J Comb Optim (2010) 20: 249–258
DOI 10.1007/s10878-008-9206-5

# Hardness of $k$-Vertex-Connected Subgraph Augmentation Problem

**Changcun Ma · Donghyun Kim · Yuexuan Wang · Wei Wang · Nassim Sohaee · Weili Wu**

Published online: 14 January 2009

**Abstract** Given a $k$-connected graph $G=(V,E)$ and $V' \subset V$, $k$-Vertex-Connected Subgraph Augmentation Problem ($k$-VCSAP) is to find $S \subset V \setminus V'$ with minimum cardinality such that the subgraph induced by $V' \cup S$ is $k$-connected. In this paper, we study the hardness of $k$-VCSAP in undirect graphs. We first prove $k$-VCSAP is APX-hard. Then, we improve the lower bound in two ways by relying on different assumptions. That is, we prove no algorithm for $k$-VCSAP has a PR better than

This work is supported in part by the National Scientific Foundation under Grant Nos. IIS-0513669, CCF-0750992, and CCF-0621829. This work is also supported in part by the National Natural Science Foundation of China Grant Nos. 60553001 and 60604033, the National Basic Research Program of China Grant Nos. 2007CB807900, 2007CB807901, and the Hi-Tech research & Development Program of China Grant No. 2006AA10Z216.

C. Ma (✉) · Y. Wang
Institute for Theoretical Computer Science, Tsinghua University, Beijing 100084,
People's Republic of China
e-mail: mcc03@mails.tsinghua.edu.cn

Y. Wang
e-mail: wangyuexuan@tsinghua.edu.cn

D. Kim · N. Sohaee · W. Wu
Department of Computer Science, University of Texas at Dallas, Richardson, TX 75083, USA

D. Kim
e-mail: donghyunkim@student.utdallas.edu

N. Sohaee
e-mail: sohaee@utdallas.edu

W. Wu
e-mail: weiliwu@utdallas.edu

W. Wang
Department of Mathematics, Xi'an Jiaotong University, Xi'an 710049, People's Republic of China
e-mail: wang_weiw@163.com

$O(\log(\log n))$ unless $P = NP$ and $O(\log n)$ unless $NP \subseteq DTIME(n^{O(\log\log n)})$, where $n$ is the size of an input graph.

**Keywords** Network survivability · Graph connectivity

## 1 Introduction

The survivability or fault-tolerance of a network can be defined as an ability of the network to perform correctly even though some part of the network fails, and is closely related to the connectivity of the network. For many network applications such as a sensor network for battlefield monitoring or wild fire tracking (Akyildiz et al. 2002), survivability is one of most important requirements, and therefore a number of related problems are proposed.

One good example is Survivable Network Design Problem (SNDP$_e$) (Frederickson and JáJá 1981). Given an undirected graph $G = (V, E)$ with non-negative edge cost and a non-negative connectivity requirement $k_{uv}$ for every (unordered) pair of vertices $u, v \in V$, SNDP$_e$ is to find a subgraph with minimum cost such that each pair of vertices $u, v \in V$ is connected by at least $k_{uv}$ edge-disjoint paths in the subgraph. SNDP$_e$ is NP-complete since Minimum Steiner Tree problem is a special case of SNDP$_e$. One variation of SNDP$_e$, say SNDP$_v$, is to find a minimum cost subgraph, in which each pair of vertices $u, v$ is connected by at least $k_{uv}$ vertex-disjoint paths. A well-studied special case of SNDP$_v$ is $k$-Vertex Connected Spanning Subgraph ($k$-VCSS), in which $k_{uv} = k$ for all $u, v$ pairs (i.e. each pair of vertices in the subgraph is connected by at least $k$ vertex-disjoint paths). $k$-VCSS is NP-hard even for $k = 2$ with uniform edge cost (Kortsarz et al. 2004). This result can be shown simply from Hamiltonian Cycle Problem. Therefore, SNDP$_v$ is NP-hard. In Kortsarz et al. (2004), they also show that SNDP$_v$ cannot be approximated within ratio $2^{\log^{1-\epsilon} n}$, for a positive constant $\epsilon$ unless $NP \subseteq DTIME(n^{polylog(n)})$. The hardness result of a degree bounded version of SNDP$_v$ is shown in Lau et al. (2007).

