Note on the hardness of generalized connectivity*

Shasha Li, Xueliang Li

Center for Combinatorics and LPMC-TJKLC

Nankai University, Tianjin 300071, China.

Email: lss@cfc.nankai.edu.cn, lxl@nankai.edu.cn

Abstract

Let G be a nontrivial connected graph of order n and let k be an integer with $2 \leq k \leq n$. For a set S of k vertices of G, let $\kappa(S)$ denote the maximum number ℓ of edge-disjoint trees $T_1, T_2, \ldots, T_\ell$ in G such that $V(T_i) \cap V(T_j) = S$ for every pair i, j of distinct integers with $1 \leq i, j \leq \ell$. Chartrand et al. generalized the concept of connectivity as follows: The *k-connectivity*, denoted by $\kappa_k(G)$, of G is defined by $\kappa_k(G)$ =min$\{\kappa(S)\}$, where the minimum is taken over all k-subsets S of $V(G)$. Thus $\kappa_2(G) = \kappa(G)$, where $\kappa(G)$ is the connectivity of G, for which there are polynomial-time algorithms to solve it.

This paper mainly focus on the complexity of determining the generalized connectivity of a graph. At first, we obtain that for two fixed positive integers k_1 and k_2, given a graph G and a k_1-subset S of $V(G)$, the problem of deciding whether G contains k_2 internally disjoint trees connecting S can be solved by a polynomial-time algorithm. Then, we show that when k_1 is a fixed integer of at least 4, but k_2 is not a fixed integer, the problem turns out to be NP-complete. On the other hand, when k_2 is a fixed integer of at least 2, but k_1 is not a fixed integer, we show that the problem also becomes NP-complete.

Keywords: k-connectivity, internally disjoint trees, complexity, polynomial-time, NP-complete

AMS Subject Classification 2010: 05C40, 05C05, 68Q25, 68R10.

*Supported by NSFC No.11071130.

1 Introduction

We follow the terminology and notation of [1] and all graphs considered here are always simple. As usual, the subgraph of G whose vertex set is X and whose edge set is the set of those edges of G that have both ends in X is called the subgraph of G induced by X and is denoted by $G[X]$. For $X = \{x_1, x_2, \ldots, x_k\}$ and $Y = \{y_1, y_2, \ldots, y_k\}$, an *$XY$-linkage* is defined as a set of k vertex-disjoint paths $x_iP_iy_i$, $1 \leq i \leq k$. The *linkage problem* is the problem of deciding whether there exists an XY-linkage for given sets X and Y. The *connectivity* $\kappa(G)$ of a graph G is defined as the minimum cardinality of a set Q of vertices of G such that $G - Q$ is disconnected or trivial. A well-known theorem of Whitney [8] provides an equivalent definition of connectivity. For each 2-subset $S = \{u, v\}$ of vertices of G, let $\kappa(S)$ denote the maximum number of internally disjoint uv-paths in G. Then $\kappa(G)$ =min$\{\kappa(S)\}$, where the minimum is taken over all 2-subsets S of $V(G)$.

In [2], the authors generalized the concept of connectivity. Let G be a nontrivial connected graph of order n and let k be an integer with $2 \leq k \leq n$. For a set S of k vertices of G, let $\kappa(S)$ denote the maximum number ℓ of edge-disjoint trees $T_1, T_2, \ldots, T_\ell$ in G such that $V(T_i) \cap V(T_j) = S$ for every pair i, j of distinct integers with $1 \leq i, j \leq \ell$ (Note that the trees are vertex-disjoint in $G \backslash S$). A collection $\{T_1, T_2, \ldots, T_\ell\}$ of trees in G with this property is called an *internally disjoint set of trees connecting S*. The *k-connectivity*, denoted by $\kappa_k(G)$, of G is then defined by $\kappa_k(G)$ =min$\{\kappa(S)\}$, where the minimum is taken over all k-subsets S of $V(G)$. Thus, $\kappa_2(G) = \kappa(G)$. Moreover, $\kappa_n(G)$ is the maximum number of edge-disjoint spanning trees of G.

