The Bi-Objective Shortest Path Network Interdiction Problem: Subgraph Algorithm and Saturation Property

KAIMING XIAO, CHENG ZHU, WEIMING ZHANG, AND XIANGYU WEI
Science and Technology on Information Systems Engineering Laboratory, National University of Defense Technology, Changsha 410073, China

ABSTRACT Critical network defense and attack problems, usually modeled as network interdiction, is of great significance. It is of great interest from the interdictor’s view to study saturation property of resource allocations. This paper presents a bi-objective shortest path network interdiction model with node interdiction, in which the interdictor seeks to interdict a set of nodes in the network that is Pareto-optimal with respect to two objectives, i.e., maximization of shortest path length and minimization of resources cost. A novel subgraph sequence algorithm is developed to identify the efficient frontier through a sequence of single-objective problems. Novel subgraph decomposition algorithms are proposed to solve single-objective problems. Features and time complexity of those problems and algorithms are analyzed. Moreover, to make a trade-off between the resource investment and the interdiction performance, the definition and the computing method of saturation point are introduced in this paper. Both simulated grid network and real-world transportation network data are used to evaluate the performance of algorithms in numerical experiments. The results show that the subgraph sequence algorithm can achieve the optimal Pareto-frontier of this bi-objective problem effectively.

INDEX TERMS Network interdiction, shortest paths, bilevel programming, bi-objective optimization, Benders decomposition, interdiction saturation.

I. INTRODUCTION

Network interdiction problem usually involves two entities, an evader and an interdictor, who compete in a special case of Stackelberg game. The evader operates a network in order to optimize his objective function (shortest path, maximum flow, minimum cost flow, etc.), while the interdictor attempts to worsen the evader’s objective function by interdicting several arcs or nodes of the network using limited interdiction resources [1]. The interdiction models are widely applied to contexts involving industrial or organizational competition, such as nuclear smuggling interdiction [2], electrical grid analysis [3], multi-commodity networks [4], and infrastructures fortification [5], [6]. Shortest path network interdiction (SPNI) is a typical network interdiction problem [7]. In this problem, an evader attempts to find the shortest path from the origin to destination, while the interdictor attempts to maximize the shortest path length by interdicting some arcs of the network with limited resources [8]. This paper studies an extension of SPNI problem, i.e., bi-objective shortest path network interdiction (BOSPINI), in which the interdictor wishes to maximize the shortest path length while minimize the interdiction resources cost.

Over the past decades, research on network interdiction has an accelerated development, but most of them focus on single-objective optimization of interdiction problems. One of the earliest works in network interdiction field is a min-max network flow problem by Wollmer [9] with focus on the effects on the maximum flow by removing $k$ arcs from a planar network. Several extensions, such as multiple source and sink nodes, undirected arcs, and multiple interdiction resources, were considered in the maximum flow network interdiction problem by Wood [10]. To extend this problem to direct node interdiction, Kennedy et al. [11] introduced several transition constraints to achieve node interdiction rather than replacing each node with two artificial nodes and an artificial link, named as node splitting. After that, different SPNI models and algorithms have been
developed for military and industrial applications. Fulkerson and Harding [12] introduced the first SPNI problem in which they assumed that the length of arc increases with the amount of resource allocated in a linear pattern. Israeli and Wood [13] studied the SPNI problem, and formulated it as a bilevel mixed-integer programming (BLMIP), in which they proposed enhanced Benders decomposition algorithms with supervalid inequalities (SVIs) and local-search procedure to improve the efficiency. A cutting plane approach was proposed in [14] for the maximum flow interdiction problem, and experiments showed the effectiveness of implementing cutting planes in the branch-and-cut procedure. All of these models for exact optimization have in common that they are limited to small problem sizes for optimal solutions, since they are NP-complete themselves. Hence, some intelligence algorithms were proposed for sub-optimal solutions of SPNI problem [15], [16]. In the multi-commodity network defense problem, non-dominated sorted genetic algorithm is developed for the estimation of the Pareto frontier [4].

Meanwhile, various kinds of network interdiction extension problems were proposed from different perspectives. Some studies modeled network interdiction problems as simultaneous games, such as the two-person zero-sum game formulations [17], the critical infrastructure protection game [18], and the interdiction game of drone delivery systems [19]. Some extensions of dynamic network flow interdiction games under uncertainty were developed in various application scenarios [20], [21]. Besides, various interdiction models were developed for different applications. A Stackelberg game model was proposed to tackle the problem of railway freight pricing, which studied the optimal pricing decision of the railway transportation enterprise and customers’ optimal purchase decisions [22]. Wei et al. [23] proposed a shortest path network interdiction problem with goal threshold thereby making a balance between the interdiction performance and resource consumption. Fu and Modiano [24] proposed an flow interdiction model using adversarial traffic flow which can be used in scenarios, such as the stealth DoS attack. A dynamic resource allocation problem in network interdiction was studied in [25] where goal recognition methodology was used to assist the decision making of the interdictor when the evader’s goal is uncertain. For the median type service network, a bilevel partial facility interdiction model was developed for the identification of critical facilities, in which the objective is maximization of the cost of network service [26]. Xiao et al. [27] studied the dynamic interdiction problem against stealthy malware propagation in Cyber-Physical Systems, in which the persistent anti-malware process has been modeled as a shortest-path tree interdiction problem. Moreover, fortification problems based on interdiction settings has attracted more interest. For instance, a tri-level defense facility location model for full coverage in r-interdiction median problem was addressed in [28]. A recent research focuses on the fortification problem on a directed network that channels single-commodity resources to fulfill random demands delivered to a subset of the nodes, where a robust stochastic approximation approach was proposed to mitigates interdiction risks effectively [29].

Some approaches were developed to deal with multi-objective optimization problems, such as maximum flow network interdiction problem [30], [31], critical infrastructure protection [32], and multi-commodity network flow [4]. For maximum flow network interdiction problem, Royset and Wood [33] proposed an algorithm for computing the optimal Pareto-frontier of the bi-objective programming through a sequence of single-objective problems solved by Lagrangian relaxation and a specialized branch-and-bound algorithm. A multi-objective approach was developed for SPNI problem with arc improvement resource, where the interdiction resources are allocated on arcs and the evader is assumed to have the ability of network improvement by lowering the costs of some arcs [34]. An evolutionary algorithm was proposed by Rocco, et al. [30] to provide an approximation to the optimal Pareto-frontier of bi- and tri-objective optimization in maximum flow network interdiction problem. The first BOSPNI model, which considers concurrent optimization of two objectives (i.e., maximization of shortest path length and minimization of resources cost), was developed by Rocco and Ramirez-Marquez [35]. To solve this problem, they proposed an evolutionary algorithm based on Monte Carlo simulation, which provides an approximation to the optimal Pareto-frontier for BOSPNI problem.

All above, few studies have modeled the problem of SPNI with node interdiction which is a common interdiction pattern in the real defense and attack scenarios in critical network. Hence, there is a need to propose a model of SPNI with node interdiction to meet the application requirements. Although various kinds of algorithms have been developed to solve SPNI problems, the solvable scale is still limited and node interdiction specified method design is also an open issue. Accordingly, few exact algorithms were proposed to solve BOSPNI problem with node interdiction efficiently. Besides, few researchers studied the theoretical properties of network interdiction problem related to Pareto-frontier, especially the saturation properties of interdiction and the trade-off between the two objectives. This issue is essential for the interdictor with limited resources. Without the knowledge of saturation properties, interdiction resources might be waste if a normal algorithm allocation decision is used.

In this paper, an exact subgraph sequence algorithm for nodal BOSPNI problem is developed in which two objectives (i.e., maximization of shortest path length and minimization of interdiction resources cost) are taken into account. This subgraph sequence algorithm identifies the Pareto-optimal solutions through a sequence of single-objective problems: (1) SPNI problem for maximizing the shortest path length and (2) problem for minimizing resources cost of shortest path network interdiction (SPNIR). To solve the SPNI problem efficiently, two subgraph decomposition algorithms are proposed using the dual programming as the master problem in the Benders decomposition. Based on the dual of SPNI problem, the SPNIR problem is reformulated as a
In this section, the problem is formally introduced and then other models and algorithms are original. The algorithm for SPNI problem in Section III was inherited, and section VI. Section VII provides the conclusions. This paper The computational results and discussion are presented in solving BOSPNI is constructed. Analyses of the saturationlems BOSPNI, SPNI and SPNIR are defined specifically. Section III provides descriptions of subgraph decomposition which focused on single-objective nodal shortest path network interdiction, but only the subgraph decomposition algorithm for SPNI problem in Section III was inherited, and other models and algorithms are original.