Another example is $(l, k)$-Vertex-Connectivity Augmentation Problem($(l, k)$-VCAP). A graph $G = (V, E)$ is called $k$ vertex-connected if $|V| \geq k+1$ and the graph remains connected after deleting any $k - 1$ vertices (Jordan 1997). Given an undirected $k$-connected graph $G = (V, E)$ with non-negative edge cost and a $l$-connected subgraph $G' = (V', E')$, $(l, k)$-VCAP is to find a minimum cost edge set $S \subseteq E$ such that subgraph $(V', E' \cup S)$ is $k$-connected. It is easy to see that $(0, 1)$-VCAP equivalent to Minimum Spanning Tree Problem and therefore is polynomial time solvable. On the other hand, $(0, 2)$-VCAP is NP-hard since it generalizes Hamiltonian Cycle Problem. $(1, 2)$-VCAP is shown to be NP-hard in Frederickson and JáJá (1981). The authors in Kortsarz et al. (2004) showed that $(k, k+1)$-VCAP is APX-hard for any $k$, and therefore is NP-hard.

We observed that all the previous work for increasing subgraph connectivity by augmentation are working on an input graph that has non-negative weight on edges and trying to find a set of edges to satisfy some requirements. In this paper, we introduce another kind of connectivity problem as follows.

**Definition 1** ($k$-Vertex Connected Subgraph Augmentation Problem($k$-VCSAP)) Given a $k$-vertex connected graph $G=(V,E)$ and a subset $S \subseteq V$, find a minimum size subset $S'$ of vertices, such that the subgraph induced by $S \cup S'$ is $k$-vertex connected.

$k$-VCSAP abstracts the problem of enhancing fault-tolerance or connectivity of a subset of nodes in a wireless network. For example, given a set of clusterheads $S$ in a wireless network, we may want to find an additional subset of nodes $S'$ with minimum cardinality which makes any two clusterheads $u, v \in S$ to be connected by $k$ vertex-disjoint paths (i.e. a graph induced $S \cup S'$ is $k$-connected).

The contributions of this paper can be summarized as follows. First, we formally define a new optimization problem $k$-VCSAP. Second, given $P \neq NP$, we introduce a reduction from Maximum 3-Dimensional Matching Problem (M3DMP) to $k$-VCSAP and use this reduction to show the APX-hardness of $k$-VCSAP. Next, we improve the lower bound to $O(\log(\log n))$ by inventing a new reduction from Minimum Set Cover Problem (MSCP) to $k$-VCSAP. At last, we use a stronger assumption that $NP \nsubseteq DTIME(n^{O(\log\log n)})$ and further improve the lower bound to $O(\log n)$ using another new reduction from Set Cover Packing Problem (SCPP).

The rest of this paper is organized as follows. Section 2 shows that APX-hardness of $k$-VCSAP. In Sects. 3 and 4, we improve the lower bound to $O(\log(\log n))$ and $O(\log n)$ depending on two different assumptions. Section 5, we make a conclusion and introduce several future work.

## 2 APX-hardness of $k$-VCSAP

In this section, we prove $k$-VCSAP is APX-hard. This can be done by a reduction from Maximum 3-Dimensional Matching Problem (M3DMP), which is proven to be MAX SNP-hard (Kann 1991). Here is the formal definition of M3DMP.

**Definition 2** (Maximum 3-Dimensional Matching Problem (M3DMP)) Given three disjoint sets $W$, $Y$, and $Z$ satisfying $|W|=|Y|=|Z|$ and a set of hyperedges $M \subseteq W \times Y \times Z$, M3DMP is to find the largest subset $M' \subseteq M$ which is a matching. In other word, if $(w, y, z)$ and $(w', y', z')$ are in $M'$ then $w \neq w'$, $y \neq y'$, and $z \neq z'$.

