



Most vital vertices for the shortest s – t path problem: complexity and Branch-and-Cut algorithm

Youcef Magnouche¹ · Sébastien Martin¹

Received: 12 December 2018 / Accepted: 23 December 2019 / Published online: 3 January 2020
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

Abstract

In this paper we study the most vital vertices for the shortest s – t path problem. This problem consists, given a digraph $D = (V \cup \{s, t\}, A)$ and a threshold $d \in \mathbb{N}$, in finding the minimum number of nodes to delete from D in such a way that there does not exist a s – t path of length less or equal than d . We prove the NP-hardness of this problem and propose an integer linear program using an exponential number of inequalities. We investigate the facial structure of the associated polytope. We prove that this polytope is integer and we derive an efficient Branch-and-Cut algorithm. Finally, we present and discuss some computational results.

Keywords Blocker · Shortest path · Complexity · Integer linear program · Integral polytope · Branch-and-Cut

1 Introduction

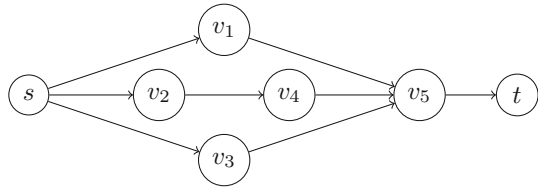
Shortest path problem between two specific vertices s and t is a well-known polynomial problem. This problem is called shortest s – t path. In many applications, the identification of the most critical vertices is a necessity to ensure security. Indeed the survivability of networks (as communication or transportation networks) is a priority to prevent failure. One of the most concerning problems is to ensure a connection (path) between s and t satisfying some special features even if the network is partially destroyed by the deletion of some nodes. In theoretical computer science, these kind of problems are known as *Network Interdiction Problems* or *Network Blocker Problems* [1–3].

✉ Youcef Magnouche
youcef.magnouche@huawei.com

Sébastien Martin
sebastien.martin@huawei.com

¹ Huawei Technologies France, 19 Quai du Point du Jour, 92100 Boulogne-Billancourt, France

Fig. 1 Example MVBSP and MVVSP



In Khachiyan et al. [2] define the Minimum Vertex Blocker to Shortest Path problem (MVBSP) as follows: let $D = (V \cup \{s, t\}, A)$ be a digraph, where $D \cup \{s, t\}$ is a set of nodes with two distinguished nodes s and t , and A is a set of arcs. Given a nonnegative length $l(a)$ associated to each arc $a \in A$ and a threshold $k \in \mathbb{Z}^+$, a vertex blocker is a set of vertices whose removal increases the s - t -distance to at least k . The objective is to find the smallest vertex blocker i.e. $\min\{|U| \mid d_{D[V \setminus U]}(s, t) \geq k, U \subseteq V \setminus \{s, t\}\}$, where $d_D(s, t)$ is the s - t -distance in D (the length of the shortest path between s and t in D). They prove that it is NP -hard to approximate the size of the smallest vertex blocker within a factor smaller than 1.36, even in bipartite graphs. From an integer linear programming point of view, some works are done to solve interdiction or blocker problems. For instance, in [4] the authors solve the maximum clique interdiction problem. To the best of our knowledge, the integer linear programming was not investigated for the blocker/interdiction or vital nodes associated with the shortest path problem.

In this article, we introduce the most vital vertices for the shortest s - t path problem (MVVSP) defined as follows: given a digraph $D = (V \cup \{s, t\}, A)$ with a nonnegative length $l(a)$ associated with every arc $a \in A$ and a threshold $d > 0$, a set of vertices is *vital* if its removal ensures that it does not exist a s - t path of length less or equal than d . The aim is to find the smallest vital set. We suppose that if $(st) \in A$, then $l(st) > d$. Otherwise, there is no solution for the problem. A node is called vital, if it belongs to a vital set. The difference between these two problems is the necessity for the MVBSP to ensure the existence of a s - t path after removing the blocker set of vertices instead of MVVSP which consists in breaking all shortest s - t paths with length less or equal than d by removing the minimum number of nodes. We assume that there is no direct arc from s to t . Otherwise the problem is trivial. For instance, in Fig. 1 the best solution for the MVBSP when $k = 4$ is $\{v_1, v_3\}$, while the best solution for the MVVSP when $d = 3$ is $\{v_5\}$. In the survivability of networks the most critical vertex is $\{v_5\}$ whereas $\{v_1, v_3\}$ are not vital.

The rest of the paper is organized as follows: In Sect. 2, we discuss some related works. Section 3 presents graph definitions and the NP -hardness of the MVVSP. In Sect. 4, we introduce the integer linear program that models the problem of MVVSP. In Sect. 5, a polyhedral analysis is done to prove the integrality of the associated polytope. Section 6 deals with experimental results. The conclusion and some future work are given in Sect. 7.

