

A solution to a conjecture on the generalized connectivity of graphs ¹

Lily Chen^{a,b}, Xueliang Li^b, Mengmeng Liu^{b,c}, Yaping Mao^{b,d}

^aSchool of Mathematics Science
Huaqiao University, Quanzhou 362021, P. R. China

^b Center for Combinatorics and LPMC-TJKLC
Nankai University, Tianjin 300071, P. R. China

^cDepartment of Mathematics
Lanzhou Jiaotong University, Lanzhou 730070, P. R. China

^d Department of Mathematics, Qinghai Normal
University, Xining, Qinghai 810008, P. R. China

lily60612@126.com; lxl@nankai.edu.cn;
liumm05@163.com; maoyaping@ymail.com.

Abstract

The generalized k -connectivity $\kappa_k(G)$ of a graph G was introduced by Chartrand et al. in 1984, which is a nice generalization of the classical connectivity. Recently, as a natural counterpart, Li et al. proposed the concept of generalized edge-connectivity for a graph. In this paper, we consider the computational complexity of the generalized connectivity and generalized edge-connectivity of a graph. We give a confirmative solution to a conjecture raised by S. Li in her Ph.D. thesis “S. Li, Some topics on generalized connectivity of graphs, Ph.D. thesis, Nankai University, 2012”. We also give a polynomial-time algorithm to a problem of generalized k -edge-connectivity.

Keywords: generalized connectivity, generalized edge-connectivity, complexity, polynomial-time algorithm, \mathcal{NP} -complete.

AMS subject classification 2010: 05C40, 05C05, 68Q25, 68R10.

1 Introduction

All graphs considered in this paper are undirected, finite and simple. We refer to the book [1] for graph theoretical notation and terminology not described here. The generalized connectivity of a graph G , introduced by Chartrand et al. in [2], is a

¹Supported by NSFC (Nos. 11071130, 11161037), the “973” program, the Scientific Research Funds of Huaqiao University (No.14BS311), and the Science Found of Qinghai Province (No. 2014-ZJ-907).

natural and nice generalization of the concept of the standard (vertex-)connectivity. For a graph $G(V, E)$ and a set $S \subseteq V(G)$ of at least two vertices, an S -Steiner tree or a Steiner tree connecting S (or simply, an S -tree) is such a subgraph $T(V', E')$ of G that is a tree with $S \subseteq V'$. Two Steiner trees T and T' connecting S are said to be *internally disjoint* if $E(T) \cap E(T') = \emptyset$ and $V(T) \cap V(T') = S$. For $S \subseteq V(G)$ and $|S| \geq 2$, the *generalized local connectivity* $\kappa(S)$ is the maximum number of internally disjoint trees connecting S in G . Note that when $|S| = 2$ a Steiner tree connecting S is just a path connecting S . For an integer k with $2 \leq k \leq n$, the *generalized k -connectivity* $\kappa_k(G)$ of G is defined as $\kappa_k(G) = \min\{\kappa(S) : S \subseteq V(G) \text{ and } |S| = k\}$. Clearly, when $|S| = 2$, $\kappa_2(G)$ is nothing new but the connectivity $\kappa(G)$ of G , that is, $\kappa_2(G) = \kappa(G)$, which is the reason why one addresses $\kappa_k(G)$ as the generalized connectivity of G . Set $\kappa_k(G) = 0$ when G is disconnected. Results on the generalized connectivity can be found in [3, 7, 9–12, 14–18].

As a natural counterpart of the generalized connectivity, Li et al. introduced the concept of generalized edge-connectivity in [13]. For $S \subseteq V(G)$ and $|S| \geq 2$, the *generalized local edge-connectivity* $\lambda(S)$ is the maximum number of edge-disjoint trees connecting S in G . For an integer k with $2 \leq k \leq n$, the *generalized k -edge-connectivity* $\lambda_k(G)$ of G is then defined as $\lambda_k(G) = \min\{\lambda(S) : S \subseteq V(G) \text{ and } |S| = k\}$. It is also clear that when $|S| = 2$, $\lambda_2(G)$ is nothing new but the standard edge-connectivity $\lambda(G)$ of G , that is, $\lambda_2(G) = \lambda(G)$, which is the reason why we address $\lambda_k(G)$ as the generalized edge-connectivity of G . Also set $\lambda_k(G) = 0$ when G is disconnected.