In addition to being a natural combinatorial measure, generalized connectivity can be motivated by its interesting interpretation in practice. For example, suppose that G represents a network. If one considers to connect a pair of vertices of G, then a path is used to connect them. However, if one wants to connect a set S of vertices of G with $|S| \geq 3$, then a tree must be used to connect them. This kind of tree for connecting a set of vertices is usually called a Steiner tree, and popularly used in the physical design of VLSI, see [7]. Usually, one wants to consider how tough a network can be, for the connection of a set of vertices. Then, the number of totally independent ways to connect them is a measure for this purpose. The generalized k-connectivity can serve for measuring the capability of a network G to connect any k vertices in G.

In [2], Chartrand et al. obtained the exact value of κ_k for a complete graph. Okamoto and Zhang in [5] investigated the generalized connectivity for a regular complete bipartite graph $K_{a,a}$. Recently, Li et al. [3] got the exact value of κ_k for a general complete bipartite graph $K_{a,b}$. But, for a general graph G and a general positive integer k, to get the exact value of $\kappa_k(G)$ is very difficult. In [4], we focused on the investigation of $\kappa_3(G)$ and mainly studied the relationship between the 2-connectivity and the 3-connectivity of a graph. We gave sharp upper and lower bounds for $\kappa_3(G)$ for general graphs G, and constructed two kinds of graphs which attain the upper and lower bounds, respectively. We also showed that if G is a connected planar graph, then $\kappa(G)-1 \leq \kappa_3(G) \leq \kappa(G)$, and gave some classes of graphs which attain the bounds. Moreover, we studied algorithmic aspects for $\kappa_3(G)$ and gave an algorithm to determine $\kappa_3(G)$ for a general graph G. This algorithm runs in polynomial time for graphs with a fixed value of connectivity, which implies that the problem of determining $\kappa_3(G)$ for graphs with small minimum degree or small connectivity can be solved in polynomial time, in particular, the problem whether $\kappa(G) = \kappa_3(G)$ for a planar graph G can be solved in polynomial time.

By the definition of $\kappa_k(G)$, it is natural to study $\kappa(S)$ first, where S is a k-subset of $V(G)$. A question is then raised: for any two positive integers k_1 and k_2, given a k_1- subset S of $V(G)$, is there a polynomial-time algorithm to determine whether $\kappa(S) \geq k_2$? In this paper, we mainly focus on this problem. At first, by generalizing the algorithm of [4], we obtain that if k_1 and k_2 are two fixed positive integers, given a graph G and a k_1-subset S of $V(G)$, the problem of deciding whether G contains k_2 internally disjoint trees connecting S can be solved by a polynomial-time algorithm. But when k_1 is a fixed integer of at least 4, and k_2 is not a fixed integer, we show that the problem turns out to be NP-complete.

Theorem 1.1. *For any fixed integer $k_1 \geq 4$, given a graph G, a k_1-subset S of $V(G)$ and an integer $2 \leq k_2 \leq n-1$, deciding whether there are k_2 internally disjoint trees connecting S, namely deciding whether $\kappa(S) \geq k_2$, is NP-complete.*

On the other hand, when k_2 is a fixed integer of at least 2, but k_1 is not a fixed integer, we show that the problem also becomes NP-complete.

Theorem 1.2. *For any fixed integer $k \geq 2$, given a graph G and a subset S of $V(G)$, deciding whether there are k internally disjoint trees connecting S, namely deciding whether $\kappa(S) \geq k$, is NP-complete.*

The rest of this paper is organized as follows. The next section simply generalizes the algorithm of [4] and makes some preparations. Sections 3 and 4 prove Theorem 1.1 and Theorem 1.2, respectively.

2 Preliminaries

At first, we introduce the following result of [4].

Lemma 2.1. *Given a fixed positive integer k, for any graph G the problem of deciding whether G contains k internally disjoint trees connecting $\{v_1, v_2, v_3\}$ can be solved by a polynomial-time algorithm, where v_1, v_2, v_3 are any three vertices of $V(G)$.*

In [4], we first showed that the trees we really want have only two types. Then we proved that if there are k internally disjoint trees connecting $\{v_1, v_2, v_3\}$, then the union of the k trees has at most $f(k)n^k$ types, where $f(k)$ is a function on k. For every $i \in [f(k)n^k]$, we can convert the problem of deciding whether G contains a union of k trees of type i into a k'-linkage problem. Since the k'-linkage problem can be solved by an algorithm with running time $O(n^3)$, see [6], and since k is a fixed integer, we finally obtain that the problem of deciding whether $\kappa\{v_1, v_2, v_3\} \geq k$ can be solved by a polynomial-time algorithm. We refer the readers to [4] for details.