II. PROBLEM FORMULATION

In this section, the problem is formally introduced and then formulated in the form of mathematical programmes.

A. PROBLEM DEFINITION

Let $G(N, A)$ be a directed graph, where $N = \{\{i, j\} | i, j \in N\}$ and $A = \{(i, j) | i, j \in N\}$ are the sets of nodes and arcs respectively. $i$ or $j$ denotes the node index, and nodes $s$ and $t$ are the origin and destination nodes of the evader, separately. The nominal length of arc $k \in A$ or node pair $(i, j) \in A$ is $c_k$. Let $FS(i)$ and $RS(i)$ be the forward and reverse sets respectively, i.e., $FS(i) = \{(i', j') | i' \in A | i' = i\}$, $RS(i) = \{(i', i') | i' \in A | i' = i\}$. Interdiction increases the length of arc $k \in FS(i)$ to $c_k + d_k$ when a node $i \in N$ is interdicted. If the value of $d_k$ is sufficient large, the model can apply to destructive interdiction making the node $i$ actually destroyed and impassable. The required amount of resource to interdict node $i$ is defined as $w_k$, and the amount of the available resource for interdiction is $R$. Note that $c_k$, $d_k$ and $w_k$ are all non-negative scalar.

Let $x_i$ be the interdictor variable representing whether or not the node $i$ is interdicted, and $w_k$ as the evader variable representing whether or not the arc $k$ is on the path the evader traverses. Note both $x_i$ and $w_k$ are binary, and the variables’ values equal to 1 if the items (i.e. nodes or arcs) are selected by interdictor or evader, and vice versa. Denote by $w_k$ the intermediate variable reflecting the nodal interdiction effects on arc $k$.

The nodal BOSPNI problem is defined as follows: while the evader wishes to traverse a path of minimum length between the origin node $s$ and the destination node $t$, the interdictor aims to interdict a set of nodes in the network that are Pareto-optimal with respect to two objectives, maximizing shortest path length $L$ and minimizing resources cost $R$.

B. FORMULATION

With consideration of two objectives: maximization of shortest path length and minimization of resources cost, the programming formulation of bi-objective SPNI in a directed network $G(N, A)$ is:

$$\text{[BOSPNI-P]} \ z^* = \min_{x, w} - \min_y \sum_{k \in A} (c_k + w_k d_k) y_k, \sum_{i \in N} r_i x_i \ (1)$$

$$s.t. \ \sum_{k \in FS(i)} y_k - \sum_{k \in RS(i)} y_k \begin{cases} \ 1 & \text{for } i = s \\ \ 0 & \forall \ i \in N \setminus \{s, t\} \end{cases} \ (2)$$

$$w_k = x_i, \ \forall \ i \in N, \ k \in FS(i) \ (3)$$

$$y_k, w_k, x_i \in \{0, 1\}, \ \forall \ i \in N, \ k \in A, \ (4)$$

where Eq. (2) is flow-balance constraint, and the Eq. (3) is used to convert the arcs interdiction to node interdiction rather than the node splitting method [11].

The efficient solutions of BOSPNI problem are identified by isolating one objective function and limiting the value of the other by a constraint. First, isolating the shortest path length objective and imposing a limit $R$ constraint on resources cost, the basic programming formulation for single-objective maximization of shortest path length can be obtained, which is the SPNI problem with node interdiction:

$$\text{[SPNI-P]} \ z^* = \max_{w, x} \min_y \sum_{k \in A} (c_k + w_k d_k) y_k \ (5)$$

$$s.t. \ \sum_{k \in FS(i)} y_k - \sum_{k \in RS(i)} y_k \begin{cases} \ 1 & \text{for } i = s \\ \ 0 & \forall \ i \in N \setminus \{s, t\} \end{cases} \ (6)$$

$$w_k = x_i, \ \forall \ i \in N, \ k \in FS(i) \ (7)$$

$$y_k, w_k, x_i \in \{0, 1\}, \ \forall \ i \in N, \ k \in A, \ (8)$$

where $X = \{x \in \{0, 1\}^{|N|} | p^T x \leq R\}$.

Second, to identify the efficient solutions of BOSPNI, the programming for minimizing objective $R$ with a constraint on shortest path length is needed, and the basic formulation is:

$$\text{[SPNIR-P]} \ R^* = \min_x \sum_{i \in N} x_i r_i \ (9)$$

$$s.t. \ \min_{x, w, y} \sum_{i \in N} (c_k + w_k d_k) y_k = L^*, \ (10)$$

$$\sum_{k \in FS(i)} y_k - \sum_{k \in RS(i)} y_k \begin{cases} \ 1 & \text{for } i = s \\ \ 0 & \forall \ i \in N \setminus \{s, t\} \end{cases} \ (11)$$

$$w_k = x_i, \ \forall \ i \in N, \ k \in FS(i) \ (12)$$

where $L^*$ is the optimal objective (viz. maximum shortest path length) of SPNI problem.
III. SHORTEST-PATH NETWORK INTERDICTON WITH NODE INTERDICTON

This section introduces the analysis of SPNI with node interdiction, and then several algorithms are designed to solve it.

A. ANALYSIS OF TIME COMPLEXITY

Before introduce the algorithm to solve SPNI with node interdiction, the time complexity of it is firstly to be analyzed. Theorem 1 demonstrates that SPNI with node interdiction is NP-hard, hence there is no polynomial-time algorithm for it unless \( P = NP \).

**Theorem 1:** The SPNI problem with node interdiction is NP-hard.

**Proof:** Firstly, a problem of finding a subset \( I \subseteq \mathcal{N} \) of nodes is introduced such that \( \sum_{i \in I} r_i \leq R \) and whose removal from \( G \) results in the largest increase in the length of the shortest path from \( s \) to \( t \), called k-most-vital-nodes problem (KVNP). It is clear that KVNP is a special case of SPNI with node interdiction.

Then it is proved that KVNP is NP-hard by showing that a k-most-vital-nodes recognition problem (KNRP) related to it is NP-complete. The KNRP is:

**INPUT:** A directed graph \( G = (N, A); r \geq 0, c \geq 0; R \geq 0; s, t \in \mathcal{N} \).

**OUTPUT:** Yes, if there is a set \( I \subseteq \mathcal{N} \) such that \( \sum_{i \in I} r_i \leq R \) and the total length of the shortest path from \( s \) to \( t \) in \( G(N \setminus I, A) \) is \( \geq U \); no, otherwise.

The existence of a polynomial-time algorithm for KVNP would imply the existence of a polynomial-time algorithm for KNRP. Hence, it is then demonstrated that KNRP is NP-complete by reducing the following knapsack problem (Knap) to it. The Knap as follows is known to be NP-complete [37].

**INPUT:** \( e_j, b_j \geq 0 \) for \( j = 1, 2, \ldots, n; E, B \geq 0 \).

**OUTPUT:** Yes, if there is a set \( S \subseteq \{1, 2, \ldots, n\} \) such that \( \sum_{j \in S} e_j \geq E \) and \( \sum_{j \in S} b_j \leq B \).

The reduction process is as follows by constructing a directed graph with \( 2n + 1 \) nodes labeled as 0, 1, 2, \ldots, 2n shown in Figure 1. In this graph, the removal costs of nodes 0, 1, 2, \ldots, \( n \) are all equal to \( B + 1 \), while \( b_j \) for \( j = 1, 2, \ldots, n \) is the removal cost of node \( n + j \). The length of arc \((j - 1, j)\) is \( e_j \) for \( j = 1, 2, \ldots, n \), while the lengths of other arcs are 0.

