Originally published in *Proc. 11th COCOON*, volume 3595 of *LNCS*, pages 757–766. Springer, 2005.

Bounded Degree Closest *k*-Tree Power is NP-Complete*

Michael Dom, Jiong Guo, and Rolf Niedermeier

Institut für Informatik, Friedrich-Schiller-Universität Jena,
Ernst-Abbe-Platz 2, D-07743 Jena, Fed. Rep. of Germany.
{dom,guo,niedermr}@minet.uni-jena.de

Abstract. An undirected graph $G = (V, E)$ is the k-power of an undirected tree $T = (V, E')$ if $(u, v) \in E$ iff u and v are connected by a path of length at most k in T. The tree T is called the tree root of G. Tree powers can be recognized in polynomial time. The thus naturally arising question is whether a graph G can be modified by adding or deleting a specified number of edges such that G becomes a tree power. This problem becomes NP-complete for $k \geq 2$. Strengthening this result, we answer the main open question of Tsukiji and Chen [COCOON 2004] by showing that the problem remains NP-complete when additionally demanding that the tree roots must have bounded degree.

1 Introduction

Root finding is a natural and well-studied problem in graph algorithmics (see [1, Section 10.6] and [10] for surveys). We call a graph $G' = (V', E')$ a *k-root* of a graph $G = (V, E)$ if $V' = V$ and there is an edge between vertices u and v in G iff there is a path of length at most k between u and v in G'. The other way round, G is the *k-power* of G'. Even determining whether a graph G possesses a 2-root is NP-complete [12].

Kearney and Corneil [8] directed the attention to a special case of the root finding problem by demanding G' to be a tree. Before that, Lin and Skiena [11] have already shown that it can be decided in linear time whether a graph is the 2-power of a tree. Kearney and Corneil generalized this result by showing that the tree root finding problem—called k-Tree Power problem—can be solved in polynomial time for any k. Moreover, they introduced an important generalization of root finding, yielding a natural graph modification problem. The question now is, given a graph G and a nonnegative integer ℓ, can G be modified by adding or deleting at most ℓ edges such that the resulting graph has a k-tree root. Call this problem Closest k-Tree Power. This "error correction scenario" takes into account that a graph might be close to being the k-power of a tree and one tries to find out how close it actually is by considering the number ℓ of edge modifications needed. Kearney and Corneil have shown that the Closest k-Tree Power problem is NP-complete for $k \geq 3$. Moreover, it

* Supported by the Deutsche Forschungsgemeinschaft (DFG), Emmy Noether research group PIAF (fixed-parameter algorithms), NI 369/4.

Originally published in *Proc. 11th COCOON*, volume 3595 of *LNCS*, pages 757–766. Springer, 2005.

is reported that it is also NP-complete in the case $k = 2$ [7]. We strengthen these results to the case that the root trees may only have bounded degree.

Motivated by applications in computational biology, variants of k-Tree Power and Closest k-Tree Power have recently been studied [13,2]. In these problems, only the leaves of the root are in one-to-one correspondence with the given graph vertices, the inner tree nodes are considered as "Steiner nodes" (see [13,2] for details). The corresponding problems Closest k-Leaf Power and Closest k-Phylogenetic Power (where in the latter case all inner nodes of the tree have to have degree at least three) are NP-complete for $k \geq 2$. Intuitively speaking, these problems allow for a higher degree of freedom by freely choosing inner tree nodes and this may explain why, as opposed to tree root finding, polynomial-time solvability of the corresponding recognition problems k-Leaf Power and k-Phylogenetic Power is only known for $k \leq 4$ [13,2]. The cases $k > 4$ are open in both settings. In addition, it has been strongly advocated to study the problems when the maximum node degree of the root tree is bounded from above by a constant [2,3,14]. In particular, Tsukiji and Chen [14] have proven that, for $k \geq 3$, the Closest k-Phylogenetic Power problem (called Closest k-Phylogenetic Root there) remains NP-complete when one demands that the root tree has bounded degree. The case $k = 2$ is open. Moreover, they emphasize that they leave open the "more fundamental" problem to determine the complexity of Closest k-Tree Power [14, page 461] in case of bounded degrees. They conjecture NP-completeness. We settle their open problem by proving this conjecture. More precisely, we show that Closest k-Tree Power is NP-complete for $k \geq 2$ and maximum node degree four in the root tree. We only leave open the case of maximum node degree three.

Let us briefly discuss our result. First, the NP-hardness proof of Kearney and Corneil [8] relies on the NP-completeness of the so-called Fitting Ultrametric Trees problem [9]. To show our result, we had to develop a completely different, more "fine-grained" sort of reduction from the NP-complete Vertex Cover for Graphs with Maximum Degree Three problem (3-Vertex Cover for short) [6]. Second, studying tree powers [8] instead of leaf powers [13] or phylogenetic powers [2], it is impossible to make use of the degree of freedom as provided by inner nodes in the latter two cases. Hence, NP-hardness appears to be harder to show here, somewhat explaining why the problem was left open and considered more fundamental in [14]. Using our new type of construction for the reduction, we could overcome this difficulty, improving Kearney and Corneil's construction [8] which makes use of unbounded degrees.

Due to the lack of space several proofs had to be omitted.