We first show that $k$-VCSAP is APX-hard for $k=2$, and then generalize this result for any $k$. Let $\mathcal{I}=(M, W, Y, Z)$ be an instance of M3DMP with $|M|=p$, $W=\{w_1, w_2, \ldots, w_q\}$, $Y=\{y_1, y_2, \ldots, y_q\}$, and $Z=\{z_1, z_2, \ldots, z_q\}$. Then, Algorithm 1 produces an instance for 2-VCSAP $\mathcal{J}=(G_{\mathcal{J}}=(V_{\mathcal{J}}, E_{\mathcal{J}}), S_{\mathcal{J}})$ from $\mathcal{I}$. Note that $G_{\mathcal{J}}$ is 2-connected and the subgraph induced by $S_{\mathcal{J}}$ is 1-connected. Then, in $\mathcal{J}$, our goal is to find a minimum cardinality set $S'_{\mathcal{J}} \subset V_{\mathcal{J}} \setminus S_{\mathcal{J}}$ such that the subgraph induced by $S_{\mathcal{J}} \cup S'_{\mathcal{J}}$ is 2-connected. Obviously, $S'_{\mathcal{J}} \subseteq \{v_m | m \in W \times Y \times Z\} \cup \{v_{\hat{m}},\ w_{\hat{m}},\ y_{\hat{m}},\ z_{\hat{m}} | \hat{m} \notin W \times Y \times Z\}$. Remember that in both M3DMP and $k$-VCSAP, the cost of a solution for each problem is the cardinality of the solution. Now, we show that $k$-VCSAP is APX-hard. In the following lemmas and theorems, we denote optimum solutions of $\mathcal{I}$ and $\mathcal{J}$ by $OPT(\mathcal{I})$ and $OPT(\mathcal{J})$, respectively.

**Algorithm 1 Reduction1** ($\mathcal{I} = (M, W, Y, Z)$)

```
$S_{\mathcal{J}} \leftarrow \{s, t\} \cup W \cup Y \cup Z$.
$V_{\mathcal{J}} \leftarrow S_{\mathcal{J}}$.
for $i = 1$ to $q$ do
    for $j = 1$ to $q$ do
        for $k = 1$ to $q$ do
            if $m = (w_i, y_j, z_k) \in M$ then
                Add a new vertex $v_m$ to $V_{\mathcal{J}}$.
            else/* $\hat{m} = (w_i, y_j, z_k) \notin M$. */
                Add four new vertices $v_{\hat{m}}$, $w_{\hat{m}}$, $y_{\hat{m}}$, and $z_{\hat{m}}$ to $V_{\mathcal{J}}$.
            end if
        end for
    end for
end for
$E_{\mathcal{J}} \leftarrow \{(s, t)\} \cup \{(s, w_i), (s, y_j), (s, z_k) | 1 \leq \forall i, j, k \leq q\}$
        $\cup \{(v_m, w_i), (v_m, y_j), (v_m, z_k) | m = (w_i, y_j, z_k) \in M\}$
        $\cup \{(v_{\hat{m}}, w_{\hat{m}}), (v_{\hat{m}}, y_{\hat{m}}), (v_{\hat{m}}, z_{\hat{m}}), (w_{\hat{m}}, w_i), (y_{\hat{m}}, y_j),$
            $(z_{\hat{m}}, z_k) | \hat{m} = (w_i, y_j, z_k) \notin M\}$.
        $\cup \{(v_m, t) | \forall m\} \cup \{(v_{\hat{m}}, t) | \forall \hat{m}\}$.
Return $\mathcal{J} = (G_{\mathcal{J}} = (V_{\mathcal{J}}, E_{\mathcal{J}}), S_{\mathcal{J}})$.
```