Finally, the problem KNRP can be defined on this graph with \( R = B, U = E \), and origin-destination pair \((s, t) = (0, 2n)\). It is clear that none of the nodes \( j - 1 \) for \( j = 1, 2, \ldots, n + 1 \) can be the member of \( I \). Hence, only nodes \( n + j \) for \( j = 1, 2, \ldots, n \) can be removed. For convenience, node \( n + j \) is simple denoted by index \( j \), and let \( I \) be any set of removed nodes that does not violate the budget constraint \( \sum_{j \in I} r_i \leq R \). Then, it is clear that \( \sum_{j \in S} b_j \leq B \) and the total length of the shortest path from \( s \) to \( t \) is \( \sum_{j \in S} e_j \). That is, there is a one to one correspondence between solutions of KNRP and solutions of Knap.

![Figure 1: A portion of the constructed graph in the reduction process.](image)

**FIGURE 1.** A portion of the constructed graph in the reduction process.

B. THE BASIC DECOMPOSITION ALGORITHM

The SPNI problem is a typical two-stage problem, and may be viewed as a bilevel programming. A linear dual of inner minimization of the problem can be used to convert this programming to a single maximization problem [13]:

\[
[\text{SPNI-D}] \quad z^* = \max_{x \in X, w} \pi_t - \pi_s \quad (13)
\]

\[
\text{s.t.} \quad \pi_j - \pi_i - d_k w_k \leq c_k, \quad \forall k \in A \quad (14)
\]

\[
w_k = x_i, \quad \forall i \in N, k \in FS(i) \quad (15)
\]

\[
\pi_s = 0. \quad (16)
\]

However, programming SPNI-D is time-consuming using a standard LP-based branch-and-bound algorithm [13]. Thus a basic decomposition algorithm for arc interdiction of SPNI was proposed by Israeli and Wood [13] based on the Benders decomposition. Since the similarity between arc and node interdiction, the algorithm can be applied to the problem of node interdiction:

\[
[\text{SPNI-B}] \quad \text{[Master(\hat{y})]} \quad z_{\hat{y}} = \max_{x \in X, \hat{y}} z \quad (17)
\]

\[
\text{s.t.} \quad z \leq c^T \hat{y} + w^T D \hat{y}, \quad \forall \hat{y} \in \hat{Y} \quad (18)
\]

\[
w_k = x_i, \quad \forall i \in N, k \in FS(i) \quad (19)
\]

\[
[\text{Sub(\hat{w})}] \quad z_{\hat{w}} = \min_{y} \sum_{k \in A} (c_k + \hat{w}_k d_k) y_k \quad (21)
\]

\[
\sum_{k \in FS(i)} y_k - \sum_{k \in RS(i)} y_k = \begin{cases} 1 & \text{for } i = s \\ 0 & \forall i \in N \setminus \{s, t\} \\ -1 & \text{for } i = t \end{cases} \quad (22)
\]

\[
y_k \geq 0, \quad \forall k \in A. \quad (23)
\]

where \( \hat{y}_k, \hat{w}_k, \hat{x}_i \) represent the arc and node incidence variables corresponding to an s-t path for evader and interdictor respectively; \( \hat{Y} \) denotes the collection of \( \hat{y} \) corresponding to a subset of all simple s-t paths in \( G \). Here let \( D = \text{diag}(d) \).

In order to solve the SPNI efficiently, Israeli and Wood [13] introduced Super-valid inequalities (SVIs) and the Local-Search procedure to improve the performance of decomposition algorithm. These technologies can be applied to [SPNI-B] because that the master problem of node interdiction is equivalent to that of arc interdiction by introducing of Eq. (19). However, the problem size which can
be solved efficiently by this algorithm is quite limited, since the master problem is hard to solve when $R$ and the size of network become large [13].

### C. THE INTERDICTION SUBGRAPH DECOMPOSITION ALGORITHM

This section develops an interdiction subgraph decomposition algorithm to solve the SPNI problem. This algorithm is a combination of the basic decomposition algorithm [SPNI-B] and the dual algorithm [SPNI-D], in which an interdiction subgraph strategy is introduced to achieve this algorithm.

**Definition 1:** (Interdiction Subgraph) For a given graph $G$ and a node incidence vector $\hat{y} \in \hat{Y}$, there is a node set $N' = \{i, j |(i, j) = k \in A, \hat{y}_k = 1\} \subseteq N$. The Interdiction Subgraph $G'(N', A')$ of graph $G(N, A)$ is defined as the induced subgraph of graph $G$, where $N' \subseteq N, A' = \{(i, j) | i, j \in N'\} \subseteq A$.

**Definition 2:** (Optimal Interdiction Subgraph) For a given graph $G(N, A)$, let $\tilde{y}^* \in \tilde{Y}$ be the optimal interdiction plan, and $N^* = \{i, j |(i, j) = k \in A, \tilde{y}_k^* = 1\}$. Then the interdiction subgraph induced by $N^*$ is defined as the Optimal Interdiction Subgraph $G^*$.

To illustrate definitions above, a simple case of network interdiction problem is introduced in Fig. 2. $s$ and $t$ are the origin and destination nodes. Numbers beside arcs in Fig. 1 (a) are the length of these arcs, i.e., $c_k$, and the added length $d_k$ is 1 here. The required resource of each node from node 1 to node 4 is assumed as 1, and nodes $s, t$ are assumed to be not interdictable. Also, the additional length for each node from node 1 to node 4 is assumed as 1, and nodes $N$ become large [13].

### Interdiction Subgraph Decomposition Algorithm

**Algorithm SPNI-S**

**Input:** An instance of SPNI.

**Output:** An optimal interdiction plan $x^*$.

**Step 0:** Initialization;

- $\hat{x} \leftarrow 0$; $\hat{z} \leftarrow -\infty$; $\hat{\tilde{z}} \leftarrow \infty$;

**Step 1:** Solve the [Sub($\hat{w}$)] problem for $\hat{y}$ and objective $\hat{z}_\hat{y}$;

- $\hat{Y} \leftarrow \hat{Y} \cup \hat{y}$; $G' \leftarrow G \cap \hat{Y}$;

If $\hat{z} < \hat{z}_\hat{y}$ then $\hat{z}_\hat{y} \leftarrow \hat{z}$ and $\hat{x} \leftarrow \hat{x}$;

If $\hat{z} = \hat{z}_\hat{y}$ then go to Step 3;

**Step 2:** Solve the [Master-Dual($G'$)] problem for $\hat{w}'$, $\hat{x}'$ and objective $z_{G'(\hat{y})}$;

- Extend $\hat{w}'$ to $\tilde{w}'$; extend $\hat{x}'$ to $\tilde{x}'$;

- $\tilde{z} \leftarrow z_{G'(-\hat{y})}$;

If $\tilde{z} > \hat{z}$ then go to Step 1;

**Step 3:** $x^* \leftarrow \tilde{x}$; Print $x^*$; Stop.
The main reason is that for a given set \( \hat{G} \) of an induced subgraph, there is always \( \forall i \in \mathbb{R} \), thus \( \forall z \in \mathbb{R} \). For the same SPNI problem, the number of interdicted paths \( \hat{G} \) is usually limited compared with that in \( G \).

**Theorem 3:** Algorithm SPNI-S correctly solves SPNI.

**Proof:** During the iteration of algorithm SPNI-S, if the maximum shortest path in \( G \) is contained by \( G' \), according to Theorem 1, the solution of [Master-Dual(\( G' \))] problem is equivalent to that of [SPNI-P], and the solution of [Sub(\( \hat{x}_1 \))] problem is equal to that of [SPNI-P] inherently, i.e., \( z = z^\ast = z \); if not, there is \( \hat{z} > z \), then algorithm goes to Step 1 and add a shorter path into \( G' \) gradually until the maximum shortest path in \( G \) is put into \( G' \). Therefore, the algorithm SPNI-S can obtain the optimal solution of SPNI. □

**Theorem 4:** Let \( I_B \) and \( I_S \) be the numbers of iterations of algorithms SPNI-B and SPNI-S respectively. Then, there is \( I_B \geq I_S \).