2 Related works

In [2], the article deals with the minimum vertex blocker to shortest path problem. The authors prove that it is NP -hard to approximate the size of the smallest vertex

(edge resp.) blocker within a factor 1.36 even in bipartite graphs. Moreover, they show that it is NP -hard to approximate the most vital vertices (edge resp.) problem within a factor 2 even in bipartite graphs. In [5], authors study the problem of critical edges for the assignment problem which consists in finding a subset of k edges in a bipartite graph such that the partial bipartite graph obtained by removing these edges increases the value of the minimum cost assignment. They also study the dual version which consists in removing a subset of edges of minimum size so that in the resulting partial bipartite subgraph, the cost of the minimum assignment is at least a given value. Using the polynomial reduction given by Hoffman and Markowitz [6] from the shortest path to the assignment problem, Bazgan et al. [5] extends this reduction to the k most vital arcs shortest path problem in order to prove that the k most vital edges assignment problem is NP -hard to approximate within a factor $2 - \epsilon$, $\forall \epsilon > 0$ and the minimum edge blocker assignment problem is NP -hard to approximate within a factor of 1.36. They also propose exact algorithms based on graphs to solve these problems.

In [7], authors work on the shortest path most vital edges which aims to find a subset of at most k edges so that the value of the shortest path in the resulting partial graph increases by at least b . They prove that this problem is polynomial when $b = 1$ and it is NP -hard for $b \geq 2$ with a reduction from the vertex cover on three-partite graphs. They also study the problem from a parameterized complexity point of view.

3 Graph properties and complexity

In this section we propose some graph properties to contract the graph or to remove some vertices without changing the optimal solution of the MVVSP. Firstly, we give some definitions. We consider digraphs. We denote a digraph by $D = (V \cup \{s, t\}, A)$, where $V \cup \{s, t\}$ is the set of vertices and A the set of arcs. We let $n = |V|$ and $m = |A|$. Let $N^+(v)$ (resp. $N^-(v)$) be the set of all nodes in G adjacent to v such that v is the first (resp. second) node of the arcs. Given a subset of vertices $W \subseteq V$, if $x \in \mathbb{R}^V$ then we let $x(W) = \sum_{v \in W} x_v$. A *path* P is an ordered set of p distinct vertices v_1, v_2, \dots, v_p such that for all $i \in \{1, \dots, p-1\}$, (v_i, v_{i+1}) is an arc. Vertices v_2, \dots, v_{p-1} are called *the internal vertices* of P . We denote by svt the path given by the arcs (s, v) and the arcs (v, t) .

Paths with three nodes If there is a node $v \in V$ such that svt exists with $l(svt) \leq d$ then v is obviously a vital node and we must delete it. Hence we can focus on the subgraph $D \setminus \{v\}$. Figure 2 shows an example of the graph reduction. Indeed, the vertex v_1 is a vital node in the graph, hence we can delete it and reduce the size of the graph.

Figure 2 shows another example of the graph reduction. Indeed, the vertices v_2 and v_3 cannot belong to any s - t path. Thus we can delete them and reduce the size of the graph.

Connectivity The digraph D is connected. Indeed, if s and t are in two different connected components, then the smallest set of vital nodes is empty. Furthermore, if s and t are in the same connected component, it follows that no vertex of the other connected component belongs to the smallest set of vital nodes. Thus we can reduce the graph by deleting them. Clearly, in Fig. 2, the vertices v_2 and v_3 cannot be vital then we can delete them.

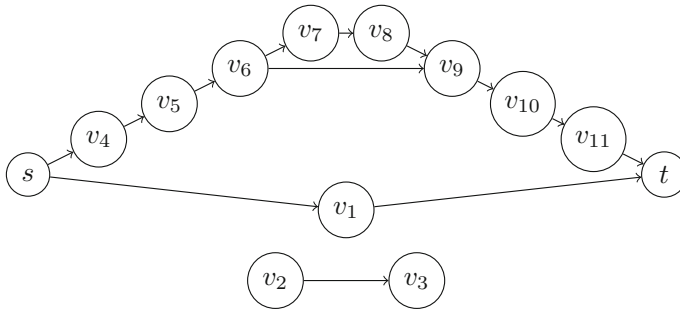
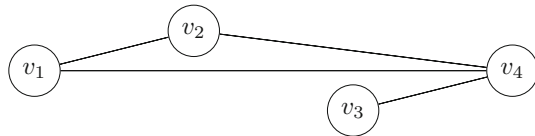


Fig. 2 Example of graph where all lengths are 1

Fig. 3 Example of graph G



Node in a path Each vertex $v \in V$ belongs to a path between s and t of length less than d . Otherwise, v cannot belong to a smallest set of vital nodes, then we can delete it. For each $v \in V$ if the length of the shortest path between s and v plus the length of the shortest path between v and t is greater than or equal to d then we can remove the vertex v . In Fig. 2, if $d = 8$ then v_7 and v_8 do not belong to a path of length less than 8.

Minimality of path A path P is said to be *minimal* if there exists no $s-t$ path P' such that $P' \subset P$. For instance, in Fig. 2, if $d = 10$ then the path $P_1 = \{s, v_4, v_5, v_6, v_7, v_8, v_9, v_{10}, v_{11}, t\}$ contains the path $P_2 = \{s, v_4, v_5, v_6, v_9, v_{10}, v_{11}, t\}$, i.e. $P_2 \subset P_1$, thus P_1 is not minimal.