By the similar method, we can also show that given a fixed positive integer k, for any graph G the problem of deciding whether G contains k internally disjoint trees connecting $\{v_1, v_2, v_3, v_4\}$ can be solved by a polynomial-time algorithm, where v_1, v_2, v_3, v_4 are any four vertices of $V(G)$.

For the tree T connecting $\{v_1, v_2, v_3, v_4\}$, we only need T belonging to one of the five types in Figure 1. Then if there are k internally disjoint trees connecting $\{v_1, v_2, v_3, v_4\}$, consider the union of the k trees and it is not hard to obtain that the number of types is at most $f(k)n^{2k}$, where $f(k)$ is a function on k and $f(k)n^{2k}$ is only a rough upper bound. Now for every $i \in [f(k)n^{2k}]$, we can convert the problem of deciding whether G contains a union of k trees of type i into a k'-linkage problem. Since the k'-linkage problem has a polynomial-time algorithm and since k is a fixed integer, we obtain that the problem of deciding whether $\kappa\{v_1, v_2, v_3, v_4\} \geq k$ can be solved by a polynomial-time algorithm.

Now, for two fixed positive integers k_1 and k_2, if we replace the set $\{v_1, v_2, v_3, v_4\}$ with a k_1-subset S of $V(G)$ and replace k with k_2, the problem can still be solved by a polynomial-time algorithm. The method is similar.

For the tree T connecting the k_1-subset S of $V(G)$, the number of types of T we really want is at most $f_1(k_1)$, where $f_1(k_1)$ is a function on k_1. Then if there are k_2 internally disjoint trees connecting S, consider the union of the k_2 trees and it is not hard to obtain that the number of types is at most $f_2(k_1, k_2)n^{(k_1-2)k_2}$, where $f_2(k_1, k_2)$ is a function on k_1 and k_2 and $f_2(k_1, k_2)n^{(k_1-2)k_2}$ is only a rough upper bound. Next, by the same way, for every

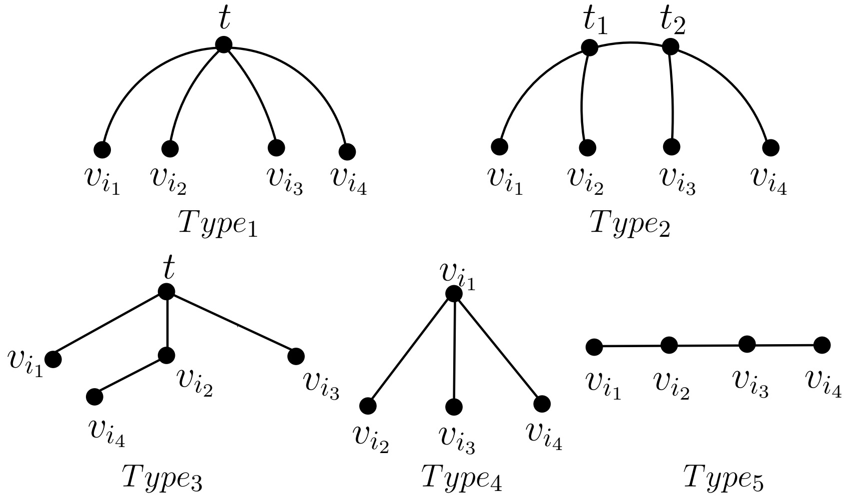

Figure 1: Five types of trees we really want, where $\{v_{i_1}, v_{i_2}, v_{i_3}, v_{i_4}\} = \{v_1, v_2, v_3, v_4\}$.

$i \in [f_2(k_1, k_2)n^{(k_1-2)k_2}]$, convert the problem of deciding whether G contains a union of k_2 trees of type i into a k'-linkage problem and a polynomial-time algorithm is then obtained. Therefore, we have the following lemma.