**Lemma 1** *If $|OPT(\mathcal{I})| = q$, then $|OPT(\mathcal{J})| = q$.*

*Proof* By the reduction, if $\mathcal{I}$ has a perfect matching $\{m_1, \ldots, m_q\}$, then $\mathcal{J}$ has a corresponding solution $S'_{\mathcal{J}} = \{v_{m_1}, \ldots, v_{m_q}\}$. We claim that $S'_{\mathcal{J}}$ is an optimum solution of $\mathcal{J}$. Since to make the subgraph induced by $S_{\mathcal{J}} \cup S'_{\mathcal{J}}$ 2-vertex connected, the degree of each vertex in $W \cup Y \cup Z$ has to be at least two in this subgraph. Meanwhile, we cannot increase the degree of two vertices in $W$ at the same time by adding only one vertex in $V_{\mathcal{J}} \setminus S_{\mathcal{J}}$ to $S'_{\mathcal{J}}$, and this is also true for $Y$ and $Z$. As a result, we have $|S'_{\mathcal{J}}| \geq q$. □

**Lemma 2** *If $|OPT(\mathcal{J})| \leq (1 + \epsilon)q$, then $|OPT(\mathcal{I})| \geq (1 - 3\epsilon)q$ for a positive constant $\epsilon$.*

*Proof* Suppose $\mathcal{J}$ has a solution $S'_{\mathcal{J}}$ satisfying $|S'_{\mathcal{J}}| \leq (1 + \epsilon)q$. We show that $\mathcal{I}$ associated with $\mathcal{J}$ has a matching of size at least $(1 - 3\epsilon)q$, which implies that $|OPT(\mathcal{I})| \geq (1 - 3\epsilon)q$. Let $P(W)$ be the set of vertices in $W$ that are adjacent to only one vertex $v_m \in S'_{\mathcal{J}}$ for some $m \in M$ and $Q(W)$ be $W \setminus P(W)$. Then, each vertex in $Q(W)$ either (1) is neighboring with at least two vertices in $S'_{\mathcal{J}}$, or (2) is neighboring with only one vertex in $S'_{\mathcal{J}}$ which is connected to $t$ through a path whose length is exactly two hops (Fig. 1). Remember that no two vertices in $W$ are adjacent to the same vertex in $V_{\mathcal{J}} \setminus S_{\mathcal{J}}$. Therefore, we have $|P(W)| + 2|Q(W)| \leq |S'_{\mathcal{J}}| \leq (1 + \epsilon)q$ and $|P(W)| + |Q(W)| = |W| = q$. As a result, we have

$$|P(W)| + 2|Q(W)| - (|P(W)| + |Q(W)|) = |Q(W)| \leq (1 + \epsilon)q - q \leq \epsilon q.$$

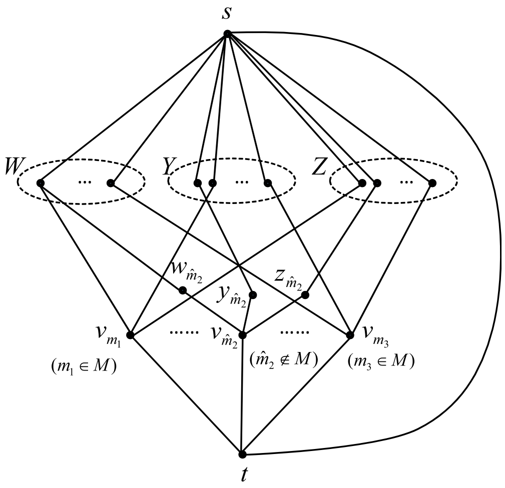

**Fig. 1** This figure shows how an M3DMP instance is reduced to a $k$-VCSAP instance by Algorithm 1

Same result holds also for $Y$ and $Z$. Let $S_w$ be the set of vertices in $S'_{\mathcal{J}}$ that are adjacent to some vertex in $P(W)$. Then, $S_w \setminus S'_w$ is a matching, where $S'_w \subset S_w$ is a set of vertices which are also connected to some vertex in $Q(Y)$ or $Q(Z)$. Since we have $|OPT(\mathcal{I})| \geq |S_w \setminus S'_w|$, $|Q(Y)| \leq \epsilon q$, and $|Q(Z)| \leq \epsilon q$, we conclude that

$$\begin{aligned} |OPT(\mathcal{I})| &\geq |S_w \setminus S'_w| \geq |S_w| - |Q(Y)| - |Q(Z)| \\ &\geq (1-\epsilon)q - \epsilon q - \epsilon q = (1-3\epsilon)q. \end{aligned}$$ □