**Proof:** For both of these two algorithms, each time the algorithm reaches Step 1, the evader responds a new path \( \hat{Y} \) to the interdictor’s plan \( \hat{w} \). In algorithm SPNI-B, this new path is directly put into the set \( \hat{Y} \) for the [Master(\( \hat{Y} \))] to interdict. However, in algorithm SPNI-S, an interdiction subgraph \( G' \) is induced by \( \hat{Y} \) for the [Master-Dual(\( G' \))] to interdict. Let \( A \) and \( A' \) be the sets of arcs in \( \hat{Y} \) and \( G' \) respectively. Similarly, let \( P \) and \( P' \) be the sets of simple \( s-t \) paths in \( \hat{Y} \) and \( G' \) respectively. According to the definition of induced subgraph, there is \( \hat{A} \subseteq A' \), and thus \( \hat{P} \subseteq P' \). Therefore, in an iteration process, algorithm SPNI-S could interdict more possible simple paths than algorithm SPNI-B. For the same SPNI problem, the number of interdicted paths is constant, thus \( I_B \geq I_S \) is valid. □

Theorem 3 illustrates that algorithm SPNI-S can reduce iterations compared with algorithm SPNI-B. Generally, for a given graph, the iterations reduction of SPNI-S is remarkable. The main reason is that: for a given set \( \hat{Y} \), there are always some shorter arcs in \( \hat{Y} \), and these arcs could be recombined with former paths \( \hat{y} \) as the next shortest path by evader with high probability. Since the Master problem is solved on the interdiction subgraph, algorithm SPNI-S can avoid the iterations caused by the newly generated shortest path.

**Proposition 1:** The interdiction subgraph generation procedure in algorithm SPNI-S runs in a time \( O(N^3) \).

**Proof:** The interdiction subgraph generation procedure needs to select \( N' \) rows from \( N \) rows of graph \( G(N, A) \)'s adjacency matrix, and then select \( N' \) columns from \( N \) columns of the selected adjacency matrix. In the worst case, the interdiction subgraph \( G'(N', A') \) is the total graph \( G(N, A) \). Therefore, through this two steps, the time complexity of interdiction subgraph generation procedure is \( O(N^2) \).

Based on proposition 1, although the complexity of interdiction subgraph procedure is \( O(N^2) \), it is acceptable, since the number of nodes in \( G' \) is usually limited compared with that in \( G \).

**D. THE ALGORITHM WITH LOCAL-SEARCH PROCEDURE**

A Local-Search procedure was proposed by Israeli and Wood [13] to reduce the iterations of decomposition algorithm, named as SPNI-BE. Similarly, in order to enhance the algorithm SPNI-S, this local-search procedure is combined with SPNI-S as follows. This procedure exploits multiple pairs \( (\hat{w}, \hat{y}') \) generated by a local-search process for a given \( \hat{w} \), where \( \hat{y}' \) represents the optimal or suboptimal response of [Sub(\( \hat{w} \))] problem.

**Definition 3 (Local-Search):** For a given graph \( G(N, A) \), a standard shortest-path tree \( T \) (i.e. a path tree containing paths form \( s \) to all other nodes in \( G \) ) can be obtained by several normal algorithms. Let \( P(t) = \{s, i_1, i_2, \ldots, i_n, t\} \)
denote the shortest s-t path in G. Similarly, let P(j) denote the shortest s-j path. Then, \( \forall i_m \in \{i_1, i_2, \ldots, i_n\}, (j, i_m) \in A, j \notin P(t), \exists a extra path P'(t) = (P(j), i_{m+1}, \ldots, i_n, t) \) represented by a incidence vector \( \hat{y} \). The process to generate all these extra paths \( \hat{y} \) is defined as Local-Search procedure [13].

In this local-search procedure, a shortest-path tree \( T \) algorithm is needed. Here the Dijkstra algorithm is employed to generate the shortest-path tree \( T \) [38]. Then, the enhanced SPNI-S algorithm is proposed as:

Algorithm SPNI-SE: Enhanced interdiction subgraph decomposition algorithm for SPNI

**Input:** An instance of SPNI.

**Output:** An optimal interdiction plan \( x^* \).

**Step 0:** Initialization;

\[ \hat{x} \leftarrow 0; \hat{z} \leftarrow \infty; \hat{z} \leftarrow \infty; \]

**Step 1:** Solve the [Sub(\( \hat{w} \))] problem for shortest-path tree \( T \) and objective \( z^*_p \);

\[ \hat{y} \leftarrow \hat{Y} \cup (\hat{y}, \hat{y})' \]

If \( \exists z < z_p \) then \( \hat{z} \leftarrow z_p \) and \( \hat{x} \leftarrow \hat{x} \);

**Step 2:** Solve the [Master-Dual(\( G' \))] problem for \( \hat{x}' , \hat{w}' \) and objective \( z^*_{G'}(\hat{y}) \);

**Step 3:** \( x^* \leftarrow \hat{x} \); Print \( x^* \); Stop.

**Theorem 5:** Algorithm SPNI-SE correctly solves SPNI.

**Proof:** let \( P_L' \) be the set of all simple paths in \( G' \) generated by Local-Search(\( \hat{x} , T \) ). It is valid that \( P' \subseteq P_L' \subseteq P \) according to Definition 3. Thus, this proof is equivalent to the proof of Theorem 2, and it’s valid.

The enhanced algorithm SPNI-SE could reduce more iterations than SPNI-S, since the number of paths to be interdicted in one iteration of SPNI-SE could be larger than that of SPNI-S. Meanwhile, it should be noted that although the adopted shortest-path tree algorithm is not much more difficult than s-t shortest path algorithm, the time consumption of finding the shortest-path tree would affect the efficiency of algorithm SPNI-SE especially when the network is quite large.

**IV. BI-OBJECTIVE SHORTEST-PATH NETWORK INTERDICATION**

After solving the SPNI problem with node interdiction, SPNIR problem should be solved based on the SPNI solution. Then, the Pareto-frontier of the whole bi-objective optimization problem can be obtained.

**A. DUAL PROGRAMMING FOR MINIMIZING RESOURCES**

In order to identify the efficient solutions, i.e., Pareto-optimal solutions of BOSPNI, a dual programming for objective \( R \) based on results of SPNI problem is proposed to solve SPNIR-P, since it could not be solved directly. Here, the dual programming SPNI-D in section II is reformulated to the dual programming for minimization of \( R \) (SPNIR-D), which can be solved using standard branch-and-bound algorithm [39]. Since SPNIR-D is a typical mixed integer programming, it is still NP-hard. Aiming to minimize \( R \), the formulation of SPNIR-R is:

\[
\begin{align*}
[\text{SPNIR-D}] \quad & R^* = \min_{\pi, w, x} \pi^T x \\
\text{s.t.} \quad & \pi_j - \pi_i - d_{ij}w_{ij} \leq c_k, \quad \forall k \in A \\
& w_i = x_i, \quad \forall i \in N, \quad k \in F(i) \\
& \pi_s = 0, \\
& \pi_t = L^*, \\
& w_k, x_j \in \{0, 1\}, \quad \forall i \in N, \quad k \in A.
\end{align*}
\]

**B. SUBGRAPH SEQUENCE ALGORITHM**

A subgraph sequence algorithm for BOSPNI problem is proposed in this section. The first single-objective problem for maximization of shortest path length could be solved using algorithms SPNI-D, SPNI-S or SPNI-SE with specific limited resources \( \hat{R} \). Based on the results of SPNI(\( \hat{R} \)), the second single-objective problem for minimization of resources cost could be solved to obtain the Pareto-optimal solution (\( -L_p, R_p \)) by SPNIR-D, and then to identify the Pareto-frontier by assigning different \( \hat{R} \).