The generalized edge-connectivity is related to an important problem, which is called the *Steiner Tree Packing Problem*. For a given graph G and $S \subseteq V(G)$, this problem asks to find a set of maximum number of edge-disjoint Steiner trees connecting S in G . One can see that the Steiner Tree Packing Problem studies local properties of graphs, but the generalized edge-connectivity focuses on global properties of graphs. The generalized edge-connectivity and the Steiner Tree Packing Problem have applications in *VLSI* circuit design, see [4–6, 19]. In this application, a Steiner tree is needed to share an electronic signal by a set of terminal nodes. Another application, which is our primary focus, arises in the Internet Domain. Imagine that a given graph G represents a network. We choose arbitrary k vertices as nodes. Suppose that one of the nodes in G is a *broadcaster*, and all the other nodes are either *users* or *routers* (also called *switches*). The broadcaster wants to broadcast as many streams of movies as possible, so that the users have the maximum number of choices. Each stream of movie is broadcasted via a tree connecting all the users and the broadcaster. So, in essence we need to find the maximum number Steiner trees connecting all the users and the broadcaster, namely, we want to get $\lambda(S)$, where S is the set of the k nodes. Clearly, it is a Steiner tree packing problem. Furthermore, if we want to know whether for any k nodes the network G has the above properties, then we need to compute $\lambda_k(G)$ in order to prescribe the reliability and the security of the network.

As we know, for any graph G , we have polynomial-time algorithms to get the connectivity $\kappa(G)$ and the edge-connectivity $\lambda(G)$. A natural question is whether there

is a polynomial-time algorithm to get the $\kappa_3(G)$ or $\lambda_3(G)$, or more generally $\kappa_k(G)$ or $\lambda_k(G)$. In [12], the authors described a polynomial-time algorithm to decide whether $\kappa_3(G) \geq \ell$.

Theorem 1.1. [12] *Given a fixed positive integer ℓ , for any graph G the problem of deciding whether $\kappa_3(G) \geq \ell$ can be solved by a polynomial-time algorithm.*

As a continuation of their investigation, S. Li and X. Li later turned their attention to the general κ_k and obtained the following results in [11].

Theorem 1.2. [11] *For two fixed positive integers k and ℓ , given a graph G , a k -subset S of $V(G)$, the problem of deciding whether there are ℓ internally disjoint trees connecting S can be solved by a polynomial-time algorithm.*

Theorem 1.3. [11] *For any fixed integer $k \geq 4$, given a graph G , a k -subset S of $V(G)$ and an integer ℓ ($2 \leq \ell \leq n - 2$), deciding whether there are ℓ internally disjoint trees connecting S , namely deciding whether $\kappa(S) \geq \ell$, is \mathcal{NP} -complete.*

Theorem 1.4. [11] *For any fixed integer $\ell \geq 2$, given a graph G and a subset S of $V(G)$, deciding whether there are ℓ internally disjoint trees connecting S , namely deciding whether $\kappa(S) \geq \ell$, is \mathcal{NP} -complete.*

In Theorem 1.3, for $k = 3$ the complexity is yet not known. So, S. Li in her Ph.D. thesis [8] conjectured that it is \mathcal{NP} -complete.

Conjecture 1.5. [8] *Given a graph G and a 3-subset S of $V(G)$ and an integer ℓ ($2 \leq \ell \leq n - 2$), deciding whether there are ℓ internally disjoint trees connecting S , namely deciding whether $\kappa(S) \geq \ell$, is \mathcal{NP} -complete.*

In this paper, we will confirm that this conjecture is true.

For the generalized k -edge-connectivity, it is also natural to consider its computational complexity problem: for any two positive integers k and ℓ , given a k -subset S of $V(G)$, is there a polynomial-time algorithm to determine whether $\lambda(S) \geq \ell$?

We will show that, if both k and ℓ are fixed integers, there is a polynomial-time algorithm to determine whether $\lambda_k(G) \geq \ell$.