Lemma 2.2. *For two fixed positive integers k_1 and k_2, given a graph G and a k_1-subset S of $V(G)$, the problem of deciding whether G contains k_2 internally disjoint trees connecting S can be solved by a polynomial-time algorithm.*

Note that Lemma 2.2 is a generalization of Lemma 2.1. When $k_1 = 3$ and $k_2 = k$, Lemma 2.2 is exactly Lemma 2.1.

Before proceeding, we recall the following two basic NP-complete problems.

3-DIMENSIONAL MATCHING (3-DM)

Given three sets U, V, and W of equal cardinality, 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, $u \neq u'$, $v \neq v'$, and $w \neq w'$?

BOOLEAN 3-SATISFIABILITY (3-SAT)

Given a boolean formula ϕ in conjunctive normal form with three literals per clause, decide whether ϕ is satisfiable ?

3 Proof of Theorem 1.1

For the problem in Lemma 2.2, when $k_1 = 4$ and k_2 is not a fixed integer, we denote this case by Problem 1.

Problem 1. Given a graph G, a 4-subset S of $V(G)$ and an integer $2 \leq k \leq n-1$, decide whether there are k internally disjoint trees connecting S, namely decide whether $\kappa(S) \geq k$?

At first, we will show that Problem 1 is NP-complete by reducing 3-DM to it, as follows.

Lemma 3.1. *Given a graph G, a 4-subset S of $V(G)$ and an integer $2 \leq k \leq n-1$, deciding whether there are k internally disjoint trees connecting S, namely deciding whether $\kappa(S) \geq k$, is NP-complete.*

Proof. It is clear that Problem 1 is in NP. So it suffices to show that 3-DM is polynomially reducible to this problem.

Given three sets of equal cardinality, denoted by $U = \{u_1, u_2, \ldots, u_n\}$, $V = \{v_1, v_2, \ldots, v_n\}$ and $W = \{w_1, w_2, \ldots, w_n\}$, and a subset $T = \{T_1, T_2, \ldots, T_m\}$ of $U \times V \times W$, we will construct a graph G', a 4-subset S of $V(G')$ and an integer $k \leq |V(G')| - 1$ such that there are k internally disjoint trees connecting S in G' if and only if there is a subset M of T with $|M| = |U| = n$ such that whenever (u_i, v_j, w_k) and $(u_{i'}, v_{j'}, w_{k'})$ are distinct triples in M, $u_i \neq u_{i'}$, $v_j \neq v_{j'}$ and $w_k \neq w_{k'}$.

We define G' as follows:

$$
\begin{aligned}
V(G') = {} & \{\hat{u}, \hat{v}, \hat{w}, \hat{t}\} \cup \{u_i : 1 \leq i \leq n\} \cup \{v_i : 1 \leq i \leq n\} \\
& \cup \{w_i : 1 \leq i \leq n\} \cup \{t_i : 1 \leq i \leq m\} \cup \{a_i : 1 \leq i \leq m-n\}; \\
E(G') = {} & \{\hat{u}u_i : 1 \leq i \leq n\} \cup \{\hat{v}v_i : 1 \leq i \leq n\} \cup \{\hat{w}w_i : 1 \leq i \leq n\} \\
& \cup \{\hat{t}t_i : 1 \leq i \leq m\} \cup \{\hat{u}a_i : 1 \leq i \leq m-n\} \cup \{\hat{v}a_i : 1 \leq i \leq m-n\} \\
& \cup \{\hat{w}a_i : 1 \leq i \leq m-n\} \cup \{t_i a_j : 1 \leq i \leq m, 1 \leq j \leq m-n\} \\
& \cup \{t_i u_j : u_j \in T_i\} \cup \{t_i v_j : v_j \in T_i\} \cup \{t_i w_j : w_j \in T_i\}.
\end{aligned}
$$

Each vertex t_i corresponds to the triple T_i, where $1 \leq i \leq m$. Note that $|V(G')| = 2n + 2m + 4$ and $|E(G')| = m(7 + m - n)$. Now let $S = \{\hat{u}, \hat{v}, \hat{w}, \hat{t}\}$ and $k = m$.

