \geq n - m \\
&\qquad q_k = 1 \\
&\qquad w_{i,j} \leq q_i, \quad \forall (i,j) \in E \\
&\qquad w_{i,j} \leq q_j, \quad \forall (i,j) \in E \\
&\qquad w_{i,j} \geq q_i + q_j - 1, \quad \forall (i,j) \in E \\
&\qquad \\
&\qquad q_i \text{ binary}, \quad \forall i \in V \\
&\qquad w_{i,j} \geq 0, \quad \forall (i,j) \in E \\
&\qquad \mathcal{L}_k \geq 0.
\end{aligned} \tag{C3}$$

Here, $\mathcal{L}_k$ represents the optimal vitality of k. It currently has no restrictions on its value.

Solving Eq. C3 determines a feasible $\mathbf{z}$ and $\mathbf{w}$, which we can use to compute the vitality of k in the dual of the linear subproblem. When taking the dual we let $\gamma_{i,s,t}$ be the dual variables corresponding to the flow balance constraints of the $x_{i,j,s,t}$'s and $\delta_{i,j,s,t}$ be the dual variables corresponding to the capacity constraints on the $x_{i,j,s,t}$'s. Similarly, we let $\zeta_{i,j,s,t}$ be the dual variables corresponding to the edge constraints on $\alpha_{i,j,s,t}$, and we let $\eta_{s,t}$ be the dual variables corresponding to the constraints on the relationship between $\beta_{s,s,t}$ and $\beta_{t,s,t}$.

The linear subproblem becomes

$$\begin{aligned}
\text{Minimize} \quad & \sum_{\substack{s,t\in V'\\ s\neq t}} \sum_{(i,j)\in E} u_{i,j} w_{i,j} \delta_{i,j,s,t} + \sum_{\substack{s,t\in V'\\ s\neq t}} \eta_{s,t} - \\
& \sum_{\substack{s,t\in V'\\ s\neq t}} \sum_{(i,j)\in E'} (1 - w_{i,j}) \zeta_{i,j,s,t} \\
\text{subject to} \quad & \\
& \gamma_{i,s,t} - \gamma_{j,s,t} + \delta_{i,j,s,t} \geq 0, \forall (i,j) \in E, \forall s,t \in V' \\
& -\gamma_{s,s,t} + \gamma_{t,s,t} \geq 1, \forall s,t \in V' \\
& \sum_{j:(i,j)\in E'} \zeta_{i,j,s,t} - \sum_{k:(k,i)\in E'} \zeta_{k,i,s,t} = \begin{cases} \eta_{s,t} & \text{if } i = s \\ -\eta_{s,t} & \text{if } i = t \\ 0 & \text{otherwise} \end{cases} \\
& \qquad \forall i,s,t \in V' \\
& \zeta_{i,j,s,t} \geq -u_{i,j}, \forall (i,j) \in E', \forall s,t \in V' \\
& \\
& \gamma_{i,j,s,t} \geq 0, \forall (i,j) \in E, \forall s,t \in V' \\
& \delta_{i,s,t} \text{ unrestricted}, \forall i,s,t \in V' \\
& \eta_{s,t} \leq 0, \forall s,t \in V' \\
& \zeta_{i,j,s,t} \leq 0, \forall (i,j) \in E', \forall s,t \in V'.
\end{aligned} \tag{C4}$$

At the beginning of each iteration c, the master is solved and we obtain the optimal values for q_i and $w_{i,j}$. Initially, we start with an infinite objective function and all $q_i = 1$. The dual of the linear subproblem, shown in Eq. C4, is then solved with the optimal $w_{i,j}$'s substituted in.

If the subproblem is unbounded, simplex returns the extreme ray, defining γ^c, δ^c, η^c and ζ^c, and we add to the master problem the constraint

$$\sum_{\substack{s,t\in V'\\ s\neq t}} \sum_{(i,j)\in E} u_{i,j} w_{i,j} \delta^c_{i,j,s,t} + \sum_{\substack{s,t\in V'\\ s\neq t}} \eta^c_{s,t} - \sum_{\substack{s,t\in V'\\ s\neq t}} \sum_{(i,j)\in E'} (1 - w_{i,j}) \zeta^c_{i,j,s,t} \geq 0.$$

If the subproblem has an objective function value less than or equal to the incumbent value of $\mathcal{L}_k$, then we add to the master problem the constraint

$$\sum_{\substack{s,t\in V'\\ s\neq t}} \sum_{(i,j)\in E} u_{i,j} w_{i,j} \delta^c_{i,j,s,t} + \sum_{\substack{s,t\in V'\\ s\neq t}} \eta^c_{s,t} - \sum_{\substack{s,t\in V'\\ s\neq t}} \sum_{(i,j)\in E'} (1 - w_{i,j}) \zeta^c_{i,j,s,t} \geq \mathcal{L}_k.$$

Otherwise, the algorithm terminates. We set a max difference of $1e-5$ between $\mathcal{L}_k$ and the subproblem objective function value as our definition of convergence and stopping condition.