Theorem 1.6. *Given two fixed positive integers k and ℓ , for any graph G the problem of deciding whether $\lambda_k(G) \geq \ell$ can be solved by a polynomial-time algorithm.*

The rest of this paper is organized as follows. In next section we give a confirmative solution to Conjectures 1.5. In section 3, we give the proof of Theorem 1.6.

2 Proof of Conjecture 1.5

In this section, we focus on solving Conjectures 1.5. In order to show that this conjecture is correct, we first introduce a basic \mathcal{NP} -complete problem and a new problem.

3-DIMENSIONAL MATCHING (3-DM): Given three sets U , V and W with $|U| = |V| = |W|$, and a subset T of $U \times V \times W$, decide whether there is a subset M of T with $|M| = |U|$ such that whenever (u, v, w) and (u', v', w') are distinct triples in M , we have $u \neq u'$, $v \neq v'$, and $w \neq w'$?

Problem 1: Given a tripartite graph $G = (V, E)$ with three partitions $(\bar{U}, \bar{V}, \bar{W})$, and $|\bar{U}| = |\bar{V}| = |\bar{W}| = q$, decide whether there is a partition of V into q disjoint 3-sets V_1, V_2, \dots, V_q such that every $V_i = \{v_{i_1}, v_{i_2}, v_{i_3}\}$ satisfies that $v_{i_1} \in \bar{U}$, $v_{i_2} \in \bar{V}$, $v_{i_3} \in \bar{W}$, and $G[V_i]$ is connected?

By reducing 3-DM to Problem 1, we can get the following result.

Lemma 2.1. *Problem 1 is \mathcal{NP} -complete.*

Proof. It is easy to see that Problem 1 is in \mathcal{NP} since given a partition of $V(G)$ into q disjoint 3-sets V_i ($i = 1, 2, \dots, q$), one can check in polynomial time that every $V_i = \{v_{i_1}, v_{i_2}, v_{i_3}\}$ satisfies that $v_{i_1} \in \bar{U}$, $v_{i_2} \in \bar{V}$, $v_{i_3} \in \bar{W}$, and $G[V_i]$ is connected.

We now prove that 3-DM is polynomially reducible to this problem.

Given three sets U , V and W with $|U| = |V| = |W| = n$, and a subset T of $U \times V \times W$. Let $T = \{T_1, T_2, \dots, T_m\}$. We will construct a tripartite graph $G[\bar{U}, \bar{V}, \bar{W}]$ with $|\bar{U}| = |\bar{V}| = |\bar{W}| = q$ such that the desired partition exists for G if and only if there is a subset M of T with $|M| = n$ and whenever (u, v, w) and (u', v', w') are distinct triples in M , we have $u \neq u'$, $v \neq v'$ and $w \neq w'$.

For each $T_i = (u_i, v_i, w_i)$, we add 18 new vertices $V_i^0 = \{t_{i_1}, t_{i_2}, \dots, t_{i_{18}}\}$ and 26 edges E_i^0 which are shown in Figure 1. Thus $G[\bar{U}, \bar{V}, \bar{W}]$ is defined by

$$\bar{U} = U \cup \{t_{i_3}, t_{i_6}, t_{i_7}, t_{i_{10}}, t_{i_{13}}, t_{i_{16}} : 1 \leq i \leq m\}$$

$$\bar{V} = V \cup \{t_{i_1}, t_{i_4}, t_{i_9}, t_{i_{12}}, t_{i_{14}}, t_{i_{17}} : 1 \leq i \leq m\}$$

$$\bar{W} = W \cup \{t_{i_2}, t_{i_5}, t_{i_8}, t_{i_{11}}, t_{i_{15}}, t_{i_{18}} : 1 \leq i \leq m\}$$

$$V = \bar{U} \cup \bar{V} \cup \bar{W}, \quad E = \bigcup_{i=1}^m E_i^0.$$

Note that $|V| = 3n + 18m$, $|E| = 26m$. Thus this instance can be constructed in polynomial time from a 3-DM instance. Now that $q = n + 6m$.