Theorem 1 *The MVVSP with only unitary length is NP-hard.*

Proof This proof is inspired from the proof given in [2] for the NP-hardness of the MVVSP. To prove the theorem, we shall use a reduction from the minimum vertex cover problem. Let $G = (V, E)$ be a graph where $V = \{v_1, \dots, v_n\}$. An example of such graph is given in Fig. 3. A minimum vertex cover in G is a set of vertices of minimum cardinality such that each edge of E is incident to at least one vertex of the set. A minimum vertex cover in G given by the Fig. 3 is $\{v_1, v_4\}$.

Let $D = (V' \cup \{s, t\}, A)$ be a graph obtained from G as follows:

- $V' = V \cup \{s, t\} \cup V^1 \cup \dots \cup V^n$ where $V^i = \{v_{1,1}^i, \dots, v_{1,4}^i, \dots, v_{1,n}^i, \dots, v_{4,n}^i\}$ for $i = 1, \dots, n$
- Add n paths $P^i = s v_{1,1}^i v_{1,2}^i v_{1,3}^i v_{1,4}^i v_{2,1}^i v_{2,2}^i v_{2,3}^i v_{2,4}^i \dots v_{n,1}^i v_{n,2}^i v_{n,3}^i v_{n,4}^i t$ for all $i \in \{1, \dots, n\}$
- For each $i = 1, \dots, n$, there are arcs $(v_i, v_{i,1}^j)$ and $(v_{i,1}^j, v_i)$, where $j \in \{1, \dots, n\}$
- For all $uv \in E$ we replace it by the arcs (u, v) and (v, u) . We denote these arcs by $A[E]$.

So we have, $E' = A[E] \cup C^1 \cup \dots \cup C^n \cup A[P^1] \cup \dots \cup A[P^n]$ where $C^i = \{(v_i, v_{i,1}^1), \dots, (v_i, v_{i,1}^n)\} \cup \{(v_{i,1}^1, v_i), \dots, (v_{i,1}^n, v_i)\}$ and $A[P^i]$ is the set of arcs from

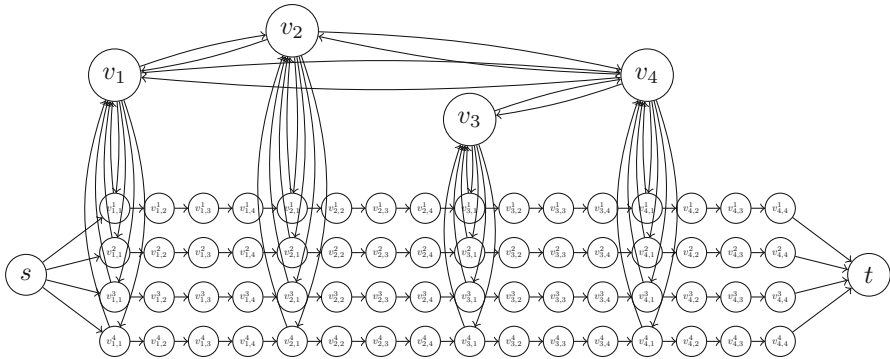


Fig. 4 Graph D obtained from G

the path P^i , for $i = 1, \dots, n$. Finally, we assign a length 1 for all arcs. Figure 4 represents such reduction for the graph given in Fig. 3.

We will show that the minimum vertex cover in G is equivalent to the most vital set for the shortest s - t path problem in D where $d = 4n$.

Note that, the length of paths P^i , $i \in \{1, \dots, n\}$, are equal to $4n + 1$, while for all other s - t paths the length is less or equal to $4n$. Indeed, each s - t path using an arc $(u, w) \in A[E]$ replaces at least four arcs $v_{i,1}^u v_{i,2}^u v_{i,3}^u v_{i,4}^u v_{i+1,1}^u$ from P^i , $u \in V$ and $i \in \{1, \dots, n\}$, by only 3 arcs $v_{i,1}^u v_u v_w v_{j,1}^u$, $v \in V$, $j \in \{i + 1, \dots, n\}$.

First, we observe that every vertex cover of G is vital set of D . Second, suppose that S is a vital set of D but not a vertex cover of G . It follows that $|S| < n$ and there exists at least one arc $v_i v_j$ where $i < j$ and $v_i, v_j \in V \setminus S$. Then there exists an integer $a \in \{0, \dots, n\}$ and a s - t path P of length strictly less than d , where $P = s v_{1,1}^a v_{1,2}^a v_{1,3}^a v_{1,4}^a \dots v_{i,1}^a v_i v_j v_{j,1}^a v_{j,2}^a v_{j,3}^a v_{j,4}^a \dots v_{n,1}^a v_{n,2}^a v_{n,3}^a v_{n,4}^a t$. This contradicts the fact that S is a vital set of D . Thus the minimum vertex cover of G is equivalent to the most vital vertices for the shortest s - t path problem in D . \square

4 Integer linear program

In this section, we propose an integer linear program to solve the MVVSP. Let $x \in \{0, 1\}^{|V|}$ defined by

$$x_v = \begin{cases} 1 & \text{if } v \text{ is a vital vertex,} \\ 0 & \text{otherwise,} \end{cases} \quad \forall v \in V.$$