Preliminary testing of the Benders decomposition of VIMAX reveals the same problem that plagues large instances of the MIP formulation: the objective function values of the linear subproblems encountered are quite large compared to the objective function value of any feasible integer solution. Thus, the cuts added do not adequately constrain the master problem. Future work is needed to develop improved Benders decompositions.

Acknowledgements This work used the Extreme Science and Engineering Discovery Environment (XSEDE) (Towns et al., 2014), which is supported by National Science Foundation grant number ACI-1548562. Specifically, this work used the XSEDE Bridges-2 Extreme Memory and Regular Memory supercomputers at the Pittsburgh Supercomputing Center through allocation MTH210021. We thank consultant T. J. Olesky for their assistance troubleshooting batch calls to AMPL, which was made possible through the XSEDE Extended

Collaborative Support Service (ECSS) program (Wilkins-Diehr et al., 2016). The authors would also like to acknowledge Doug Altner, Michael Ernst, Elizabeth Ferme, Sam Gutekunst, Danika Lindsay, Yaniv Ovadia, Sean Plott, and Andrew S. Ronan for their contributions to early efforts in this work (Martonosi et al., 2011; Gutekunst, 2014; Ovadia, 2010). This work was supported by the National Science Foundation Research Experiences for Undergraduates program (NSF-DMS-0755540).

Funding Open access funding provided by SCELC, Statewide California Electronic Library Consortium This work was supported by the National Science Foundation Research Experiences for Undergraduates program (NSF-DMS-0755540).

Code and data availability https://github.com/alicepaul/network_interdiction

Declarations

Conflict of interest The authors have no financial or proprietary interests in any material discussed in this article.

References

Albert, R., Jeong, H., & Barabási, A. L. (2000). Error and attack tolerance of complex networks. *Nature, 406*(6794), 378–382.

Alderson, D. L., Brown, G. G., Carlyle, W. M., & Cox, L. A. (2013). Sometimes there is no "most-vital" arc: Assessing and improving the operational resilience of systems. *Military Operations Research, 18*(1), 21–37.

Alderson, D. L., Brown, G. G., & Carlyle, W. M. (2015). Operational models of infrastructure resilience. *Risk Analysis, 35*(4), 562–586.

Altner, D. S., Ergun, O., & Uhan, N. A. (2010). The maximum flow network interdiction problem: Valid inequalities, integrality gaps and approximability. *Operations Research Letters, 38*, 33–38.

Anzoom, R., Nagi, R., & Vogiatzis, C. (2021). A review of research in illicit supply-chain networks and new directions to thwart them. *IISE Transactions, 54*(2), 134–158. https://doi.org/10.1080/24725854.2021.1939466

Ausiello, G., Franciosa, P. G., Lari, I., & Ribichini, A. (2019). Max flow vitality in general and st-planar graphs. *Networks, 74*(1), 70–78. https://doi.org/10.1002/net.21878

Balcioglu, A., & Wood, R.K. (2003). In Woodruff, D.L. (ed.) Enumerating Near-Min s-t Cuts. Network Interdiction and Stochastic Integer Programming, pp. 21–49. Kluwer Academic Publishers, Norwell, MA, United States.

Basu, A. (2005). Social network analysis of terrorist organizations in india. In Proceedings of the 2005 Conference of the North American Association for Computational Social and Organizational Science.

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

Brown, G. G., Carlyle, M. W., Salmerón, J., & Wood, R. K. (2006). Defending critical infrastructure. *Interfaces, 36*, 530–544.

Callaway, D. S., Newman, M. E. J., Strogatz, S. H., & Watts, D. J. (2000). Network robustness and fragility: Percolation on random graphs. *Physical Review Letters, 85*(25), 5468–5471.

Carpenter, T., Karakostas, G., & Shallcross, D. (2002). *Practical issues and algorithms for analyzing terrorist networks*. Telecordia Technologies, Morristown, NJ: Technical Report.