Since the efficiency of algorithm SPNI-S is better than algorithms SPNI-D and SPNI-B, the result \( L^* \) from algorithm SPNI-S is used as the input of SPNIR-D, and solve SPNIR-D in the graph \( G(N, A) \). Based on the two single-objective programmings SPNI-S and SPNI-D, a subgraph sequence algorithm for BOSPNI problem can be obtained:

Algorithm BOSPNI-S: Subgraph sequence algorithm for BOSPNI

**Input:** An instance of BOSPNI.

**Output:** A solution (\( -L^*, R^* \)) and an interdiction plan \( x^* \).

**Step 0:** Initialization: \( R \leftarrow \hat{R} \);

**Step 1:** Solve SPNI problem with algorithm SPNI-S for \( L^* \);

**Step 2:** Solve SPNIR problem with programming SPNIR-D in \( G \) for \( R^* \) and \( x^* \);

**Step 3:** Print (\( -L^*, R^* \)) and \( x^* \); Stop.

**Theorem 6:** The solution of BOSPNI-S is the efficient solution of BOSPNI problem, i.e., (\( -L_p, R_p \)).

**Proof:** For a given BOSPNI problem with specific resources \( \hat{R} \), let (\( -L^*, R^* \)) be the optimal solution of BOSPNI-S, and let \( S, Z = f(S) \) be the variable and objective feasible region of BOSPNI, respectively. Let (\( -L, R \) \( \in Z, \forall R \in [0, R^*] \)), i.e., \( R \leq R^* \), there is \( -L \geq -L^* \). Since \( R^* \) obtained by SPNI-R is the minimum \( R \) when \( -L = -L^* \), \( \forall R \leq R^* \); \( \hat{R} = \hat{L} = -L^* \). Thus, \( \hat{R} x_j \in S: (-L, R) < (-L^*, R^*) \). In addition, \( \forall R \in (R^*, +\infty) \), i.e., \( R > R^* \),
there is \(-L \leq L^*\), thus, \(\exists x \in S: (\mathbf{L}, R) < (L^*, R^*)\) 

Therefore, optimal solution of BOSPNI-S \((-L^*, R^*)\) is the efficient solution of BOSPNI problem, i.e., \((-L_p, R_p)\). □

According to Theorem 6, the solution of OBOSPNI-S is the efficient solution, i.e., the Pareto-optimal solution of BOSPNI problem. The Pareto-frontier of BOSPNI problem can be obtained by using algorithm BOSPNI-S with different specific resources \(\bar{R}\).

V. INTERDICATION SATURATION

In this section, the interdiction saturation properties of shortest path network interdiction are discussed. As the available resources \(\bar{R}\) increase, the optimal objective \(L^*\) of SPNI problem would not decrease inherently. However, there is a bound of objective \(L^*\), i.e., a finite value of path length for non-destroyed interdiction or an infinite value for destroyed interdiction. This extreme limit of objective \(L^*\) presents the maximum damage that the an interdictor can make, which can be viewed as a metric of robustness for SPNI problem. To achieve this extreme limit of \(L^*\), the resources \(\bar{R}\) must reach to a saturated value, and this saturated value is usually much smaller than that of the total interdiction resource cost of the whole network, i.e., \(\sum_{i \in N \setminus \{s,t\}} r_i\). Therefore, it is necessary to define and analyse this interdiction saturation phenomenon.

Theorem 7: The solutions of SPNI-S and SPNI-D are both weak efficient solutions of BOSPNI problem, i.e., \((-L_{wp}, R_{wp})\) and \((-L_{wp}, R_{wp})\) be the weak efficient solutions of SPNI-S and SPNI-D, respectively.

There is \(R_{wp} \leq \sum_{i \in N \setminus \{s,t\}} r_i\) and \(R_{wp} \leq \sum_{i \in N \setminus \{s,t\}} r_i\).

Proof: Firstly, for a given SPNI problem with specific resources \(\bar{R}\), let \((-L^*, R^*)\) be the optimal solution of SPNI-S. Let \(S, Z = f(S)\) be the variable and objective feasible region of BOSPNI, respectively. Let \((-L, R) \in Z, \forall R \in [0, R_\star]\), i.e., \(R \leq R_\star\), there is \(-L \geq -L^*,\) thus, \(\exists x \in S: (\mathbf{L}, R) < (L^*, R^*)\).

In addition, \(\forall R \in (R^*, +\infty), i.e., R > R_\star\), there is \(-L \leq -L^*\), thus, \(\exists x \in S: (\mathbf{L}, R) < (L^*, R^*)\). Therefore, optimal solution of SPNI-S \((-L^*, R^*)\) is the weak efficient solution of BOSPNI problem, i.e., \((-L_{wp}, R_{wp})\). Similarly, the proof for SPNI-D can be obtained.

Secondly, for SPNI-S, the problem is solved on the optimal interdiction subgraph \(G\), thus \(\forall R \in [0, +\infty), there is R_{wp} \leq \sum_{i \in N \setminus \{s,t\}} r_i\). For SPNI-D, it is solved on the graph \(G\), thus \(\forall R \in [0, +\infty), there is R_{wp} \leq \sum_{i \in N \setminus \{s,t\}} r_i\). □

The weak efficient solutions of BOSPNI problem can be obtained by solving SPNI problem with single objective to maximize the shortest path length with specific resources according to Theorem 7, and the upper bound of \(R_{wp}\) is lower than that of \(R_{wp}\) which means fewer resources waste in most cases. When \(R_{wp}\) reaches its upper bound, \(L_{wp}\) reaches its extreme limit inherently, thus a definition of weak saturation solution can be introduced.

Definition 4 (Weak Saturation Solution): The weak saturation solution of BOSPNI is defined as: \((-L_{ws}, R_{ws}) = arg\max((-L_{wp}, R_{wp})\).

Corollary 1: Let \((-L_{ws}^S, R_{ws}^S)\) and \((-L_{ws}^D, R_{ws}^D)\) be the weak saturation solutions of SPNI-S and SPNI-D, respectively.

There is \(R_{ws}^S = \sum_{i \in N \setminus \{s,t\}} r_i \leq R_{ws}^D = \sum_{i \in N \setminus \{s,t\}} r_i\).

In addition, \((-L_{ws}^S, R_{ws}^S) \leq (-L_{ws}^D, R_{ws}^D)\).

Proof: Firstly, according to Theorem 7, the solutions of SPNI-S and SPNI-D are both weak efficient solutions of BOSPNI problem. For SPNI-S, the problem is solved on the optimal interdiction subgraph \(G\), thus \(\forall R \in [0, +\infty), there is R_{wp} \leq \sum_{i \in N \setminus \{s,t\}} r_i\), i.e., \(R_{ws}^S = \sum_{i \in N \setminus \{s,t\}} r_i\). For SPNI-D, it is solved on the graph \(G\), thus \(\forall R \in [0, +\infty), there is R_{wp} \leq \sum_{i \in N \setminus \{s,t\}} r_i\). Meanwhile, since \(G^* \subseteq G\), there is \(R_{ws}^S \leq R_{ws}^D\). Secondly, according to Theorem 2, the maximum shortest path lengths of SPNI-S and SPNI-D are equal when the interdiction subgraph is optimal interdiction subgraph \(G^*\), that is \(-L_{ws}^S = -L_{ws}^D\). Therefore, \((-L_{ws}^S, R_{ws}^S) \leq (-L_{ws}^D, R_{ws}^D)\).

It can be concluded that the weak saturation solution of SPNI-S is better than that of SPNI-D according to Corollary 1, i.e., algorithm SPNI-D tends to waste more resources than SPNI-S for a same SPNI problem while the objective \(-L_{ws}^S = -L_{ws}^D\).

Definition 5 (Saturation Solution, Saturation Point): The saturation solution of BOSPNI is defined as: \((-L_s, R_s) = arg\max((-L_p, R_p)\). The point \((-L_s, R_s)\) is defined as saturation point of BOSPNI problem.

Theorem 8: For a given graph \(G(N, A)\), let \(\bar{R} = \sum_{i \in N \setminus \{s,t\}} r_i\). The solution of BOSPNI-S with \(\bar{R}\) is the saturation solution of BOSPNI.