The MVVSP is equivalent to the following (P') :

$$\begin{aligned} & \min \sum_{v \in V} x_v \\ (P') \quad & \sum_{v \in P} x_v \geq 1, \quad \text{for all } P \in \mathcal{P}, \end{aligned} \tag{1}$$

$$x_v \in \{0, 1\}, \quad \text{for all } v \in V. \tag{2}$$

where \mathcal{P} is the set of all s - t paths of length less or equal to d . The inequalities (1) ensure that we interdict at least one node in all paths of length less or equal to d . Remark that, the length of s - t paths does not appear in the model. Indeed, the length is in the definition of \mathcal{P} .

The inequalities (1) are in exponential number. In order to solve (P') using a Branch-and-cut approach, one needs an efficient algorithm for the separating inequalities (1).

Separation algorithm The separation problem for inequalities (1) consists, given a solution $x^* \in \mathbb{R}^{|V|}$, in determining whether x^* satisfies inequalities (1), and if not in finding an inequality violated by x^* . The separation problem consists in finding a s - t path P such that $l(P) \leq d$ and minimizing the cost function $\sum_{v \in P} x_v^*$. We denote by $A(P)$ the sequence of arcs associated with the path P . In the following, we prove that the separation problem is NP-hard using a polynomial reduction from the shortest weight-constrained s - t path problem [8], known to be NP-hard. Given a directed graph $D = (V, A)$, a cost function $c : A \Rightarrow \mathbb{Z}^+$, a weight function $w : A \Rightarrow \mathbb{Z}^+$, a threshold $W \in \mathbb{N}$ and two terminal vertices $s, t \in V$, the shortest weight-constrained s - t path problem consists in finding a s - t path P such that $\sum_{a \in A(P)} w(a) \leq W$ and the cost ($\sum_{a \in A(P)} c(a)$) is minimum. Let $D' = (V', A')$ be the graph obtained from D by adding, for each arc $a = (u, v) \in A$, a vertex z^a to V' with a cost $c(a)$ and by replacing arc a by two arcs (u, z^a) and (z^a, v) , each one with a weight $\frac{w(a)}{2}$. The costs of u and v are 0. Clearly, the optimal solution given by solving the separation problem on D' can be transformed to an optimal solution of the shortest weight-constrained s - t path problem on D , in a polynomial time. We can deduce that the separation problem is NP-Hard.

However, if we consider a separation with an integer vector $x^* \in \mathbb{N}^{|V|}$ then the separation becomes polynomial. This case consists in finding a s - t path P such that $l(P) \leq d$ and $\sum_{v \in P} x_v^* < 1$, and since the vector x^* is integer, then we search a shortest s - t path on the graph without any vertex $v \in V$ satisfying $x_v^* = 1$.

5 Polyhedral analysis

Let $P(D, d) = \text{conv}(x \in \{0, 1\}^{|V|} | x \text{ satisfies (1)})$ be the polytope of vital vertices sets for a length d in graph $D = (V, A)$.

Given an inequality $ax \leq b$, where $a \in \mathbb{R}^V$, the support graph of $ax \leq b$ is the subgraph induced by the vertices corresponding to variables having a non-zero coefficient in the inequality.

Theorem 2 $P(D, d)$ is full-dimensional.

Proof We need to exhibit $|V| + 1$ subsets of vertices such that their incidence vectors are affinely independent. Let $S_0 = V$. Clearly S_0 is a vital vertices set of D . For each $v \in V$, let $S_v = V \setminus \{v\}$. Indeed, in Sect. 3, we supposed that D does not contain any s - t

path of length 2. It follows that S_v is a vital vertices set. This constitutes a set of $n + 1$ vital sets of D . Moreover, their incidence vectors are clearly affinely independent. \square

For a path P between s and t , let a be the vertex of P adjacent to s and b the vertex in P adjacent to t .

Theorem 3 *For a path $P \in \mathcal{P}$, inequality (1) defines a facet of $P(D, d)$ if and only if P is minimal.*

Proof (\Leftarrow) If there exists a path $P' \in \mathcal{P}$ such that $P' \subset P$, then inequality (1) associated with P can be obtained by summing inequality (1) associated with P' and inequalities $x_v \geq 0$ for all $v \in P \setminus P'$.