Suppose that there is a subset M of T with $|M| = |U| = n$ such that whenever (u_i, v_j, w_k) and $(u_{i'}, v_{j'}, w_{k'})$ are distinct triples in M, $u_i \neq u_{i'}$, $v_j \neq v_{j'}$ and $w_k \neq w_{k'}$. Then for every $T_i \in M$, we can construct a tree whose vertex set consists of S, t_i and three vertices corresponding to three elements in T_i. For each $T_i \notin M$, $G[t_i, a_j, \hat{u}, \hat{v}, \hat{w}, \hat{t}]$ is a tree connecting S, for some $1 \leq j \leq m-n$. So we can easily find out k internally disjoint trees connecting S in G'.

Now suppose that there are $k = m$ internally disjoint trees connecting S in G'. Since $\hat{u}, \hat{v}, \hat{w}$ and $\hat{t}$ all have degree m, then among the m trees, there are n trees, each of which contains the vertices in S, a vertex from $\{t_i : 1 \leq i \leq m\}$, a vertex from $\{u_i : 1 \leq i \leq n\}$, a vertex from $\{v_i : 1 \leq i \leq n\}$ and a vertex from $\{w_i : 1 \leq i \leq n\}$ and can not contain any other vertex. Since the n trees are internally disjoint, it can be easily checked that n triples $T_i \in U \times V \times W$ corresponding to n vertices t_i in the n trees form a subset M of T with $|M| = |U| = n$ such that whenever (u_i, v_j, w_k) and $(u_{i'}, v_{j'}, w_{k'})$ are distinct triples in M, $u_i \neq u_{i'}$, $v_j \neq v_{j'}$ and $w_k \neq w_{k'}$. The proof is complete. ∎

Now we show that for a fixed integer $k_1 \geq 5$, in Problem 1 replacing the 4-subset of $V(G)$ with a k_1-subset of $V(G)$, the problem is still NP-complete, which can easily be proved by reducing Problem 1 to it.

Lemma 3.2. *For any fixed integer $k_1 \geq 5$, given a graph G, a k_1-subset S of $V(G)$ and an integer $2 \leq k_2 \leq n-1$, deciding whether there are k_2 internally disjoint trees connecting S, namely deciding whether $\kappa(S) \geq k_2$, is NP-complete.*

Proof. Clearly, the problem is in NP. We will prove that Problem 1 is polynomially reducible to it.

For any given graph G, a 4-subset $S = \{v_1, v_2, v_3, v_4\}$ of $V(G)$ and an integer $2 \leq k \leq n-1$, we construct a new graph $G' = (V', E')$ and a k_1-subset S' of $V(G')$ and let $k_2 = k$ such that there are $k_2 = k$ internally disjoint trees connecting S' in G' if and only if there are k internally disjoint trees connecting S in G.

We construct $G' = (V', E')$ by adding $k_1 - 4$ new vertices $\{\hat{a}^1, \hat{a}^2, \ldots, \hat{a}^{k_1-4}\}$ to G and for every $i \leq k_1 - 4$, adding k_2 internally disjoint $\hat{a}^i v_1$-paths $\{\hat{a}^i a^i_j v_1 : 1 \leq j \leq k_2\}$ of length two, where a^i_j is also a new vertex and if $i_1 \neq i_2$, $a^{i_1}_{j_1} \neq a^{i_2}_{j_2}$. We have $|V(G')| = (k_1 - 4)(1 + k_2) + n$ and $|E(G')| = 2k_2(k_1 - 4) + m$. Now let $S' = \{v_1, v_2, v_3, v_4, \hat{a}^1, \hat{a}^2, \ldots, \hat{a}^{k_1-4}\}$. It is not hard to check that $\kappa_{G'}(S') \geq k_2 = k$ if and only if $\kappa_G(S) \geq k$. The proof is complete. ∎

Combining Lemma 3.1 with Lemma 3.2, we obtain Theorem 1.1, namely, we complete the proof of Theorem 1.1.

4 Proof of Theorem 1.2

For the problem in Lemma 2.2, when $k_2 = 2$ and k_1 is not a fixed integer, we denote this case by Problem 2.