The following result is proven in Petrank (1992), in which M3DMP-B represents a bounded version of M3DMP where an element in $W \cup Y \cup Z$ appears at most $B$ times in $M$.

**Theorem 1** (Petrank) *For some fixed* $\epsilon_0 > 0$, *it is NP-hard to distinguish whether an instance of M3DMP-B with* $|W| = |Y| = |Z| = q$ *has a perfect matching* (*of size* $q$) *or every matching has size at most* $(1-\epsilon_0)q$.

Combining Lemmas 1 and 2 together with Theorem 1, we can prove following theorem.

**Theorem 2** *For some constant* $\epsilon > 0$, *it is NP-hard to approximate $k$-VCSAP within a factor of* $1+\epsilon$.

*Proof* Since our construction does not have any requirement on the occurrence of elements, our reduction also works for M3DMP-B to $k$-VCSAP. By Lemma 1, if $\mathcal{I}$ has a perfect matching of size $q$, $\mathcal{J}$ has an optimum solution of size $q$. As a contrapositive of Lemma 2, if $|OPT(\mathcal{I})| < (1-\epsilon_0)q$, then $|OPT(\mathcal{J})| \geq (1+\epsilon_0/3)q$. That is, our reduction is gap-preserving. This finishes the proof for $k=2$. For $k=1$, we simply eliminate $s$ in the reduction and can prove this theorem using the same arguments for $k=2$. At last, for $k \geq 3$, we (1) add $k-2$ new vertices $\{s_1, s_2, \ldots, s_{k-2}\}$, (2) connect them to every vertices in graph $G_{\mathcal{J}}$, (3) include them in $S_{\mathcal{J}}$ in the reduction, and (4) use the same arguments for the other cases. □

## 3 An improved lower bound of $O(\log(\log n))$ on approximation ratio for algorithms for $k$-VCSAP

In this section, we improve the lower bound for $k$-VCSAP from $1+\epsilon$ to $O(\log(\log n))$ under the assumption $P \neq NP$. This is done by a reduction from Minimum Set Cover Problem (MSCP). We start with the definition of MSCP.

**Definition 3** (Minimum Set Cover Problem (MSCP)) Given a collection $C$ of subsets of a finite set $S$, find $C' \subseteq C$ with minimum number of subsets such that every element in $S$ belongs to at least one member of $C'$.

MSCP is a classic problem in complexity theory. This problem is one of Karp's 21 NP-complete problems in Johnson (1973), which also introduces the best result for approximating MSCP, $O(\log |S|)$. As for the hardness of approximation, Raz and Safra proved the following theorem (Raz and Safra 1997).

**Theorem 3** (Inapproximability of MSCP) *Approximating MSCP within $c\log n$ is NP-hard for some constant c and $|S| = n$.*

Now, we introduce the reduction and prove the better bound. As we did in the previous section, we first work on this for the case $k = 2$ and then generalize it for any $k$. Algorithm 2 introduces a reduction from MSCP to $k$-VCSAP. It returns an