Cavallaro, L., Ficara, A., De Meo, P., Fiumara, G., Catanese, S., Bagdasar, O., Song, W., & Liotta, A. (2004). Disrupting resilient criminal networks through data analysis: The case of Sicilian mafia. *PLOS One, 15*(8), 0236476. https://doi.org/10.1371/journal.pone.0236476

Church, R. L., Scaparra, M. P., & Middleton, R. S. (2004). Identifying critical infrastructure: The median and covering facility interdiction problems. *Annals of the Association of American Geographers, 94*, 491–502.

Cintron-Arias, A., Curet, N., Denogean, L., Ellis, R., Gonzalez, C., Oruganti, S., & Quillen, P. (2001). A network diversion vulnerability problem. Retrieved from the University of Minnesota Digital Conservancy, https://hdl.handle.net/11299/3553.

Corley, H. W., Jr., & Chang, H. (1974). Finding the *n* most vital nodes in a flow network. *Management Science, 21*(3), 362–364.

Cormican, K. J., Morton, D. P., & Wood, R. K. (1998). Stochastic network interdiction. *Operations Research, 46*, 184–197.

Crucitti, P., Latora, V., & Marchiori, M. (2004). Model for cascading failures in complex networks. *Physical Review E, 69*(4), 045104.

Crucitti, P., Latora, V., Marchiori, M., & Rapisarda, A. (2004). Error and attack tolerance of complex networks. *Physica A-Statistical Mechanics and its Applications, 340*(1–3), 388–394.

Cullenbine, C. A., Wood, R. K., & Newman, A. M. (2013). Theoretical and computational advances for network diversion. *Networks, 62*(3), 225–242. https://doi.org/10.1002/net.21514

Curet, N. (2001). The network diversion problem. *Military Operations Research, 6*(2), 35–44.

Dahan, M., Sela, L., & Aminc, S. (2022). Network inspection for detecting strategic attacks. *Operations Research, 70*(2), 1008–1024.

Dodds, P. S., Watts, D. J., & Sabel, C. F. (2003). Information exchange and the robustness of organizational networks. *Proceedings of the National Academy of Sciences of the United States of America, 100*(21), 12516–12521.

Enayaty-Ahangar, F., Rainwater, C. E., & Sharkey, T. C. (2019). A logic-based decomposition approach for multi-period network interdiction models. *Omega, 87*, 71–85.

Estrada, E. (2006). Network robustness to targeted attacks: The interplay of expansibility and degree distribution. *European Physical Journal B, 52*(4), 563–574.

Flaxman, A. D., Frieze, A. M., & Vera, J. (2007). Adversarial deletion in a scale-free random graph process. *Combinatorics Probability and Computing, 16*(2), 261–270.

Freeman, L. C., Borgatti, S. P., & White, D. R. (1991). Centrality in valued graphs: A measure of betweenness based on network flow. *Social Networks, 13*, 141.

Gallos, L. K., Argyrakis, P., Bunde, A., Cohen, R., & Havlin, S. (2004). Tolerance of scale-free networks: from friendly to intentional attack strategies. *Physica A-Statistical Mechanics and its Applications, 344*(3–4), 504–509.

Gallos, L. K., Cohen, R., Argyrakis, P., Bunde, A., & Havlin, S. (2005). Stability and topology of scale-free networks under attack and defense strategies. *Physical Review Letters, 94*(18), 188701.

Gallos, L.K., Cohen, R., Liljeros, F., Argyrakis, P., Bunde, A., & Havlin, S. (2006). Attack strategies on complex networks. In Computational Science - ICCS 2006, Pt 3, Proceedings **3993**, 1048–1055. http://www.springerlink.com/content/p31817656v18234j/fulltext.pdf.

Gierszewski, T., Molisz, W., & Rak, J. (2006). On certain behavior of scale-free networks under malicious attacks. *Computer Safety, Reliability, and Security, Proceedings, 4166*, 29–41.

Gomory, R. E., & Hu, T. C. (1961). Multi-terminal network flows. *SIAM Journal on Applied Mathematics, 9*, 551–556.

Grassi, R., Calderoni, F., Bianchi, M., & Torriero, A. (2019). Betweenness to assess leaders in criminal networks: New evidence using the dual projection approach. *Social Networks, 56*, 23–32. https://doi.org/10.1016/j.socnet.2018.08.001

Grubesic, T. H., Matisziw, T. C., Murray, A. T., & Snediker, D. (2008). Comparative approaches for assessing network vulnerability. *International Regional Science Review, 31*(1), 88–112.