Problem 2. Given a graph G and a subset S of $V(G)$, decide whether there are two internally disjoint trees connecting S, namely decide whether $\kappa(S) \geq 2$?

Firstly, the following lemma proves that Problem 2 is NP-complete by reducing 3-SAT to it.

Lemma 4.1. *Given a graph G and a subset S of $V(G)$, deciding whether there are two internally disjoint trees connecting S, namely deciding whether $\kappa(S) \geq 2$, is NP-complete.*

Proof. Clearly, Problem 2 is in NP. So it suffices to show that 3-SAT is polynomially reducible to this problem.

Given a 3-CNF formula $\phi = \bigwedge_{i=1}^{m} c_i$ over variables $x_1, x_2, \ldots, x_n$, we construct a graph G_ϕ and a subset S of $V(G_\phi)$ such that there are two internally disjoint trees connecting S if and only if ϕ is satisfiable.

We define G_ϕ as follows:

$$\begin{aligned} V(G_\phi) &= \{\hat{x}_i : 1 \leq i \leq n\} \cup \{x_i : 1 \leq i \leq n\} \cup \{\ \bar{x}_i : 1 \leq i \leq n\} \\ &\quad \cup \{c_i : 1 \leq i \leq m\} \cup \{a\}; \\ E(G_\phi) &= \{\hat{x}_i x_i : 1 \leq i \leq n\} \cup \{\hat{x}_i \bar{x}_i : 1 \leq i \leq n\} \\ &\quad \cup \{x_i c_j : x_i \in c_j\} \cup \{\bar{x}_i c_j : \bar{x}_i \in c_j\} \\ &\quad \cup \{x_1 x_i : 2 \leq i \leq n\} \cup \{x_1 \bar{x}_i : 2 \leq i \leq n\} \cup \{\bar{x}_1 x_i : 2 \leq i \leq n\} \cup \{\bar{x}_1 \bar{x}_i : 2 \leq i \leq n\} \\ &\quad \cup \{a x_i : 1 \leq i \leq n\} \cup \{a \bar{x}_i : 1 \leq i \leq n\} \cup \{a c_i : 1 \leq i \leq m\}, \end{aligned}$$

where the notation $x_i \in c_j (\bar{x}_i \in c_j)$ signifies that $x_i (\bar{x}_i)$ is a literal of the clause c_j. Note that $|V(G')| = 3n + m + 1$ and $|E(G')| = 4n + 4m + 4(n-1)$. Now let $S = \{\hat{x}_i : 1 \leq i \leq n\} \cup \{c_i : 1 \leq i \leq m\}$.

Suppose that there is a true assignment t satisfying ϕ. Then for every clause $c_i (1 \leq i \leq m)$, there must exist a literal $x_j \in c_i$ such that $t(x_j) = 1$ or $\bar{x}_j \in c_i$ such that $t(x_j) = 0$, for some $1 \leq j \leq n$. For such literals x_j or $\bar{x}_j$, let T_1 be a graph such that $E(T_1) = \{c_i x_j$ (or $c_i \bar{x}_j) : 1 \leq i \leq m\}$. Obviously, at most one of the two vertices x_j and $\bar{x}_j$ exists in $V(T_1)$. If neither x_j nor $\bar{x}_j$ is in $V(T_1)$, we can add any one of them to $V(T_1)$. Now, if $x_1 \in V(T_1)$, add $x_1 x_i$(if $x_i \in V(T_1)$) or $x_1 \bar{x}_i$(if $\bar{x}_i \in V(T_1)$) to $E(T_1)$, for $2 \leq i \leq n$. Otherwise, add $\bar{x}_1 x_i$(if $x_i \in V(T_1)$) or $\bar{x}_1 \bar{x}_i$(if $\bar{x}_i \in V(T_1)$) to $E(T_1)$. Finally, add edges $x_i \hat{x}_i$(if $x_i \in V(T_1)$) or $\bar{x}_i \hat{x}_i$(if

$\bar{x}_i \in V(T_1)$) to $E(T_1)$, for $1 \le i \le n$. Now it is easy to check that T_1 is a tree connecting S. Then let T_2 be a tree containing ac_i for $1 \le i \le m$, ax_j and $x_j\hat{x}_j$(if $\bar{x}_j \in V(T_1)$) or $a\bar{x}_j$ and $\bar{x}_j\hat{x}_j$(if $x_j \in V(T_1)$) for $1 \le j \le n$. T_1 and T_2 are two internally disjoint trees connecting S.