**Algorithm 2 Reduction2 ($\mathcal{I} = (C, S)$)**

```
$S_{\mathcal{J}} \leftarrow \{s, t\} \cup S$.
$V_{\mathcal{J}} \leftarrow S_{\mathcal{J}} \cup C$.
$E_{\mathcal{J}} \leftarrow (s, t)$. /* Connect $s$ to $t$. */
for each $C' \in C$ do /* Construct a bipartite graph from $\mathcal{I}$. */
    for each $v \in S$ do
        if $v \in C'$ in $\mathcal{I}$ then
            Add $(C', v)$ to $E_{\mathcal{J}}$.
        end if
    end for
end for
for each $C' \in C$ do /* Connect $s$ to each node in $C$. */
    Add $(s, C')$ to $E_{\mathcal{J}}$.
end for
for each $v \in S$ do /* Connect each node in $S$ to $t$. */
    Add $(v, t)$ to $E_{\mathcal{J}}$.
end for
Return $\mathcal{J} = (G_{\mathcal{J}} = (V_{\mathcal{J}}, E_{\mathcal{J}}), S_{\mathcal{J}})$.
```

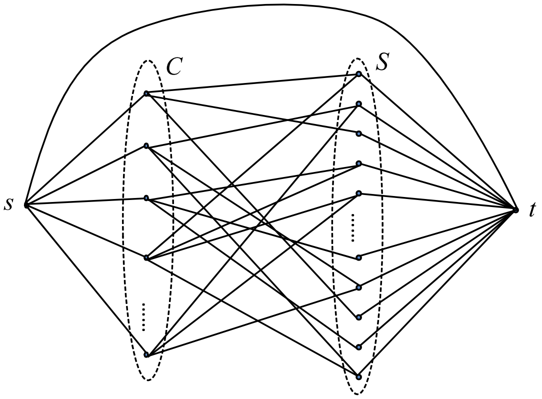

**Fig. 2** This figure shows how an MSCP instance is reduced to a $k$-VCSAP instance by Algorithm 2

instance $\mathcal{J} = (G_{\mathcal{J}} = (V_{\mathcal{J}}, E_{\mathcal{J}}), S_{\mathcal{J}})$ of $k$-VCSAP from an instance $\mathcal{I} = (C, S)$ of MSCP. By Lines 4–10 in Algorithm 2, an MSCP instance is represented by a bipartite graph. Then, we add additional edges on the bipartite graph and finish the reduction (Fig. 2). The subgraph induced by $S_{\mathcal{J}} = \{s, t\} \cup S$ is 1-connected. Then, in $\mathcal{J}$, we need to find a subset $S'_{\mathcal{J}}$ of vertices from $C$ such that the subgraph induced by $S_{\mathcal{J}} \cup S'_{\mathcal{J}}$ is 2-vertex connected. Note that in a 2-vertex connected graph, every vertex has degree at least 2. Consequently, every vertex in $S$ should be adjacent to at least one vertex of $S'_{\mathcal{J}}$, which makes $S'_{\mathcal{J}}$ a set cover of $S$. On the other hand, if we have a set cover $S'_{\mathcal{I}}$ of $S$, it is easy to verify that the subgraph induced by $S_{\mathcal{J}} \cup S'_{\mathcal{I}}$ is 2-vertex connected. In conclusion, $\mathcal{I}$ and $\mathcal{J}$ are equivalent, i.e., an optimum solution of $\mathcal{I}$ is also an optimum solution of $\mathcal{J}$, and vice versa.

**Theorem 4** *Approximating $k$-VCSAP within $c^* \log(\log n)$ for some constant $c^*$ is NP-hard, where $n$ is the number of nodes in the $k$-VCSAP problem instance.*

*Proof* As we have shown before, when $k = 2$ finding a minimum set cover for $\mathcal{I}$ and finding an optimum solution for $\mathcal{J}$ are equivalent. Together with Theorem 3, we know that $\mathcal{J}$ cannot be approximated within $c \log |S|$. Since $C$ could be power set of $S$, i.e., $|C| \le 2^{|S|}$, $n = |V_{\mathcal{J}}| \le 2^{|S|} + |S| + 2$, which means that $|S| = \Omega(\log n)$. Then, for some constant $c'$ and a large $n$, $|S| \ge c' \log n$. So, for a sufficiently large $n$, we have

$$c \log |S| \ge c \log(c' \log n) = c \log(\log n) + c \log c' \ge c^* \log \log n.$$