(\Rightarrow) We need to exhibit $|V|$ subsets of vital vertices such that their incidence vectors are affinely independent. For $v \in P$, let $S_0^v = (V \setminus P) \cup \{v\}$, S_0^v is a vital vertices set of D since P is minimal. For each $w \in V \setminus P$, let $S'_w = S_0^a \setminus \{w\}$ if w is adjacent to t and $S'_w = S_0^b \setminus \{w\}$ otherwise. Since D does not contain any s - t path of length 2 (see, Sect. 3), it follows that S'_w is a vital vertices set. This constitutes a set of n vital vertices sets of D . Moreover, their incidence vectors are clearly affinely independent. \square

Theorem 4 *The polytope given by (1) and trivial inequalities is integer.*

Proof We suppose that (1) and trivial inequalities are not enough to characterize completely the MVVSP polytope and we prove that there is a contradiction. The MVVSP polytope is not integer means that there exists a fractional extreme point $x^* \in [0, 1]^V$ that is the unique solution of the following linear system of equalities \mathcal{A} :

$$\sum_{v \in P} x_v^* = 1 \quad \forall P \in \mathcal{P}', \tag{3}$$

$$x_v^* = 1 \quad \forall v \in V_1, \tag{4}$$

$$x_v^* = 0 \quad \forall v \in V_2. \tag{5}$$

Such that $\mathcal{P}' \subseteq \mathcal{P}$, $V_1, V_2 \subseteq V$ and $|\mathcal{P}'| + |V_1| + |V_2| = |V|$. We suppose that D is minimum (i.e., for any graph $(V' \cup T', E')$ with $|V'| < |V|$ the polytope given by inequalities (1) and trivial inequalities is integer).

By the following propositions we prove that x^* cannot exist.

For $v \in V$, let the v -contracted graph D'_v be the graph obtained from D by adding an arc (ab) , for each path avb of length $l(av) + l(vb)$ and then deleting vertex v . If (ab) already exists, we do not add a double arc. We update the length of (ab) by $\min(l(ab), l(av) + l(vb))$ and we delete v . \square

Proposition 1 *If $P \in \mathcal{P}$ is a s - t path in D containing path avb then $P \setminus \{v\}$ is a path in D'_v such that $l(P \setminus \{v\}) \leq l(P)$.*

Proof Trivial. \square

Proposition 2 *For all s - t path P' in D'_v containing arc (ab) , there exists a s - t path P in D containing v such that $P \setminus \{v\} = P'$*

Proof Trivial. □

Proposition 3 $x_v^* > 0$ for all $v \in V$ (i.e., $V_2 = \emptyset$).

Proof Let us assume the contrary. Let v be a vertex of V such that $x_v^* = 0$. Let D'_v be the v -contracted graph of D . Let \bar{x} be the restriction of x^* on $V \setminus \{v\}$. From Propositions 1 and 2, we have the following claims. □

Claim 1 $\bar{x} \in P(D'_v, d)$.

Proof Suppose that $\bar{x} \notin P(D'_v, d)$. It follows that there exists an inequality (1) violated by \bar{x} . Let P' be the s - t path in D'_v associated with this inequality. We distinguish three cases

- P' does not contain arc (ab) : it follows that P' is a s - t path in D and x^* violated the inequality (1) associated with P' .
- P' contains arc (ab) and (ab) does not exists in D : let P'' be the s - t path in D obtained from P' by replacing the arc (ab) by the path avb . Since $x_v^* = 0$, it follows that x^* violates inequality (1) associated with P'' .
- P' contains arc (ab) and (ab) exists in D : P' is also a s - t path in D and x^* violates its associated inequality (1).

This contradicts the fact that $x^* \in P(D, d)$.

Let \mathcal{A}' be the linear system obtained from \mathcal{A} by deleting equality $x_v^* = 0$ and variable x_v^* from all equalities. □

Claim 2 The support graph of each non-trivial equality of \mathcal{A}' represents a s - t path in D'_v .

Proof By definition, the support graph of each non-trivial equality of \mathcal{A} represents a s - t path in D . Moreover, from Proposition 1, for each path P of those support graphs, $P \setminus \{v\}$ is a path of D'_v of a maximum length d which ends the proof.

Corollary 1 From Claims (1) and (2), it follows that \bar{x} is an extreme point of $P(D'_v, d)$.

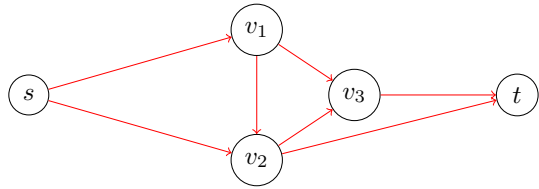
Corollary 1 contradicts the fact that D is minimum. It follows that $x_v^* > 0$ for all $v \in V$. □

Proposition 4 There exists at least one node $a \in N^+(s)$ and $b \in N^-(t)$ such that x_a^* and x_b^* are fractional.

Proof From Proposition 3, each variable $x_v^* > 0$ for all $v \in V$. For instance, suppose that all variables associated with the vertices of $N^+(s)$ are equal to 1. Since x^* is fractional, there must exists a vertex $v \in V \setminus N^+(s)$ such that x_v^* is fractional. We distinguish two cases

- x_v^* does not appear in any equality (3), let $x' \in [0, 1]^V$ be the solution such that $x'_u = x_v^*$ for all $u \in V \setminus \{v\}$ and $x'_v = 1$. Clearly, x' is another feasible solution for \mathcal{A}' and this contradicts the fact that x^* is a unique solution of \mathcal{A}' and x^* is an extreme point of $P(D, d)$.

Fig. 5 Example



- x_v^* appears in at least one equality (3). Since the right hand side of this equality is 1 and all variables associated with the vertices of $N^+(s)$ are equal to 1, it follows that $x_v^* = 0$. Contradiction with the fact that x_v^* is fractional.