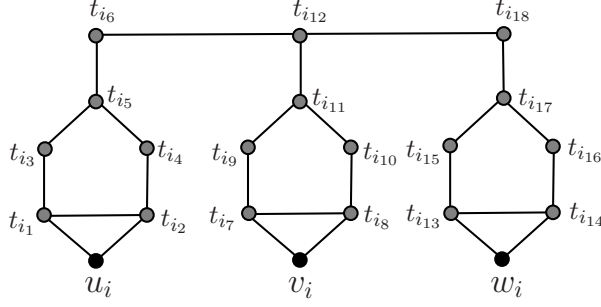


Figure 1. Graphs for Lemma 2.1.

If there is a subset M of T with $|M| = n$, and whenever (u, v, w) and (u', v', w') are distinct triples in M we have $u \neq u'$, $v \neq v'$, and $w \neq w'$, then the corresponding partition $V = V_1 \cup V_2 \cup \dots \cup V_q$ is given by taking $\{u_i, t_{i_1}, t_{i_2}\}$, $\{v_i, t_{i_7}, t_{i_8}\}$, $\{w_i, t_{i_{13}}, t_{i_{14}}\}$, $\{t_{i_3}, t_{i_4}, t_{i_5}\}$, $\{t_{i_9}, t_{i_{10}}, t_{i_{11}}\}$, $\{t_{i_{15}}, t_{i_{16}}, t_{i_{17}}\}$, $\{t_{i_6}, t_{i_{12}}, t_{i_{18}}\}$ from the vertices of $V_i^0 \cup T_i$ whenever $T_i = (u_i, v_i, w_i)$ is in M , and by taking $\{t_{i_1}, t_{i_2}, t_{i_3}\}$, $\{t_{i_4}, t_{i_5}, t_{i_6}\}$, $\{t_{i_7}, t_{i_8}, t_{i_9}\}$, $\{t_{i_{10}}, t_{i_{11}}, t_{i_{12}}\}$, $\{t_{i_{13}}, t_{i_{14}}, t_{i_{15}}\}$, $\{t_{i_{16}}, t_{i_{17}}, t_{i_{18}}\}$ from the vertices of V_i^0 whenever $T_i = (u_i, v_i, w_i)$ is not in M .

Since $|M| = n$, $|T \setminus M| = m - n$, we can find $7n + 6(m - n) = n + 6m = q$ partition sets, each set consists of three vertices which belong to \overline{U} , \overline{V} , \overline{W} , respectively, and they induce a connected subgraph.

Conversely, let V_1, V_2, \dots, V_q be the desired partition of $V(G)$. In the following, we call a 3-set $\{u, v, w\}$ a *partition set*, if there is some j such that $\{u, v, w\} = V_j$. Then we choose $T_i \in M$ if $\{t_{i_6}, t_{i_{12}}, t_{i_{18}}\}$ is a partition set.

Now we claim that $|M| = n$, and whenever (u, v, w) and (u', v', w') are distinct triples in M we have $u \neq u'$, $v \neq v'$ and $w \neq w'$. Indeed, let $T_i = (u_i, v_i, w_i)$. If $\{t_{i_6}, t_{i_{12}}, t_{i_{18}}\}$ is a partition set, then $\{t_{i_3}, t_{i_4}, t_{i_5}\}$ and $\{u_i, t_{i_1}, t_{i_2}\}$, or $\{t_{i_2}, t_{i_4}, t_{i_5}\}$ and $\{u_i, t_{i_1}, t_{i_3}\}$, or $\{t_{i_1}, t_{i_3}, t_{i_5}\}$ and $\{u_i, t_{i_2}, t_{i_4}\}$ are the partition sets. In either cases, u_i must belong to a partition set with the other two elements belong to V_i^0 . Similar thing is true for v_i and w_i . If $\{t_{i_6}, t_{i_{12}}, t_{i_{18}}\}$ is not a partition set, then $\{t_{i_6}, t_{i_{11}}, t_{i_{12}}\}$ or $\{t_{i_4}, t_{i_5}, t_{i_6}\}$ is a partition set. But $\{t_{i_6}, t_{i_{11}}, t_{i_{12}}\}$ can not be a partition set. If so, then $\{t_{i_8}, t_{i_{10}}, v_i\}$, $\{t_{i_7}, t_{i_8}, t_{i_9}\}$ or $\{t_{i_7}, t_{i_8}, v_i\}$ must be a partition set, and no matter in which cases, t_{i_9} or $t_{i_{10}}$ can not be in a partition set. Thus only $\{t_{i_4}, t_{i_5}, t_{i_6}\}$ is a partition set. Similarly, $\{t_{i_1}, t_{i_2}, t_{i_3}\}$, $\{t_{i_7}, t_{i_8}, t_{i_9}\}$, $\{t_{i_{10}}, t_{i_{11}}, t_{i_{12}}\}$, $\{t_{i_{13}}, t_{i_{14}}, t_{i_{15}}\}$, $\{t_{i_{16}}, t_{i_{17}}, t_{i_{18}}\}$ must be partition sets. Then u_i, v_i, w_i can not be in any partition sets with some vertices in V_i^0 .