To see that this theorem is correct for any $k$, we need to modify our reduction in this section as follows and apply the arguments used for $k = 2$. Now, we prove this theorem for $k$ values other than 2. For $k = 1$, we simply remove vertex $t$ from $\mathcal{J}$. For $k \ge 3$, we add $k - 2$ new vertices $\{t_1, t_2, \dots, t_{k-2}\}$, connect them to every vertices in graph $G$ and include them in $S_{\mathcal{J}}$. One remark here is for sufficiently large $n$, we have $k < |S|$ since $k$ is a given constant, therefore $n \le 2^{|S|} + 2|S|$ which means that $c'$ can be independent from $k$ and so is $c^*$. □

## 4 An improved lower bound of $O(\log n)$ on approximation ratio for algorithms for $k$-VCSAP under stronger assumption

In Sects. 2 and 3, our proofs for the lower bounds relied on the assumption that $P \neq NP$. In this section, we further improve the lower bound under a stronger assumption that $NP \not\subseteq DTIME(n^{O(\log\log n)})$. This is done by a reduction from Set Cover Packing Problem (SCPP). Here is the definition of SCPP.

**Definition 4** (Set Cover Packing Problem (SCPP)) Let $G = (V_1, V_2, E)$ be a bipartite graph. A set cover (of $V_2$) in $G$ is a subset $S \subseteq V_1$ such that every vertex of $V_2$ is adjacent to at least one element in $S$. SCPP is to find a maximum number of pairwise-disjoint set covers of $V_2$.

Feige et al. proved the following theorem for SCPP (Feige et al. 2002).

**Theorem 5** *Let $L \in NP$ and $\epsilon$ be a positive constant. Then, there exists an algorithm (or a reduction) $\mathcal{A}$ satisfying following conditions.*

- *The running time of $\mathcal{A}$ is nearly polynomial, namely $O(n^{O(\log\log n)})$.*
- *Given an instance $x$ for $L$, $\mathcal{A}$ returns an instance $G = (V_1, V_2, E)$ for SCPP and a constant $d$ such that*

  1. $|V_1| \le |V_2|^{\epsilon}$,
  2. *if $x \in L$, $V_1$ can be partitioned into $d$ equal-sized set covers of $V_2$, which makes the size of minimum set cover for $V_2$ to be at most $|V_1|/d$, and*
  3. *if $x \notin L$, then the size of any set cover of $V_2$ is no less than $(|V_1|/d) \cdot (1-\epsilon)\ln|V_1 \cup V_2|$.*

For our purpose in this section, we reuse Algorithm 2 to reduce SCPP to $k$-VCSAP. That is, for a SCPP instance $\mathcal{I} = (V_1, V_2, E)$, we construct an MSCP instance $\mathcal{I}' = (C, S)$ and get a $k$-VCSAP instacne $\mathcal{J} = \text{Reduction2}(\mathcal{I}')$ (Fig. 3). Now, we improve the lower bound under a stronger assumption that $NP \not\subseteq DTIME(n^{O(\log\log n)})$.

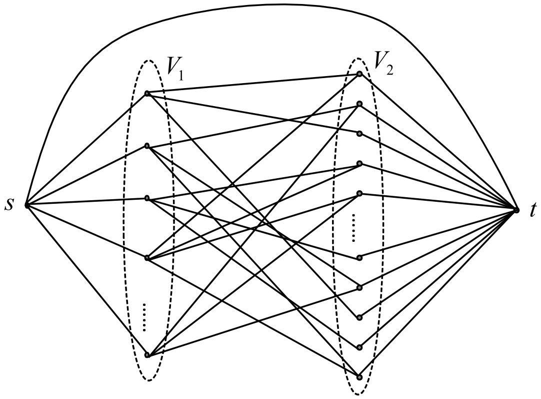

**Fig. 3** This figure shows how a $k$-VCSAP instance is constructed from an SCPP instance by our algorithm

**Theorem 6** *$k$-VCSAP cannot be approximated within* $(1-\epsilon^*)\ln n$ *unless NP* $\subseteq$ *DTIME*$(n^{O(\log\log n)})$ *for any positive constant* $\epsilon^*$.

