An efficient algorithm for $st$-Connected Vertex Separator problem

ALTHOBY Haeder Younis

Dept. of Math., Collage of Science, University of AL-Qadisiyah, AL-Diwaniyah, Iraq.

haeder.althoby@qu.edu.iq

April 2021

Abstract

In this paper, we give some basic notions concerning the $st$-connected vertex separator problem ($st$-CVS problem), then we give mathematical formula for $st$-CVS problem, then some special cases on some types of graphs, after then introduce an integer programming formulations for this problem. Also, we introduce a heuristic to solve this problem.

Key words. graph partitioning, connected separator, heuristic algorithm.

1 Introduction

Vertex connectivity is the most fundamental concepts in network reliability theory. There are many articles that have studied $st$-CVS problem as specific cases, clique vertex separator, for example ([3, 4, 8]). This type of algorithm is known as a Fixed-Parameter Tractable Algorithm (FPT), with the parameter being the solution size. Marx et al. [6] show that the $st$-CVS problem is NP-Hard, and in FPT. Narayanaswamy and Sadagopan [7] show that $st$-CVS is NP-Complete on graphs of chordality at least 5 and present an algorithm in polynomial-time for this problem on chordality 4 graphs. They also introduced an algorithm to solve the problem and proved that $st$-CVS parameterized is $W[2]$-hard. Recently, this problem was studied by [5] as complexity and polyhedral point of view. Altoby [1] gives three integer programming formulations and investigation the related polyhedron and gave some valued inequality and heuristic to solve the problem.
Let $G = (V, E)$ be a simple and connected undirected graph where $V$ is the set of vertices and $E$ is the set of edges. Let $n := |V|$ the number of vertices and $m := |E|$ the number of edges. Moreover, each edge $\{u, v\}$ is denoted $uv$. For all $S, T \subseteq V$, we put $\delta(S, T) := \{uv \in E : u \in S, v \in T\}$, $\delta(S) := \delta(S, V \setminus S)$ and $E(S) := \{ij \in E : i, j \in S\}$. For all $F \subseteq E$, we put $V(F) := \{u, v \in V : uv \in F\}$. Moreover, we put $N(S) := \{u \in V \setminus S : \exists v \in S, uv \in E\}$. For each vertex $u \in V$, we set $\delta(u) := \delta(\{u\})$ the set of incident edges at the vertex $u$, the neighborhood of the vertex $u$ is $N(u) := \{v \in V : uv \in E\}$ and $d(u) := |\delta(u)|$ The degree of the vertex $u$. For the simple graphs we have $d(u) = |N(u)|$. For all $S \subseteq V$, $G[S]$ is the graph induced by $S$. For all $S \subseteq V$, for all $u \in V$, we put $NS(u) := N(u) \cap S$ the neighborhood of the vertex $u$ in $G[S]$ and $dS(u) := |NS(u)|$ The degree of the vertex $u$ in $G[S]$. $\beta_{st}$ is the maximum number of disjoint paths between $s$ and $t$.

2 The $st$-connected separator

Let $s$ and $t$ be two disjoint vertices of $V$, not adjacent. An $st$- connected separator in the graph $G$ is a set $S \subseteq V \setminus \{s, t\}$ such that

(i) there are no more paths between $s$ and $t$ in $G[G \setminus S]$, and

(ii) the graph $G[S]$ is connected.

We shall assume that $G$ is connected. Moreover, for any $st$-connected separator $S$ in $G$, there exists at least one bi-partition $(A, B)$ of $V \setminus S$ such that $s \in A$, $t \in B$, and $\delta(A, B) = \phi$.

Let $a \in \mathbb{Z}_+$ be a positive integer. A non-adjacent pair $(s, t) \in V$ in two different components of $G(V \setminus S)$ and $w : V \rightarrow \mathbb{R}_+$. For every connected separator $S$, we associate an incidence vector $(x) \in \mathbb{R}^n$ defined by $x_v = 1$ if $v \in S$ and 0 otherwise, $S \subseteq V \setminus \{s, t\}$, and $|S| \leq a$. The convex hull of the incidence vectors of all separators is called the connected $(s, t)$ vertex separator polyhedron and denoted $P_{st}(G)$, that is,

$$P_{st}(G) = \text{conv}\{x^S : S \text{ is a } st - \text{connected separator }\}$$

The $(s, t) - CVS$ is equivalent to solving the linear programm

$$\min \left\{ \sum_{v \in V} (w_v x_v), \ x \in P_{st}(G) \right\}.$$
The \( st \)-connected separator problem can be formulated as follows:

\[
\min \sum_{v \in V} (x_v)
\]
\[
s.t.
\]
\[
x_s = x_t = 0 \quad (2.1)
\]
\[
X(V_I(P_{st})) \geq 1 \quad \forall P_{st} \in \Gamma_{st} \quad (2.2)
\]
\[
X(N(U)) \geq x_u + x_v - 1 \quad \forall u, v \in V \quad (2.3)
\]
\[
x_v \in \{0, 1\} \quad \forall v \in V \quad (2.4)
\]

where \( P_{st} \) is a path between \( s \) and \( t \), \( V_I(P_{st}) \) is the set of vertices in the internal path \( P_{st} \) and \( \Gamma_{st} \) is the family of all paths between \( s \) and \( t \), and \( N(u) \) is the neighborhood of the vertex \( u \).

Inequalities (2.2) mean that from any path between \( s \) and \( t \), there exist at least one vertex from the internal path in \( S \).

Inequalities (2.3) give us guarantee that the separator is connected.