Therefore, the proposition holds. □

Proposition 5 For each inequality (3), the associated $s-t$ path P contains exactly one vertex from $N^+(s)$ and one vertex from $N^-(t)$.

Proof Suppose that P contains at least two vertices of $N^+(s)$ (the proof is similar for $N^-(t)$). It follows that there exists another $s-t$ path P' included in P intersecting once $N^+(s)$. From Proposition 3, inequality (1) associated with P cannot be tight and then it cannot belongs to \mathcal{A} . □

Let $\bar{x} \in [0, 1]^V$ be a solution obtained from x^* by the following operations

- $\bar{x}_v = x_v^*$ for all $v \in V \setminus N^+(s) \cup N^-(t)$.
- $\bar{x}_v = x_v^*$ for all $v \in N^+(s) \cup N^-(t)$ such that x_v^* is integer.
- $\bar{x}_v = x_v^* + \epsilon$ for all $v \in N^+(s)$ such that x_v^* is fractional.
- $\bar{x}_v = x_v^* - \epsilon$ for all $v \in N^-(t)$ such that x_v^* is fractional.

Clearly each path of \mathcal{P} contains one vertex of $N^+(s)$ and one vertex of $N^-(t)$. There exists a small $\epsilon > 0$ such that \bar{x} satisfies the linear system \mathcal{A} . Contradiction with the fact that x^* is an extreme point of $P(D, d)$. □

Proposition 6 The polytope given by inequalities I subset of inequalities (1) and trivial inequalities is not always integer.

Proof An example is sufficient to show the proposition.

Let D be the graph in Fig. 5 and $d = 3$. The set of $s-t$ paths \mathcal{P} in D contains the following paths:

- $p_1 : s - v_1 - v_3 - t$
- $p_2 : s - v_2 - t$
- $p_3 : s - v_2 - v_3 - t$
- $p_4 : s - v_1 - v_2 - t$

Let $\mathcal{P}' = \mathcal{P} \setminus \{p_2\}$ be a subset of $s-t$ paths in \mathcal{P} . The inequalities (1) associated with \mathcal{P}' are the following :

$$\begin{aligned}
 x_{v_1} + x_{v_3} &\geq 1, \\
 x_{v_2} + x_{v_3} &\geq 1, \\
 x_{v_1} + x_{v_2} &\geq 1, \\
 0 &\leq x_{v_1} \leq 1, \\
 0 &\leq x_{v_2} \leq 1, \\
 0 &\leq x_{v_3} \leq 1.
 \end{aligned}$$

Clearly, $x_{v_1} = 0.5$, $x_{v_2} = 0.5$, $x_{v_3} = 0.5$ is a feasible solution of the above linear system. Moreover, it represents an extreme point of $P(D, d)$ (there are 3 variables and 3 linear inequalities satisfied to equality, that are linearly independents). Moreover, the inequality associated to p_2 permits to cut the fractional extreme point. \square

6 Experimental results

We developed a Branch-and-Cut algorithm to solve the MVVSP. As mentioned in the Sect. 4, the integer linear program has an exponential number of inequalities (1). In our Branch-and-Cut, we separate only the integer solutions by separating inequalities (1) using Dijkstra algorithm for solving the shortest $s-t$ path problem.

We now describe the framework of our algorithm. To start the optimization, we consider the linear program only with the trivial inequalities. The actually optimal solution $x^* \in \mathbb{R}^V$ of this relaxation of the MVVSP is feasible for the problem if x^* is an integer vector that satisfies all inequalities (1). Usually, the solution x^* is either fractional or not feasible for the MVVSP. In each iteration of the Branch-and-Cut algorithm, if x^* is fractional, one has to branch on a fractional variable x_i by generating two child nodes, one with an additional constraint $x_i = \lfloor x_i \rfloor$ and the other one with $x_i = \lceil x_i \rceil$. However, when x^* is integer but not feasible, it is necessary to generate further inequalities (1) violated by the current solution x^* . For this, one has to solve the so-called separation problem. This consists, given a class of inequalities, in deciding whether the current solution x^* satisfies all the inequalities of this class, and if not, in finding an inequality that is violated by x^* . An algorithm solving this problem is called a separation algorithm. The Branch-and-Cut algorithm uses only inequalities (1). In our implementation, the solver Cplex is used to handle the branching tree and solving the linear programs.

Computational results are obtained using Cplex 12.6 and Lemon 1.3.1. Two sets of instances were used, the DIMACS and random instances. The DIMACS instances are composed by 50 instances originally proposed for Maximum Clique, Graph Coloring, and Satisfiability in the second DIMACS challenge [9]. However, these instances are designed to be a challenge for combinatorial problems. The required CPU time is measured in seconds. We limit to 3600 seconds the algorithm running time for each instance, by using at most 8 GB of RAM and a processor Intel Core i5-3340M CPU of $2.70 \text{ GHz} \times 4$.