If $T_i = (u_i, v_i, w_i)$ and $T_j = (u_j, v_j, w_j)$ are distinct triples in M , then u_i is in a partition set with the other two elements in V_i^0 , and u_j is in a partition set with the other two elements in V_j^0 , and thus $u_i \neq u_j$, and so do $v_i \neq v_j$ and $w_i \neq w_j$. Assume that $|M| = \ell$. If $T_i \in M$, then there are 7 partition sets in $V_i^0 \cup T_i$. If $T_i \notin M$, then there are 6 partition sets in V_i^0 . Since $7\ell + 6(m - \ell) = q = n + 6m$, $\ell = n$, that is $|M| = n$. The proof is now complete. \square

Now we prove that Conjecture 1.5 is true by reducing Problem 1 to it.

Theorem 2.2. *Given a graph G , a 3-subset S of $V(G)$ and an integer ℓ ($2 \leq \ell \leq n-2$), the problem of deciding whether G contains ℓ internally disjoint trees connecting S is \mathcal{NP} -complete.*

Proof. It is easy to see that this problem is in \mathcal{NP} .

Let G be a tripartite graph with partition $(\bar{U}, \bar{V}, \bar{W})$ and $|\bar{U}| = |\bar{V}| = |\bar{W}| = q$. We will construct a graph G' , and a 3-subset S and an integer ℓ such that there are ℓ internally disjoint trees connecting S in G' if and only if G contains a partition of $V(G)$ into q disjoint sets V_1, V_2, \dots, V_q each having three vertices, such that every $V_i = \{v_{i_1}, v_{i_2}, v_{i_3}\}$ satisfies that $v_{i_1} \in \bar{U}$, $v_{i_2} \in \bar{V}$, $v_{i_3} \in \bar{W}$, and $G[V_i]$ is connected.

We define G' as follows:

- $V(G') = V(G) \cup \{a, b, c\}$;
- $E(G') = E(G) \cup \{au : u \in \bar{U}\} \cup \{bv : v \in \bar{V}\} \cup \{cw : w \in \bar{W}\}$.

Then we define $S = \{a, b, c\}$, and $\ell = q$.

If there are q internally disjoint trees connecting S in G' , then, since a, b and c all have degree q , each tree contains a vertex from \bar{U} , a vertex from \bar{V} and a vertex from \bar{W} , and they induce a connected subgraph. Since these q trees are internally disjoint, they form a partition of $V(G)$.

Conversely, if V_1, V_2, \dots, V_q is a partition of $V(G)$ each having three vertices, such that every $V_i = \{v_{i_1}, v_{i_2}, v_{i_3}\}$ satisfies that $v_{i_1} \in \bar{U}$, $v_{i_2} \in \bar{V}$, $v_{i_3} \in \bar{W}$, and $G[V_i]$ is connected, then let T_i be the spanning tree of $G[V_i]$ together with the edges $av_{i_1}, bv_{i_2}, cv_{i_3}$, where $V_i = \{v_{i_1}, v_{i_2}, v_{i_3}\}$. It is easy to check that T_1, T_2, \dots, T_q are the desired internally disjoint trees connecting S .

This completes the proof. □

3 Proof of Theorem 1.6

In this section we give a proof of Theorem 1.6.

Proof. Given a connected graph G of order n and a k -subset S of $V(G)$. Let $V(G) = \{v_1, v_2, \dots, v_n\}$ and $E(G) = \{e_1, e_2, \dots, e_m\}$. We will construct a graph G' and a k -subset S' of $V(G')$ such that there are ℓ edge-disjoint trees connecting S in G if and only if there are ℓ internally disjoint trees connecting S' in G' .