**Proposition 2.1** For \( (st-CVS) \) problem, \( \beta_{st} \leq \text{optimal solution} \leq n - 2. \)

**proof**: From the inequality (2.2), every path between \( s \) and \( t \) have at least one vertex in \( S \), means \( \beta_{st} < \text{optimal solution} \), if \( S \) is connected then \( \beta_{st} = \text{optimal solution} \). In the other side, it is obvious that from (2.1) that the optimal solution \( \leq n - 2. \)

We consider \( \beta_{st} \) is a lower bound for our problem.

### 3 Study of particular cases of graphs

#### 3.1 Study of the entire polytope for trees

Let \( T := (V, E) \) be a tree and \( s, t \in V \) two leaves of the tree \( T \). Is the whole polyhedron \( P(G) \) described by the following constraints:

\[
\begin{align*}
\begin{cases}
  x(P_{st}) \geq 1, & \forall P_{st} \in \Gamma_{st}, \\
  x(N(u)) \geq x_u + x_v - 1, & \forall u, v \in V \text{ and } uv \notin E, \\
  0 \leq x_u \leq 1, & \forall u \in V.
\end{cases}
\end{align*}
\]
3.2 Study of the integral polyhedron for wheel graphs

Let $G := (V, E)$ be a wheel graph where $V := C \cup \{u^*\}$ and $u^*$ is the center of the wheel and $C$ is the vertices which form the cycle of the wheel. And let $s, t \in C$. The integral polyhedron $P(G)$ is described by the following constraints:

$$\begin{align*}
& x(P_{st}) \geq 1, \quad \forall P_{st} \in \Gamma_{st}, \\
& x_{u^*} = 1, \\
& 0 \leq x_u \leq 1, \quad \forall u \in V.
\end{align*}$$

(3.2)

4 Heuristic

In this section, we introduce an algorithm to solve mixed-integer program for our problem (we say it P):

$$\text{Min} \sum_{v \in V} (x_v)$$

subject to

$$\begin{align*}
& x_s = 0, \quad x_t = 0 \quad (4.1) \\
& x_u + x_v \leq 1, \quad x_v + x_u \leq 1 \quad \forall (uv) \in E \quad (4.2) \\
& x_v + x_u \leq 1 \quad \forall v, u \in V \quad (4.3) \\
& \sum_{v \in V} (x_v + x_u) = n - \beta_{st} \quad (4.4) \\
& x_v, x_u \in \{0, 1\} \quad \forall v, u \in V \quad (4.5)
\end{align*}$$

in polynomial time. This algorithm depends on neighbourhood of vertices in set $C$. The algorithm consist two stages, the first stage is to find a set of separator $S$ by solving the problem (P) integer programming. The second stage is check $S$ is connected or not. Let $C$ be the set of vertices from sovle $P$, and let $S$ be a set of all vertices in $C$ and its neighbourhood. We provide details for our algorithm for CVSP.
Algorithm 1 ALTHOBY

1: Solve P1 to find $C=v_0, v_1, ..., v_k$.
2: $S = (v_0 \cup N(v_0)), (v_1 \cup N(v_1)), ..., (v_k \cup N(v_k))$.
3: Pick the vertex in $C$ has the biggest degree (let $v_0$) and put it in the set $S'$ and delete it from $S$.
4: Take every element $v_i$ in $C$ and check it if its an adjacent $v_0$ put it in $S'$ and delete it from $S$.
5: else, check if any vertex in $N(v_i)$ an adjacent $v_0$ then put it in $S'$ with $v_i$ and delete it from $S$.
6: If $S$ is empty then stop and $S'$ is the solution.
7: else ($S$ is not empty) then consider $S = S' \cup N(S')$ and $C = C - \forall v_i \in S'$ and goto step 3.
5 Conclusions and Remarks

In this article, we have considered mathematical form for \(st\)-connected separator problem. Also, we gave a good lower bound. Also, we gave some particular cases of graphs and then a heuristic to solve this problem in a short time. To develop this work, we need to solve this problem exactly by using Branch and Cut. Also, we need to expand our search within this problem by studying particular cases of graphs such Halin graphs, Free graph, ...etc.

References

[1] H. Althoby, *Theoretical and Numerical studies on the graph partitioning problem*, Ph.D thesis. Université de Caen Normandie, Caen, (2017).

[2] B.V. Cherkassky and A.V. Goldberg, *On Implementing Push-Relabel Method for the Maximum Flow Problem*, Algorithmica, 19(1997), pp. 390-410.

[3] E. Dahlhaus; M. Karpinski; and M.B. Novick, *Fast parallel algorithms for the clique separator decomposition*, In Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms: SODA 90, Philadelphia, PA, USA, (1990), pp. 244–251.

[4] H.G. Leimer, *Optimal decomposition by clique separators*, Discrete Math., n. 113,(1993), pp. 99-123.

[5] Y. Magnouche, *The multi-terminal vertex separator problem: Complexity, Polyhedra and Algorithms*, Ph.D thesis. Université de Dauphine, Paris, (2017).

[6] D. Marx, B. O’Sullivan, and I. Razgon, *Finding small separators in linear time via treewidth reduction*, ACM Trans. Algorithms, n. 9(4),(2013), pp. 1-30.

[7] N.S. Narayanaswamy and N. Sadagopan, *Connected \(s, t\)-Vertex Separator Parameterized by Chordality*, Journal of Graph Algorithms and Applications, n. 1, 19(2015), pp. 549–565.

[8] K.G. Olesen; and A.L. Madsen, *Maximal prime subgraph decomposition of Bayesian networks*, IEEE Trans. Syst. Man Cybernet, n. 32,(2002), pp. 21-31.
[9] HTTP://www.avglab.com/andrew/soft.html

[10] HTTP://www.ilog.com