Gusfield, D. (1990). Very simple methods for all pairs network flow analysis. *SIAM Journal on Computing, 19*, 143–155.

Gutekunst, S. (2014). Characterizing Forced Communication in Networks. Senior thesis (Claremont: Harvey Mudd College).

Holme, P., Kim, B. J., Yoon, C. N., & Han, S. K. (2002). Attack vulnerability of complex networks. *Physical Review E, 65*(5), 056109.

Holmgren, A. J. (2006). Using graph models to analyze the vulnerability of electric power networks. *Risk Analysis, 26*(4), 955–969.

Holzmann, T., & Smith, J. C. (2021). The shortest path interdiction problem with randomized interdiction strategies: Complexity and algorithms. *Operations Research, 69*(1), 82–99.

Israeli, E., & Wood, R. K. (2002). Shortest-path network interdiction. *Networks, 40*, 97–111.
Knuth, D.E. (2014). Art of Computer Programming, Volume 2: Seminumerical Algorithms. Addison-Wesley Professional, Boston, MA, United States.
Konrad, R. A., Trapp, A. C., Palmbach, T., & Blom, J. S. (2017). Overcoming human trafficking via operations research and analytics: Opportunities for methods, models, and applications. *European Journal of Operational Research, 259*(2), 733–745.
Koschützki, D., Lehmann, K. A., Peeters, L., Richter, S., Tenfelde-Podehl, D., & Zlotowski, O. (2005). Centrality indices. In U. Brandes & T. Erlebach (Eds.), *Network Analysis.* Berlin Heidelberg: Lecture Notes in Computer Science, Springer.
Lee, C., Cho, D., & Park, S. (2019). A combinatorial Benders decomposition algorithm for the directed multiflow network diversion problem. *Military Operations Research, 24*(1), 23–40.
Lei, X., Shen, S., & Song, Y. (2018). Stochastic maximum flow interdiction problems under heterogeneous risk preferences. *Computers and Operations Research, 90*, 97–109.
Lim, C., & Smith, J. C. (2007). Algorithms for discrete and continuous multicommodity flow network interdiction problems. *IIE Transactions, 39*, 15–26.
Magliocca, N. R., McSweeney, K., Sesnie, S. E., Tellman, E., Devine, J. A., Nielsen, E. A., Pearson, Z., & Wrathall, D. J. (2019). Modeling cocaine traffickers and counterdrug interdiction forces as a complex adaptive system. *PNAS, 116*(16), 7784–7792. https://doi.org/10.1073/pnas.1812459116
Martonosi, S.E., Altner, D.S., Ernst, M., Ferme, E., Langsjoen, K., Lindsay, D., Plott, S., & Ronan, A. (2011). A New Framework for Network Disruption. Unpublished manuscript. arxiv:1109.2954.
McMasters, A. W., & Mustin, T. M. (1970). Optimal interdiction of a supply network. *Naval Research Logistics Quarterly, 17*, 261–268.
Medina, A., Taft, N., Salamatian, K., Bhattacharyya, S., & Diot, C. (2002). Traffic matrix estimation: Existing techniques and new directions. *ACM SIGCOMM Computer Communication Review, 32*(4), 161–174.
Memon, N., Harkiolakis, N., & Hicks, D.L. (2008). Detecting high-value individuals in covert networks: 7/7 London bombing case study. In IEEE/ACS International Conference on Computer Systems and Applications, Doha, (pp. 206–215).
Morselli, C., Giguère, C., & Petit, K. (2007). The efficiency/security trade-off in criminal networks. *Social Networks, 29*, 143–153.
Motter, A. E., & Lai, Y. C. (2002). Cascade-based attacks on complex networks. *Physical Review E, 66*(6), 065102.
Natarajan, M.: In: Natarajan, M., & Hough, M. (eds.) (2000). Understanding the Structure of a Drug Trafficking Organization: A Conversational Analysis. From Illegal Drug Markets: From Research to Prevention Policy, pp. 273–298. Criminal Justice Press/Willow Tree Press, United States.
Ovadia, Y. (2010). Computational Feasibility of Increasing the Visibility of Vertices in Covert Networks. Senior thesis (Claremont: Harvey Mudd College).
Paul, A. (2012). Detecting Covert Members of Terrorist Networks. Senior thesis (Claremont: Harvey Mudd College).
Paul, G., Sreenivasan, S., & Stanley, H. E. (2005). Resilience of complex networks to random breakdown. *Physical Review E, 72*(5), 056130.
Pay, B. S., Merrick, J. R. W., & Song, Y. (2019). Stochastic network interdiction with incomplete preference. *Networks, 73*, 3–22.
Phillips, C.A. (1993). The network inhibition problem. In Proceedings of the 25th Annual ACM Symposium on the Theory of Computing, pp. 776–785.
Rasti, S., & Vogiatzis, C. (2019). A survey of computational methods in protein-protein interaction networks. *Annals of Operations Research, 276*(1–2), 35–87. https://doi.org/10.1007/s10479-018-2956-2
Rasti, S., & Vogiatzis, C. (2022). Novel centrality metrics for studying essentiality in protein-protein interaction networks based on group structures. *Networks, 80*(1), 3–50. https://doi.org/10.1002/net.22071
Ratliff, H. D., Sicilia, G. T., & Lubore, S. H. (1975). Finding the *n* most vital links in flow networks. *Management Science, 21*(5), 531–539.
Royset, J. O., & Wood, R. K. (2007). Solving the bi-objective maximum-flow network-interdiction problem. *INFORMS Journal on Computing, 19*(2), 175–184.
Sageman, M. (2004). *Understanding Terror Networks* (p. 220). Philadelphia: University of Pennsylvania Press.
Sharkey, T. C., Nurre Pinkley, S. G., Eisenberg, D. A., & Alderson, D. L. (2021). In search of network resilience: An optimization-based view. *Networks, 77*(2), 225–254.
Smith, J. C., & Song, Y. (2020). A survey of network interdiction models and algorithms. *European Journal of Operational Research, 283*(3), 797–811. https://doi.org/10.1016/j.ejor.2019.06.024
Stephenson, K., & Zelen, M. (1989). Rethinking centrality: Methods and examples. *Social Networks, 11*(1), 1–37.