We define G' as follows:

- $V(G') = V(G) \cup V(L(G)) = \{v_1, v_2, \dots, v_n, e_1, e_2, \dots, e_m\}$;

• $E(G') = \{e_i e_j : e_i e_j \in E(L(G))\} \cup \{v_i e_j : e_j \text{ is incident to } v_i \text{ in } G\}$;
 where $L(G)$ is the line graph of G . We define $S' = S$.

If there are ℓ edge-disjoint trees connecting S in G , say T_1, T_2, \dots, T_ℓ . First for each tree T_i ($1 \leq i \leq \ell$), we replace every edge $e_j = v_{j_1} v_{j_2}$ by a path $v_{j_1} e_j v_{j_2}$. The obtained graph T'_i now is a tree in G' . Clearly, the trees $T'_1, T'_2, \dots, T'_\ell$ are ℓ edge-disjoint trees connecting $S' = S$ in G' . Consider the tree T'_i . If there is a vertex $v \in V(T'_i) \setminus S$ such that its neighbors in T'_i are $e_{i_1}, e_{i_2}, \dots, e_{i_p}$, then we delete the vertex v and its incident edges $ve_{i_1}, ve_{i_2}, \dots, ve_{i_p}$, add a path $e_{i_1} e_{i_2} \dots e_{i_p}$. We do this operation for all the vertex $v \in V(T'_i) \setminus S$ for $1 \leq i \leq \ell$. The resulting trees are denoted by $T''_1, T''_2, \dots, T''_\ell$. It is easy to check that they are internally disjoint trees connecting S' in G' .

Conversely, if there are ℓ internally disjoint trees T_1, T_2, \dots, T_ℓ connecting S' in G' . Consider any tree T_i ($1 \leq i \leq \ell$). If there is an edge $e_{j_1} e_{j_2}$ in $E(T_i)$, by the definition of $E(G')$, e_{j_1} and e_{j_2} are adjacent in G and hence they have a common vertex v_j in G . Note that $v_j e_{j_1}, v_j e_{j_2}$ are also edges of G' . Thus we replace the edge $e_{j_1} e_{j_2}$ by a path $e_{j_1} v_j e_{j_2}$. We do this for all the edges of this type in T_i . The resulting connected graph is denoted by G_i . Now there is no such edge $e_i e_j$ in G_i and $d_{G_i}(e) \leq 2$. If $d_{G_i}(e) = 1$, we just delete it. If $d_{G_i}(e) = 2$, there are two vertices v_{j_1} and v_{j_2} adjacent with e , where v_{j_1} and v_{j_2} are the endpoints of e in G , so we delete the vertex e and add an edge $v_{j_1} v_{j_2}$, then the obtained graph G'_i is a connected graph with $S \subseteq V(G'_i)$. Let T'_i be the spanning tree of G'_i . It is easy to check that $T'_1, T'_2, \dots, T'_\ell$ are ℓ edge-disjoint trees connecting S in G .

From the above reduction, if we want to know whether there are ℓ edge-disjoint trees connecting S in G , we can construct a graph G' , and decide whether there are ℓ internally disjoint trees connecting S' in G' . By Theorem 1.2, since k and ℓ are fixed, the problem of deciding whether there are ℓ internally disjoint trees connecting S can be solved by a polynomial-time algorithm. Therefore, the problem of deciding whether there are ℓ edge-disjoint trees connecting S can be solved by a polynomial-time algorithm. Since $\lambda_k(G) = \min\{\lambda(S) : S \subseteq V(G) \text{ and } |S| = k\}$, we need to check $\binom{n}{k}$ subsets S of $V(G)$. And because k is fixed, the problem of deciding whether $\lambda_k(G) \geq \ell$ can be solved by a polynomial-time algorithm. The proof is complete. \square

4 Concluding remarks

In this paper, we give a confirmative solution to a conjecture raised by S. Li in her Ph.D. thesis. We also consider the computational complexity problem of generalized k -edge-connectivity: for any two positive integers k and ℓ , given a k -subset S of $V(G)$, is there a polynomial-time algorithm to determine whether $\lambda(S) \geq \ell$?