Proof: Let \((-L_p, R_p)\) be the solution of BOSPNI-S with \(\bar{R} = +\infty, \forall R \in [0, +\infty), there is \(-L^* \geq \bar{R}\) according to Step 1 of BOSPNI-S. If there is another efficient solution \((-L^*_p, R^*_p)\): \(\bar{R}^*_p > \bar{R}_p\), then there are \(\bar{R}^*_p > \bar{R}_p\) and \(-L^*_p \geq \bar{R}_p\), i.e., \(\exists (\bar{L}_p, \bar{R}_p) < (-L^*_p, R^*_p)\), thus \((-L^*_p, R^*_p)\) is not the efficient solution of BOSPNI, which is inconsistent with the assumption. Therefore, \(\bar{R}_p\) is the maximum efficient resource solution. Meanwhile, the solution of BOSPNI-S with \(\bar{R} = +\infty\) is the same with the solution with \(\bar{R} = \sum_{i \in N \setminus \{s,t\}} r_i\), which means all node could be interdicted, therefore the solution of BOSPNI-S with \(\bar{R}\) is the saturation solution of BOSPNI.

According to Theorem 8, the saturation point of BOSPNI problem can be obtained by algorithm SPI:

Algorithm SPI: Algorithm for Saturation Point Identification

Input: An instance of BOSPNI.

Output: The saturation point \((-L_s, R_s)\) and an interdiction plan \(x^*\).

Step 0: Initialization: \(R \leftarrow \bar{R} = \sum_{i \in N \setminus \{s,t\}} r_i\).

Step 1: Solve BOSPNI problem with algorithm BOSPNI-S for efficient solution \((-L_p, R_p)\) and \(x^*\).

Step 2: \((-L_s, R_s) \leftarrow (-L_p, R_p)\), and print \((-L_s, R_s)\) and \(x^*\); Stop.
and $R_s$ is the minimization of resources cost to achieve $L_s$. If $d_r \rightarrow +\infty$, which means destroyed interdiction, there is $L_s \rightarrow +\infty$; otherwise, $L_s$ is a finite value. The saturation point of BOSPNI can also help us to make a trade-off between the resource investment and the interdiction performance of a network with respect to shortest path interdiction, though a full Pareto-frontier can provide more details.

VI. EXPERIMENTS AND DISCUSSIONS

In this section, test settings and the computing environment are introduced first. Then, results of computational experiments on both simulated grid networks and real road networks are presented and discussed.

A. TEST PROBLEMS AND ENVIRONMENT

A set of simulated directed grid networks and two real cases of transportation networks are used as test problems to illustrate the algorithm performance in shortest path network interdiction. The generation of simulated data followed the general generation principles in the field of network interdiction and the standard parameter setting guidelines was followed as well [8], [13].

The simulated networks are $n_1 \times n_2$ grid networks, where $n_1$ and $n_2$ are the number of rows and columns, respectively. Each node at grid position $(r, c)$ has a probability of $p$ to connect to its neighbour nodes at grid positions $(r - 1, c - 1), (r - 1, c), (r - 1, c + 1), (r, c - 1), (r, c + 1), (r + 1, c - 1), (r + 1, c), (r + 1, c + 1)$. If there exists a node at that particular position. For the SPNI problem, let node $(1, 1)$ and node $(n_1, n_2)$ be the origin node and destination node respectively, which are assumed to be not interdictable. In addition, the integer normal length $c_k$ and additional length $d_k$ for each interdictable arc $k$ are assumed to be uniformly distributed on $[1, c]$ and $[1, d]$, respectively. The resources consumption $r_i$ for each interdictable node $i$ is assumed as the node’s degree. With these assumptions, 10 instances were created by randomly generating different $r$, $c$, and $d$ for each test set. Also, $t$-test on the results of the following different algorithms were conducted, and the results were statistically significant in difficult cases.

The real cases of transportation networks evaluated are the Oldenburg Road Network [40] and the Chicago Sketch Road Network [41]. There are 6,105 nodes and 14058 arcs in Oldenburg road network shown in Fig. 3, and the Chicago Sketch Road Network shown in Fig. 4 (a) contains 933 nodes and 2,950 arcs. Here, these networks are assumed to be indirected, and the length of each arc $c_k$ is a approximate integer of the real distance between two nodes.

Algorithms above are coded with the MATLAB toolbox YALMIP [42] using the CPLEX 12.5 callable library, and tested on a Windows7 (32) computer with 2.40 GHz Intel(R) Core i5 CPU and 3.0G RAM.

B. ALGORITHMS PERFORMANCE FOR SINGLE-OBJECTIVE PROBLEM

Firstly, the performance of different algorithms for SPNI problem is investigated as the total interdiction resource $R$ changes. As shown in Table. 1, algorithms SPNI-S and SPNI-SE are faster than other algorithms especially when $R$ is larger. The iteration number of SPNI-S is smaller than that of SPNI-B reporting 14 v.s. 36 when $R = 40$. The local-search procedure can also introduce an improve in iteration number reduction.

All algorithms are sensitive to $R$. As $R$ increases from a small value, time increases rapidly. But when $R$ is sufficiently large, the solving time will decrease as $R$ increases. The reason of this phenomenon is similar to Israeli’s explanation for arc interdiction problem [13], i.e., firstly increasing $R$ allows more combinations of nodes to be interdicted which is harder for the branch and bound procedure; then, if $R$ is sufficiently large, almost all nodes can be interdicted leading to a simple interdiction problem.

The sensitivity of the algorithms to the network size $(n_1, n_2)$ is shown in Table 2. Since the origin and destination nodes are $(1, 1)$ and $(n_1, n_2)$, $n_1$ and $n_2$ are equivalent for network size. Here, algorithms on 9 sets of instances are tested which row-to-column ratio is 1. Results in Table 2 show that SPNI-SE is the fastest of the five algorithms. When network size become larger, algorithms SPNI-S and SPNI-SE are faster than the other three algorithms by a wide margin. Moreover, SPNI-S and SPNI-SE hold relatively smaller values of standard deviation of $T$ compared with SPNI-B and SPNI-BE, which illustrates a better stability in solving various problems.

Comparing with SPNI-B, algorithm SPNI-S has fewer number of iterations in average, which agrees with the theoretical analysis above in Theorem 5. That’s the main reason why the proposed interdiction subgraph decomposition algorithm is faster than the basic decomposition algorithm. Algorithm SPNI-SE has fewer number of iterations in average than algorithm SPNI-S, since the Local-search procedure is adopted. The Local-search procedure can speed up the algorithm to converge to the optimal solution mentioned in Section III part D.
To study the efficiency and scalability of those algorithms, a real-world transportation network is evaluated for case study. The additional length $d_k$ for each interdictable arc $k$ is assumed as an integer multiple of $c_k$. Also, the resources consumption $r_i$ for each interdictable node $i$ is assumed as the node’s degree, and the origin and destination nodes are defined shown in Fig. 3.

Results tested on the transportation network are shown in Table 3. It can be observed that the arcs of real networks is less intensive than that of grid networks. Hence, the SPNI programming is easier to solve on this real network. However, when the network size $G(N, A)$ is quite large, especially $N$ is large, the algorithm SPNI-BE and SPNI-SE become very time-consuming, since the complexity of adopted Local-search procedure is $O(N^3)$. In addition, although some improvements could be used to reduce the running time to $O(NA\lg N)$ or $O(N^2 \lg N + NA)$ [43], the Local-search procedure is not efficient when $N$ is quite large. On the
contrast, when the total interdiction resource $R$ is large, the running time of SPNI-S is still acceptable. The reason is that the procedure of the interdiction subgraph generation runs in a time $O(N^2)$ proved in Proposition 1.

To sum up, the proposed algorithm SPNI-S performs better than other algorithms especially when the available resource $R$ or the network size (i.e., the scale of the problem) is large. This superiority of SPNI-S in time efficiency comes from the iteration decrease which is the result of applying the proposed interdiction subgraph. Besides, although the Local-search procedure can, to some extent, improve algorithm’s time efficiency, the performance will degenerate as the increase of nodes number in the graph. Hence, SPNI-S can be used to solve relatively large problems in less time which is an advantage in application.