Sun, S., Liu, Z. X., Chen, Z. Q., & Yuan, Z. Z. (2007). Error and attack tolerance of evolving networks with local preferential attachment. *Physica A-Statistical Mechanics and its Applications, 373*, 851–860.

Tezcan, B., & Maass, K.L. (January 2023). Human trafficking interdiction with decision dependent success. engrxiv.org (2022). Accessed on 11 at https://doi.org/10.31224/osf.io/dt8fs.

Towns, J., Cockerill, T., Dahan, M., Foster, I., Gaither, K., Grimshaw, A., Hazlewood, V., Lathrop, S., Lifka, D., Peterson, G. D., Roskies, R., Scott, J. R., & Wilkins-Diehr, N. (2014). XSEDE: Accelerating scientific discovery. *Computing in Science & Engineering, 16*(5), 62–74. https://doi.org/10.1109/MCSE.2014.80

Vogiatzis, C., Veremyev, A., Pasiliao, E. L., & Pardalos, P. M. (2015). An integer programming approach for finding the most and the least central cliques. *Optimization Letters, 9*(4), 615–633. https://doi.org/10.1007/s11590-014-0782-2

Wasserman, S., & Faust, K. (1994). *Social Network Analysis: Methods and Applications* (p. 825). United States of America: Cambridge University Press.

Wilkins-Diehr, N., Sanielevici, S., Alameda, J., Cazes, J., Crosby, L., Pierce, M., & Roskies, R. (2016). An overview of the XSEDE extended collaborative support program. In High Performance Computer Applications - 6th International Conference, ISUM 2015, Revised Selected Papers. Communications in Computer and Information Science, vol. **595**, pp. 3–13. Springer, Germany. https://doi.org/10.1007/978-3-319-32243-8_1

Wollmer, R.D. (1963). Some methods for determining the most vital link in a railway network. Technical report, RAND Corporation, Santa Monica, CA.

Wood, R. K. (1993). Deterministic network interdiction. *Mathematical and Computer Modelling, 17*, 1–18.

Wu, J., Deng, H. Z., Tan, Y. J., & Zhu, D. Z. (2007). Vulnerability of complex networks under intentional attack with incomplete information. *Journal of Physics A-Mathematical and Theoretical, 40*(11), 2665–2671.

Zhang, J., Zhuang, J., & Behlendorf, B. (2018). Stochastic shortest path network interdiction with a case study of Arizona-Mexico border. *Reliability Engineering and System Safety, 179*, 62–73.

Zhang, Y., Roughan, M., Lund, C., & Donoho, D. (2003). An information-theoretic approach to traffic matrix estimation. In Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, pp. 301–312.

Zhao, L., Park, K. H., Lai, Y. C., & Ye, N. (2005). Tolerance of scale-free networks against attack-induced cascades. *Physical Review E, 72*(2), 025104.