We show that, if both k and ℓ are fixed integers, there is a polynomial-time algorithm to determine whether $\lambda_k(G) \geq \ell$. But when k or ℓ is not a fixed integer, the complexity of this problem is not known. We conjecture that it is \mathcal{NP} -complete.

Conjecture 4.1. For any fixed integer $k \geq 3$, given a graph G , a k -subset S of $V(G)$, and an integer ℓ ($2 \leq \ell \leq n - 2$), deciding whether there are ℓ edge-disjoint trees connecting S , namely deciding whether $\lambda(S) \geq \ell$, is \mathcal{NP} -complete.

Conjecture 4.2. For any fixed integer $\ell \geq 2$, given a graph G , a subset S of $V(G)$, deciding whether there are ℓ edge-disjoint trees connecting S , namely deciding whether $\lambda(S) \geq \ell$, is \mathcal{NP} -complete.

References

- [1] J.A. Bondy, U.S.R. Murty, Graph Theory, GTM 244, Springer, 2008.
- [2] G. Chartrand, S. Kappor, L. Lesniak, D. Lick, *Generalized connectivity in graphs*, Bull. Bombay Math. Colloq. 2(1984), 1-6.
- [3] G. Chartrand, F. Okamoto, P. Zhang, *Rainbow trees in graphs and generalized connectivity*, Networks 55(4)(2010), 360-367.
- [4] M. Grötschel, *The Steiner tree packing problem in VLSI design*, Math. Program. 78(1997), 265-281.
- [5] M. Grötschel, A. Martin, R. Weismantel, *Packing Steiner trees: A cutting plane algorithm and computational results*, Math. Program. 72(1996), 125-145.
- [6] K. Jain, M. Mahdian, M. Salavatipour, *Packing Steiner trees*, in: Proc. of the 14th ACM-SIAM symposium on Discrete Algorithms, Baltimore, 2003, 266-274.
- [7] H. Li, X. Li, Y. Sun, *The generalied 3-connectivity of Cartesian product graphs*, Discrete Math. Theor. Comput. Sci. 14(1)(2012), 43-54.
- [8] S. Li, *Some topics on generalized connectivity of graphs*, Ph.D. thesis, Nankai University, 2012.
- [9] S. Li, W. Li, X. Li, *The generalized connectivity of complete bipartite graphs*, Ars Combin. 104(2012), 65-79.
- [10] S. Li, W. Li, X. Li, *The generalized connectivity of complete equipartition 3-partite graphs*, Bull. Malays. Math. Sci. Soc. (2)37(1)(2014), 103-121.
- [11] S. Li, X. Li, *Note on the hardness of generalized connectivity*, J. Combin. Optimization 24(2012), 389-396.
- [12] S. Li, X. Li, W. Zhou, *Sharp bounds for the generalized connectivity $\kappa_3(G)$* , Discrete Math. 310(2010), 2147-2163.
- [13] X. Li, Y. Mao, Y. Sun, *On the generalized (edge-)connectivity of graphs*, Australasian J. Combin. 58(2)(2014), 304-319.
- [14] X. Li, Y. Mao, *On extremal graphs with at most ℓ internally disjoint Steiner trees connecting any $n - 1$ vertices*, accepted by Graphs & Combin.

- [15] X. Li, Y. Mao, *The generalied 3-connectivity of lexicographical product graphs*, Discrete Math. Theor. Comput. Sci. 16(1)(2014), 339–354.
- [16] X. Li, Y. Mao, *Nordhaus-Gaddum-type results for the generalized edge-connectivity of graphs*, Discrete Appl. Math. 185(2015), 102–112.
- [17] X. Li, Y. Mao, *The minimal size of a graph with given generalized 3-edge-connectivity*, Ars Combin. 118(2015), 63–72.
- [18] F. Okamoto, P. Zhang, *The tree connectivity of regular complete bipartite graphs*, J. Combin. Math. Combin. Comput. 74(2010), 279-293.
- [19] N.A. Sherwani, Algorithms for VLSI physical design automation, 3rd Edition, Kluwer Acad. Pub., London, 1999.