Now suppose that there are two internally disjoint trees T_1, T_2 connecting S. Since $a \notin S$, only one tree can contain the vertex a. Without loss of generality, assume that $a \notin V(T_1)$. Since for every $1 \le i \le n$, $\hat{x}_i \in S$ has degree two, $V(T_1)$ must contain one and only one of its two neighbors x_i and $\bar{x}_i$. Then let the value of a variable x_i be 1 if its corresponding vertex x_i is contained in $V(T_1)$. Otherwise let the value be 0. Moreover, because $a \notin V(T_1)$, for every $c_i (1 \le i \le m)$, there must exist some vertex $x_j \in V(T_1)$ such that $c_ix_j \in E(T_1)$ or $\bar{x}_j \in V(T_1)$ such that $c_i\bar{x}_j \in E(T_1)$. So, ϕ is obviously satisfiable by the above true assignment. The proof is complete. ∎

Now we show that for a fixed integer $k \ge 3$, in Problem 2 if we want to decide whether there are k internally disjoint trees connecting S rather than two, the problem is still NP-complete, which can easily be proved by reducing Problem 2 to it.

Lemma 4.2. *For any fixed integer $k \ge 3$, given a graph G and a subset S of $V(G)$, deciding whether there are k internally disjoint trees connecting S, namely deciding whether $\kappa(S) \ge k$, is NP-complete.*

Proof. Clearly, the problem is in NP. We will show that Problem 2 is polynomially reducible to this problem.

Note that k is an fixed integer of at least 3. For any given graph G and a subset S of $V(G)$, we construct a graph $G' = (V', E')$ by adding $k-2$ new vertices to G and joining every new vertex to all vertices in S. We have $|V(G')| = n+k-2$ and $|E(G')| = m+(k-2)|S| \le m+(k-2)n$. Now let S' be a subset of $V(G')$ such that $S' = S$.

If $\kappa_G(S) \ge 2$, it is clear that $\kappa_{G'}(S') \ge k$.

Suppose that there are k internally disjoint trees connecting S' in G', namely $\kappa_{G'}(S') \ge k$. Since there are only $k-2$ new vertices, at least two trees can not contain any new vertex, which means the two trees are actually two internally disjoint trees connecting $S' = S$ in G. The proof is complete. ∎

Combining Lemma 4.1 with Lemma 4.2, we obtain Theorem 1.2, namely, we complete the proof of Theorem 1.2.

Acknowledgement. The authors are grateful to the referees for useful comments and suggestions, which helped to improve the presentation of the paper.

References

[1] J.A. Bondy and U.S.R. Murty, Graph Theory, GTM 244, Springer, 2008.

[2] G. Chartrand, F. Okamoto, P. Zhang, Rainbow trees in graphs and generalized connectivity, Networks 55(4)(2010), 360–367.

[3] S. Li, W. Li, X. Li, The generalized connectivity of complete bipartite graphs, arXiv:1012.5710v1 [math.CO].

[4] S. Li, X. Li, W. Zhou, Sharp bounds for the generalized connectivity $\kappa_3(G)$, Discrete Math. 310(2010), 2147–2163.

[5] F. Okamoto and P. Zhang, The tree connectivity of regular complete bipartite graphs, Journal of Combinatorial Mathematics and Combinatorial Computing 74(2010), 279-293.

[6] N. Robertson, P. Seymour, Graph minors XIII. The disjoint paths problem, J. Combin. Theory Ser.B 63(1995), 65–110.

[7] N.A. Sherwani, Algorithms for VLSI physical design automation, 3rd Edition, Kluwer Acad. Pub., London, 1999.

[8] H. Whitney, Congruent graphs and the connectivity of graphs and the connectivity of graphs, Amer. J. Math. 54(1932), 150–168.