The integer linear program is tested on the following proposed benchmark of instances. The graph density is equal to 2, 10, 25, 50 and 75 percent. For each instance,

Table 1 Average of CPU time

$ V $	Density	d	Nodes	CPU	#cuts	Size blocker
16,000	2	3	1	9.61	7	7
16,000	2	4	1	1136.42	857	334
16,000	2	5	1	1267.29	871	335
19,000	2	3	1	16.35	7	7
19,000	2	4	1	2482.30	1110	360
6000	10	3	1	64.24	77	77
8000	10	3	1	138.77	83	83
8000	10	4	1	Time limit	2759	780
1500	25	3	1	15.19	87	87
1500	25	4	1	175.03	1293	365
2000	25	3	1	44.44	130	130
2000	25	4	1	490.33	1816	515
2500	25	3	1	50.73	148	148
2500	25	4	1	955.44	2333	610
1500	50	3	1	110.14	376	376
1500	50	4	1	297.69	1672	753
2000	50	3	1	161.38	468	468
2000	50	4	1	521.51	2406	967
2500	50	3	1	351.03	648	648
2500	50	4	1	1765.47	2722	1258
1500	75	3	1	169.87	844	844
1500	75	4	1	218.69	1812	1111
2000	75	3	1	444.06	1144	1144
2000	75	4	1	817.83	2470	1496
2500	75	3	1	1451.76	1389	1389
2500	75	4	1	1567.47	2622	1826

we consider all different values of d , between $sp + 1$ and $disc$ where sp is equal to the length of the shortest $s-t$ path and $disc$ is the minimum value for d for which the smallest vital set disconnects s and t . The next tables provide the following informations:

- $|V|$, the number of vertices;
- density, the density of D ;
- d , the bound of remaining shortest $s-t$ paths (all shortest $s-t$ path must be strictly greater than d);
- Nodes, the number of nodes in the branching tree;
- CPU, Computational time (limited to 1 h);
- #cuts, the number of inequalities (1) added in the Branch-and-Cut algorithm;
- Size Blocker, the size of the blocker for this instance.

Table 2 Computational results from DIMACS instances

Name	$ V $	Density	d	Nodes	CPU	#cuts	Size blocker
anna	138	5	3	1	0.001	1	1
david	87	10	3	1	0.001	1	1
david	87	10	10	1	0.001	3	2
huck	74	11	1	1	0.000	0	0
inithx.i.1	864	2	1	1	0.004	0	0
inithx.i.2	645	3	3	1	0.039	23	23
inithx.i.3	621	3	3	1	0.010	4	4
inithx.i.3	621	3	64	10	0.128	75	24
jean	80	8	5	1	0.000	1	1
le450_5a	450	2	3	1	0.006	4	4
le450_5a	450	2	47	1	0.042	55	24
le450_5b	450	2	3	1	0.005	4	4
le450_5b	450	2	47	1	0.055	74	29
le450_5c	450	4	3	1	0.005	3	3
le450_5c	450	4	47	1	0.072	66	35
le450_5d	450	4	3	1	0.007	5	5
le450_5d	450	4	47	1	0.101	100	42
le450_15a	450	4	3	1	0.002	1	1
le450_15a	450	4	47	1	0.093	68	33
le450_15b	450	4	3	1	0.005	2	2
le450_15b	450	4	47	1	0.042	40	22
le450_15c	450	8	3	1	0.043	21	21
le450_15c	450	8	47	1	0.256	147	69
le450_15d	450	8	3	1	0.062	25	25
le450_15d	450	8	47	1	0.355	193	84
le450_25a	450	4	3	1	0.003	2	2
le450_25a	450	4	47	1	0.110	110	45
le450_25b	450	4	3	1	0.003	2	2
le450_25b	450	4	47	1	0.008	8	6
le450_25c	450	8	3	1	0.015	7	7
le450_25c	450	8	47	1	0.282	146	55
le450_25d	450	8	3	1	0.030	15	15
le450_25d	450	8	47	1	0.188	56	39
miles250	128	4	3	1	0.000	2	2
miles500	128	14	3	1	0.002	6	6
miles500	128	14	14	16	0.021	79	17

Table 2 continued

Name	$ V $	Density	d	Nodes	CPU	#cuts	Size blocker
miles750	128	25	5	1	0.001	2	1
miles750	128	25	16	7	0.011	20	6
multsol.i.1	197	10	3	1	0.003	3	3
multsol.i.1	197	10	21	1	0.017	39	23
multsol.i.2	188	11	1	1	0.000	0	0
multsol.i.3	184	11	3	1	0.002	2	2
multsol.i.3	184	11	20	1	0.034	73	19
multsol.i.4	185	11	3	1	0.005	14	14