C. ALGORITHM PERFORMANCE AND SATURATION PROPERTY FOR BI-OBJECTIVE PROBLEM

Seven grid networks with different size and a real transportation network were used shown in Fig. 4 (a) (viz. Chicago Sketch Road Network) to perform the effectiveness of algorithm BOSPNI-S. It is assumed that $d_k = \min_{c \in A} c_k$ to make it easier for BOSPNI problem solving with different $R$. Results about time consumption are shown in Table 4 and Table 5, and the Pareto-optimal solutions and weak efficient solutions of BOSPNI problem are listed as well. It can be observed that the running time of BOSPNI-S is longer than that of SPNI-S, since there is an extra dual programming procedure SPNIR-D in BOSPNI-S. Time consumptions of procedures SPNI-S and SPNIR-D increase with network size in Table 4, thus running time of BOSPNI-S increases with network size.

Since there is $\sum_{i \in N \setminus \{s,t\}} r_i = 2,939$, the Pareto-frontier of BOSPNI problem on Chicago Sketch Road Network can be obtained by assigning different specific $\bar{R} \in [0, 3 \times 10^{3}]$, which is shown in Fig. 4 (b). Meanwhile, the weak efficient solutions of SPNI-S and SPNI-D are shown as well to illustrate the interdiction saturation properties of BOSPNI problem. It can be observed that weak efficient solutions from SPNI-S and SPNI-D may lead to resource waste, especially when $\bar{R} > R_s$ in which the objective $-L$ could not be improved as the objective $R$ increases. The saturation point of BOSPNI problem, and the weak saturation solutions from SPNI-S and SPNI-D are marked in Fig. 4 (b) separately. It can be observed that $(-L_s, R_s) = (-159, 876)$, $(-L_s^w, R_s^w) = (-159, 1, 557)$, $(-L_s^d, R_s^d) = (-159, 2, 939)$, thus $R_s < R_s^w < R_s^d$, which means algorithm SPNI-D tends to waste more resources than SPNI-S especially when $\bar{R} > R_s$. These experiment results confirm the theoretical analysis of saturation properties in shortest path network interdiction problems in section V. The saturation point illustrates that the upper bound of maximum shortest path length is 159, and the minimum resources cost to achieve this upper bound is 876. Therefore, the saturation point for BOSPNI problem reflects the saturation properties of a network, and the identification of Pareto-frontier can improve the quality of the decision making in BOSPNI problem.

To summarize, the proposed SPNI-S tends to save more resources compared with SPNI-D achieving the same shortest path length. This resources-saving feature is essential for interdictors with limited resources in real network

---

**TABLE 4. Computational results for BOSPNI problem.**

| Problem | SPNI-S | SPNIR-D | BOSPNI-S |
|---------|--------|---------|----------|
| No.     | $n_1$  | $N$     | $A$      | $T$ | $(-L_{wp}, R_{wp})$ | $T$ | $T$ | $(-L_p, R_p)$ |
| 1       | 20     | 400     | 2,659    | 13  | (-92,50) | 5   | 18  | (-92,50) |
| 2       | 25     | 625     | 4,232    | 53  | (-107,49) | 70  | 123 | (-107,49) |
| 3       | 30     | 900     | 6,162    | 87  | (-116,50) | 88  | 175 | (-116,49) |
| 4       | 35     | 1,225   | 8,451    | 94  | (-132,50) | 82  | 175 | (-132,47) |
| 5       | 40     | 1,600   | 11,081   | 186 | (-155,49) | 159 | 345 | (-155,48) |
| 6       | 45     | 2,025   | 14,120   | 532 | (-175,48) | 208 | 740 | (-175,45) |
| 7       | 50     | 2,500   | 17,544   | 947 | (-176,49) | 844 | 1,791 | (-176,48) |

**TABLE 5. Computational results for BOSPNI problem.**

| Chicago Sketch Road Network: $N = 933$, $A = 2,950$, $d_k = c_k$ | SPNI-S | SPNIR-D | BOSPNI-S |
|---------------------------------------------------------------------|--------|---------|----------|
| No.                    | $\bar{R}$ | $T$ | $(-L_{wp}, R_{wp})$ | $T$ | $T$ | $(-L_p, R_p)$ |
| 1                      | 30      | 7   | (-146,30) | 2   | 9   | (-146,28) |
| 2                      | 50      | 27  | (-152,50) | 18  | 45  | (-152,50) |
| 3                      | 70      | 49  | (-157,70) | 30  | 78  | (-157,69) |
| 4                      | 90      | 67  | (-161,90) | 19  | 86  | (-161,86) |
| 5                      | 110     | 197 | (-165,110)| 68  | 265 | (-165,108)|
| 6                      | 130     | 910 | (-168,130)| 41  | 951 | (-168,125)|
interdiction scenarios. Moreover, the proposed subgraph sequence algorithm based on SPNI-S can be used to solve BOSPNI problem for the Pareto-frontier effectively.

VII. CONCLUSION
This paper presents an innovative subgraph sequence algorithm for bi-objective shortest path network interdiction problem with respect to two objectives: maximization of shortest path length and minimization of resources cost. In this algorithm, Pareto-optimal solutions can be identified through a sequence of single-objective problems: (1) programming of shortest path length maximization solved by novel subgraph decomposition algorithms and (2) dual programming of resources cost minimization solved by a standard branch-and-bound algorithm. The performance of algorithms and interdiction saturation properties of BOSPNI problem are discussed based on theoretical analysis and computational experiments. The NP-hardness of shortest-path network interdiction with node interdiction has been theoretically proved in this paper. Time complexity of the interdiction subgraph generation procedure is $O(N^2)$, and it is proved that this procedure can reduce iterations of subgraph decomposition algorithm. Experimental results show that the proposed subgraph sequence algorithm could be used to solve BOSPNI problem for Pareto-frontier effectively. Meanwhile, a method to identify the saturation point of BOSPNI problem is proposed. This saturation point of BOSPNI can help us to make a trade-off between the resource investment and the interdiction performance of a network. Due to the NP-hardness of the problem, the solvable scale of problem is still limited. Hence, future research is needed to make more promotions in solving large-scale problems.

REFERENCES
[1] J. C. Smith and Y. Song, “A survey of network interdiction models and algorithms,” Eur. J. Oper. Res., vol. 283, no. 3, pp. 797–811, Jun. 2020.
[2] D. P. Morton, F. Pan, and K. J. Saeger, “Models for nuclear smuggling interdiction,” IIE Trans., vol. 39, no. 1, pp. 3–14, Jan. 2007.
[3] A. Delgadillo, J. M. Arroyo, and N. Alguacil, “Analysis of electric grid interdiction with line switching,” IEEE Trans. Power Syst., vol. 25, no. 2, pp. 633–641, May 2010.
[4] M. McCarter, K. Barker, J. Johansson, and J. E. Ramirez-Marquez, “A bi-objective formulation for robust defense strategies in multi-commodity networks,” Rel. Eng. Syst. Saf., vol. 176, pp. 154–161, Aug. 2018.
[5] M. P. Scaparra and R. L. Church, “An exact solution approach for the interdiction median problem with fortification,” Eur. J. Oper. Res., vol. 189, no. 1, pp. 76–92, Aug. 2008.
[6] Y. Xiao, P. Yang, S. Zhang, S. Zhou, W. Chang, and Y. Zhang, “Dynamic gaming case of the R-interdiction median problem with fortification and an MILP-based solution approach,” Sustainability, vol. 12, no. 2, p. 581, Jan. 2020.
[7] J. C. Smith, M. Prince, and J. Geunes, “Modern network interdiction problems and algorithms,” in Handbook of Combinatorial Optimization. New York, NY, USA: Springer, 2013, pp. 1949–1987.
[8] H. Bayrak and M. D. Bailey, “Shortest path network interdiction with asymmetric information,” Networks, vol. 52, no. 3, pp. 133–140, Oct. 2008.
[9] R. Wollmer, “Removing arcs from a network,” Operations Res., vol. 12, no. 6, pp. 934–940, Dec. 1964.
[10] R. K. Wood, “Deterministic network interdiction,” Math. Comput. Model., vol. 17, no. 2, pp. 1–18, Jan. 1993.
[11] K. T. Kennedy, R. F. Deckro, J. T. Moore, and K. M. Hopkins, “Nodal interdiction,” Math. Comput. Model., vol. 54, nos. 11–12, pp. 3116–3125, Dec. 2011.
[12] D. R. Fulkerson and G. C. Harding, “Maximizing the minimum source-sink path subject to a budget constraint,” Math. Program., vol. 13, no. 1, pp. 116–118, Dec. 1977.
[13] E. Israeli and R. K. Wood, “Shortest-path network interdiction,” Networks, vol. 40, no. 2, pp. 97–111, Sep. 2002, doi: 10.1002/net.10039.
[14] J. Naoum-Sawaya and B. Ghaddar, “Cutting plane approach for the maximum flow interdiction problem,” J. Oper. Res. Soc., vol. 68, no. 12, pp. 1533–1569, Dec. 2017.
[15] C. M. Rocco S and J. E. Ramirez-Marquez, “Deterministic network interdiction optimization via an evolutionary approach,” Rel. Eng. Syst. Saf., vol. 94, no. 2, pp. 568–576, Feb. 2009.
[16] J. Yates and K. Lakshmanan, “A constrained binary knapsack approximation for shortest path network interdiction,” Comput. Ind. Eng., vol. 61, no. 4, pp. 981–992, Nov. 2011.
[17] A. Washburn and K. Wood, “Two-person zero-sum games for network interdiction,” Oper. Res., vol. 43, no. 4, pp. 243–251, Apr. 1995.
[18] C. Zhang, J. E. Ramirez-Marquez, and J. Wang, “Critical infrastructure protection using secrecy–a discrete simultaneous game,” Eur. J. Oper. Res., vol. 242, no. 1, pp. 212–221, Apr. 2015.
[19] A. Sanjab, W. Saad, and T. Basar, “Prospect theory for enhanced cyber-physical security of drone delivery systems: A network interdiction game,” in Proc. IEEE Int. Conf. Commun. (ICC), May 2017, pp. 1–6.
K. Xiao et al.: BOSPNI Problem: Subgraph Algorithm and Saturation Property