*Proof* Using the similar arguments as previous section, we can show that the optimum solution of $\mathcal{J}$ is a minimum set cover of $V_2$ and vice versa. According to Theorem 5, if $x \in L$, then the size of a minimum set cover for $V_2$ is at most $|V_1|/d$. On the other hand, if $x \notin L$, the size of any set cover of $V_2$ is at least $(|V_1|/d) \cdot (1-\epsilon)\ln|V_1 \cup V_2|$. Consequently, we cannot distinguish these two cases in polynomial time since otherwise, we should be able to determine whether $x$ is in $L$ or not in polynomial time, which implies that *NP* $\subseteq$ *DTIME*$(n^{O(\log\log n)})$. For any fixed $\epsilon^* > 0$ and a sufficiently large positive $n$, we can find $\epsilon > 0$ such that $(1-\epsilon)\ln n > (1-\epsilon^*)\ln(n+2)$. This finishes proof for $k=2$. To see that this theorem is correct for any $k$, we simply need to modify our reduction in this section as below and apply the arguments used for $k=2$. For $k=1$, we simply remove vertex $t$ from $\mathcal{J}$. For $k \geq 3$, we add $k-2$ new vertices $\{t_1, t_2, \ldots, t_{k-2}\}$, connect them to every vertices in graph $G_{\mathcal{J}}$, and include them in $S_{\mathcal{J}}$. □

## 5 Conclusion and future work

In this paper, we established the APX-hardness of $k$-VCSAP and introduced two better lower bounds on approximation ratio of algorithms for this problem depending on different assumptions. As a future work, we are interested in improving the lower bound as well as finding an upper bound by inventing an approximation algorithm for $k$-VCSAP. In Definition 5, we introduce a new interesting network connectivity problem, which is one variation of $k$-Vertex Connected Spanning Subgraph Problem ($k$-VCSSP).

**Definition 5** In a $k$-vertex connected graph $G=(V,E)$, find a set $S$ with minimum number of vertices such that the subgraph induced by $S$ is $k$-vertex connected.

A simple observation is that when $k=1,2$ this problem is polynomial time solvable. As for $k \geq 3$, we do not even know whether it is NP-hard or not.

## References

Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E (2002) A survey on sensor networks. IEEE Commun Mag 40:102–114

Feige U, Halldórsson MM, Kortsarz G, Srinivasan A (2002) Approximating the domatic number. SIAM J Comput 32(1):172–195

Frederickson GN, JáJá J (1981) Approximation algorithms for several graph augmentation problems. SIAM J Comput 10(2):270–283

Johnson DS (1973) Approximation algorithms for combinatorial problems. In: Proc of the fifth annual ACM symposium on theory of computing, pp 38–49

Jordan T (1997) A note on the vertex-connectivity augmentation problem. University of Southern Denmark

Kann V (1991) Maximum bounded 3-dimensional matching is MAX SNP-complete. Inf Process Lett 37:27–35

Kortsarz G, Krauthgamer R, Lee JR (2004) Hardness of approximation for vertex-connectivity network design problems. SIAM J Comput 33(3):704–720

Lau LC, Naor JS, Salavatipour MR, Singh M (2007) Survivable network design with degree or order constraints. In: Proceedings of the thirty-ninth annual ACM symposium on theory of computing, June 11–13, 2007, San Diego, California, USA

Petrank E (1992) The hardness of approximation: gap location. Technical report 754, Computer Science Department, Technion, Israel Institute of Technology, Haifa, Israel

Raz R, Safra S (1997) A sub-constant error-probability low-degree test, and sub-constant error-probability PCP characterization of NP. In: Proc of the twenty-ninth annual ACM symposium on theory of computing, pp 475–484