Table 3 Computational results from DIMACS instances

Name	$ V $	Density	d	Nodes	CPU	#cuts	Size blocker
multsol.i.4	185	11	20	1	0.012	38	23
multsol.i.5	186	11	3	1	0.008	18	18
myciel3	11	18	3	1	0.000	2	2
myciel3	11	18	3	1	0.000	2	2
myciel3	11	18	4	1	0.001	4	4
myciel4	23	14	3	1	0.000	1	1
myciel4	23	14	4	1	0.000	7	4
myciel5	47	10	3	1	0.000	3	3
myciel5	47	10	6	1	0.002	24	12
myciel6	95	8	3	1	0.001	8	8
myciel6	95	8	11	1	0.003	26	16
myciel7	191	6	3	1	0.002	3	3
myciel7	191	6	21	1	0.008	18	10
queen7_7	49	40	3	1	0.001	5	5
queen7_7	49	40	6	1	0.006	34	18
queen8_8	64	36	3	1	0.001	9	9
queen8_8	64	36	8	1	0.007	43	23
queen8_12	96	30	3	1	0.004	8	8
queen8_12	96	30	11	1	0.018	62	29
queen9_9	81	32	3	1	0.002	5	5
queen9_9	81	32	10	1	0.008	44	24
queen10_10	100	29	3	1	0.004	10	10
queen10_10	100	29	12	1	0.018	71	33
queen11_11	121	27	3	1	0.004	6	6
queen11_11	121	27	14	1	0.023	64	30
queen12_12	144	25	3	1	0.004	7	7
queen12_12	144	25	16	1	0.046	90	35

Table 3 continued

Name	$ V $	Density	d	Nodes	CPU	#cuts	Size blocker
queen13_13	169	23	3	1	0.005	5	5
queen13_13	169	23	18	1	0.063	76	36
queen14_14	196	21	3	1	0.011	6	6
queen14_14	196	21	21	1	0.110	96	41
queen15_15	225	20	3	1	0.011	8	8
queen15_15	225	20	24	1	0.098	91	42
queen16_16	256	19	3	1	0.022	12	12
queen16_16	256	19	27	1	0.143	106	53
school1	385	12	3	1	0.058	27	27
school1	385	12	40	58	0.867	569	93
school1_nsh	352	11	3	1	0.049	24	24
school1_nsh	352	11	37	1	0.403	296	97
zeroin.i.1	211	9	1	1	0.001	0	0
zeroin.i.2	211	7	3	1	0.002	2	2
zeroin.i.2	211	7	23	1	0.040	78	24
zeroin.i.3	206	8	3	1	0.006	7	7
zeroin.i.3	206	8	22	1	0.029	56	24

In Table 1, we consider random instances. Note that all these instances are solved in the root node even for the large graphs. Furthermore, we solved all the instances in less than 1 h. Notice that, when d is equal to $sp + 1$ then the number of cuts and the size blocker are equals. This is not true when d is equal to $disc$. Our Branch-and-Cut algorithm can solve instances with 2500 vertices in less than 30 min. We remark that the density and the d impact the efficiency of our algorithm. In most of instances we note that $sp + 1 = disc - 1$, except when the density is 2%, there is an instance with $sp + 1 = disc - 2$.

In Tables 2 and 3, we consider some DIMACS instances. Note that all these instances are solved in less than 1 second. For these instances, 4 instances are not solved to the root node, when $d = disc$. Furthermore, for all these DIMACS instances the number of generated cut is less than 600 explaining the good performances given by our algorithm. Remark that, when $d = sp + 1$, only for one instance (miles750), the number of cuts is greater than the size blocker, otherwise the number of cut is equal to the size blocker when $d = sp + 1$.

7 Conclusion

In this paper we considered the most vital vertices shortest s - t path problem. We studied the complexity of this problem and we proposed an integer linear model to solve it. We showed that the associated polytope is integer. Since the problem is NP-hard we deduce that the separation problem, when the vector is fractional, is NP-hard

but we showed that the problem becomes polynomial when the vector is integer. Based on this result we proposed an efficient Branch-and-cut algorithm.

References

1. Israeli, E., Wood, P.K.: Shortest-path network interdiction. *Networks* **40**(2), 97–111 (2002)
2. Khachiyan, L., Boros, E., Borys, K., Elbassioni, K., Gurvich, V., Rudolf, G., Zhao, J.: On short paths interdiction problems: total and node-wise limited interdiction. *Theory Comput. Syst.* **43**, 204–233 (2008)
3. Song, Y., Shen, S.: Risk averse shortest path interdiction. *INFORMS J. Comput.* **23**(3), 527–539 (2016)
4. Furini, F., Ljubić, I., Martin, S., Segundo, P.S.: The maximum clique interdiction problem. *Eur. J. Oper. Res.* **43**, 112–127 (2019)
5. Bazgan, C., Toubaline, S., Vanderpooten, D.: Critical edges for the assignment problem: complexity and exact resolution. *Oper. Res. Lett.* **41**, 685–689 (2013)
6. Hoffman, A.J., Markowitz, H.: A note on shortest path, assignment and transportation problems. *Nav. Res. Logist. Q.* **10**(1), 375–379 (1963)
7. Bazgan, C., Nichterlein, A., Niedermeier, R.: A rene complexity analysis of finding the most vital edges for undirected shortest paths. In: Conference CIAC 2015, pp. 47–60 (2015)
8. Garey, M.R., Johnson, D.S.A.: *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York (1979)
9. Johnson, D.S., Trick, M.A.: *Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge*, October 11–13, 1993, vol. 26. American Mathematical Society (1996). <http://archive.dimacs.rutgers.edu/Challenges/>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.