[20] B. J. Lunday and H. D. Sherali, “A dynamic network interdiction problem,” Informatica, vol. 21, no. 4, pp. 553–574, Jan. 2010.

[21] E. Gutin, D. Kuhn, and W. Wiesemann, “Interdiction games on Markovian PERT networks,” Manage. Sci., vol. 61, no. 5, pp. 999–1017, May 2015.

[22] J. Guo, Z. Xie, and Q. Li, “Stackelberg game model of railway freight pricing based on option theory,” Discrete Dyn. Nature Soc., vol. 2020, pp. 1–11, Jun. 2020.

[23] X. Wei, C. Zhu, K. Xiao, Q. Yin, and Y. Zha, “Shortest path network interdiction with goal threshold,” IEEE Access, vol. 6, pp. 29332–29343, 2018.

[24] X. Fu and E. Modiano, “Network interdiction using adversarial traffic flows,” in Proc. IEEE Conf. Comput. Commun. (IEEE INFOCOM), Apr. 2019, pp. 1765–1773.

[25] K. Xu, K. Xiao, Q. Yin, Y. Zha, and C. Zhu, “Bridging the gap between observation and decision making: Goal recognition and flexible resource allocation in dynamic network interdiction,” in Proc. Twenty-Sixth Int. Joint Conf. Artif. Intell., Aug. 2017, pp. 4477–4483.

[26] D. Aksen, S. Şengül Akca, and N. Aras, “A bilevel partial interdiction problem with capacitated facilities and demand outsourcing,” Comput. Oper. Res., vol. 41, pp. 346–358, Jan. 2014.

[27] K. Xiao, C. Zhu, J. Xie, Y. Zhou, X. Zhu, and W. Zhang, “Dynamic defense strategy against stealth malware propagation in cyber-physical systems,” in Proc. IEEE Conf. Comput. Commun. (IEEE INFOCOM), Apr. 2018, pp. 1790–1798.

[28] M. Mahmoodjanloo, S. P. Parvasi, and R. Ramezanian, “A tri-level covering fortification model for facility protection against disturbance in r-interdiction median problem,” Comput. Ind. Eng., vol. 102, pp. 219–232, Dec. 2016.

[29] L. T. K. Hien, M. Sim, and H. Xu, “Mitigating interdiction risk with fortification,” Oper. Res., pp. 348–362, Feb. 2020.

[30] C. M. Rocco S., J. Emmanuel Ramirez-Marquez, and D. E. Salazar A., “Bi and tri-objective optimization in the deterministic network interdiction problem,” Rel. Eng. Syst. Saf., vol. 95, no. 8, pp. 887–896, Aug. 2010.

[31] Y. Chen, C. Guo, and S. Yu, “Bi-objective optimization models for network interdiction,” RAIRO-Oper. Res., vol. 53, no. 2, pp. 461–472, Apr. 2019.

[32] C. Zhang and J. E. Ramirez-Marquez, “Protecting critical infrastructures against intentional attacks: A two-stage game with incomplete information,” IEEE Trans., vol. 45, no. 3, pp. 244–258, Mar. 2013.

[33] J. O. Royset and R. K. Wood, “Solving the bi-objective maximum-flow network-interdiction problem,” Inform. J. Comput., vol. 19, no. 2, pp. 175–184, May 2007.

[34] T. Holzmann and J. C. Smith, “Shortest path interdiction problem with arc improvement recourse: A multiobjective approach,” Nav. Res. Logistics, vol. 66, no. 3, pp. 230–252, Apr. 2019.

[35] C. M. Rocco S. and J. E. Ramirez-Marquez, “A bi-objective approach for shortest-path network interdiction,” Comput. Ind. Eng., vol. 59, no. 2, pp. 232–240, Sep. 2010.

[36] K. Xiao, C. Zhu, W. Zhang, X. Wei, and S. Hu, “Stackelberg network interdiction game: Nodal model and algorithm,” in Proc. 5th Int. Conf. Game Theory Netw. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers Inc., Nov. 2014, pp. 1–5.

[37] M. R. Garey and D. S. Johnson, Computers and Intractability, San Francisco, CA, USA: Freeman, 1979, vol. 174.

[38] R. K. Ahuja, T. L. Magnanti, J. B. Orlin, and K. Weihe, “Network flows: Theory, algorithms and applications,” ZOR-Methods Models Oper. Res., vol. 41, no. 3, pp. 252–254, 1995.

[39] B. Golden, “A problem in network interdiction,” Nav. Res. Logistics Quart., vol. 25, no. 4, pp. 711–713, Dec. 1978.

[40] T. Brinkhoff, “A framework for generating network-based moving objects,” GeoInformatica, vol. 6, no. 2, pp. 153–180, 2002.

[41] H. Bar-Gera. Transportation Network Test Problems. Accessed: Aug. 10, 2020. [Online]. Available: http://www.bgu.ac.il/~ bargera/tmp/

[42] J. Löfberg, “Yalmip: A toolbox for modeling and optimization in MATLAB,” in Proc. CDC Conf., Taipei, Taiwan, 2004, pp. 284–289. [Online]. Available: http://users.isy.liu.se/johan/yalmip

[43] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, Introduction to Algorithms, vol. 2. Cambridge, MA, USA: MIT Press, 2001.

KAIMING XIAO received the B.S. degree in mechanical engineering and automation from Tsinghua University, Beijing, China, in 2013, and the M.S. degree in management science and engineering from the National University of Defense Technology (NUDT), Changsha, China, in 2015, where he is currently pursuing the Ph.D. degree.

His research interests include network optimization, network interdiction, and security game.

CHENG ZHU received the Ph.D. degree in management science and engineering from the National University of Defense Technology (NUDT), China, in 2005.

He is currently a Professor with the Science and Technology on Information Systems Engineering Laboratory, NUDT. His current research interests include computer networks and data mining.

WEIMING ZHANG received the Ph.D. degree in management science and engineering from the National University of Defense Technology (NUDT), China, in 2001.

He is currently the Chief of the Science and Technology on Information Systems Engineering Laboratory, NUDT. His current research interest includes command and control organizations.

XIANGYU WEI received the Ph.D. degree in management science and engineering from the National University of Defense Technology (NUDT), Changsha, China, in 2018.

His research interests include network security games and human behavior modeling.

* * *