2 Preliminaries

We consider only undirected graphs $G = (V, E)$ with $n := |V|$ and $m := |E|$. Edges are denoted as tuples (u, v). The *degree* of a vertex v is the number of adjacent vertices. For a graph $G = (V, E)$ and $u, v \in V$, let $d_G(u, v)$ denote the length of the shortest path between u and v in G. With $E(G)$, we denote

Originally published in *Proc. 11th COCOON*, volume 3595 of *LNCS*, pages 757–766. Springer, 2005.

the edge set of a graph G. We call a graph $G' = (V', E')$ an *induced subgraph* of $G = (V, E)$ if $V' \subseteq V$ and $E' = \{(u, v) \mid u, v \in V'$ and $(u, v) \in E\}$. For two sets A and B, $A \triangle B$ denotes the *symmetric difference* $(A \setminus B) \cup (B \setminus A)$.

Given an unrooted tree T with node set V, the k*-tree power* of T is a graph, denoted by T^k with $T^k := (V, E)$, where $E := \{(u, v) \mid u, v \in V$ and $d_T(u, v) \leq k\}$. It can be decided in $O(n^3)$ time whether, for specified k, a graph is a k-tree power or not [8]. The more general graph modification problem that asks whether a given graph G is close to any k-tree power T^k then reads as follows.

Closest k-Tree Power (CTPk): Given a graph $G = (V, E)$ and a nonnegative integer ℓ, is there a tree T such that T^k and G differ by at most ℓ edges, that is $|E(T^k) \triangle E(G)| \leq \ell$? CTP$k$ is NP-complete for $k \geq 2$ [8,7]. In this paper we study a special case of CTPk where the degree of every node in T is bounded from above by a fixed constant Δ.

Closest k-Tree Power with Maximum Degree Δ (Δ-CTPk): Given a graph $G = (V, E)$ and a nonnegative integer ℓ, is there a tree T with maximum node degree Δ such that T^k and G differ by at most ℓ edges, that is $|E(T^k) \triangle E(G)| \leq \ell$? Clearly, Δ-CTPk is in NP, because tree powers can be recognized in polynomial time [8]. It remains to show the NP-hardness.

Our reference point for showing NP-completeness of Δ-CTPk is 3-Vertex Cover: Given a graph $G = (V, E)$ with a maximum vertex degree 3 and a nonnegative integer ℓ, is there a set $C \subseteq V$ of at most ℓ vertices such that each edge from E has at least one endpoint in C? 3-Vertex Cover is NP-complete [6]. We show NP-completeness of Δ-CTPk for $k \geq 2$ and $\Delta \geq 4$ by proceeding as follows. First, we study the somewhat simpler case $k \geq 3$. Observe, however, that NP-completeness for some k does not immediately imply NP-completeness for $k + 1$. Second, we strengthen our findings by showing NP-completeness for $k = 2$ where some additional technical expenditure is needed.

To make the presentation clearer, we will speak of vertices when referring to a Vertex Cover input instance in the following sections and we will speak of nodes when referring to a Δ-CTPk instance.

3 Δ-CTPk is NP-complete for $k \geq 3$ and $\Delta \geq 4$

The central point in the NP-completeness proof is to "simulate" the Vertex Cover problem by the graph modification problem CTPk. In the course of this, we will ensure that a Vertex Cover input instance with maximum vertex degree three translates into an instance of CTPk with a desired tree root with maximum node degree four. In what follows, we briefly describe the fundamental ideas behind this reduction of 3-Vertex Cover to 4-CTPk.

Vertex covering means to find a minimum set of vertices that covers all edges. Equivalently, we may consider the following problem. Subdivide each edge of the graph into two edges by inserting a new vertex each time. Then, Vertex Cover can be seen as an edge deletion problem where the task is to break the graph into n connected components such that each connected component contains exactly one original vertex. Moreover, one wants to maximize the number of connected

Originally published in *Proc. 11th COCOON*, volume 3595 of *LNCS*, pages 757–766. Springer, 2005.

components which consist of isolated vertices (or, equivalently, to minimize the number of connected components that contain at least one edge—the corresponding original vertices form the vertex cover). At first sight, this simply sounds as a rather complicated reformulation of VERTEX COVER. The advantage is that this formulation is a step closer to our final goal, a graph modification problem where we modify edges.

So far, observe that in the "new" problem we always have to delete m edges to achieve n connected components as described above. Thus, one difficulty that remains to be solved is to interrelate the number of vertices in a vertex cover and the number of edges modified in 4-CTPk. In addition, we still have to bring into play the tree root problem as such. To this end, we make a construction as follows. Firstly, note that we will connect the n connected components described above by an additional "*backbone structure*" such that we finally can have a connected graph that has a tree root. Secondly, we employ an *edge gadget* which translates the edge deletion scenario of the reformulated VERTEX COVER problem into an edge deletion *and* insertion scenario on the 4-CTPk side. It basically "expresses" that in vertex covering all edges have to be covered. Thirdly, we employ a *vertex gadget* which ensures that we have a one-to-one functional correspondence between the number of vertices of the 3-VERTEX COVER instance and the number of edges to be deleted and inserted in the 4-CTPk instance. It basically expresses that we want to minimize the size of the vertex cover. More specifically, the 3-Vertex Cover problem has a solution of size ℓ iff the constructed 4-CTPk instance has a solution of size $3m+2\ell$. In fact, the first term comes from the edge gadgets, and the second term comes from the vertex gadgets. We illustrate our reduction by focussing on Δ-CTP3.

Construction of the reduction. We now describe the details of the construction for Δ-CTP3 for $\Delta \geq 4$. Given an instance $G = (V, E)$ of 3-VERTEX COVER with $V := \{v_1, \dots, v_n\}$, we construct the graph $G_{\text{CTP}} = (V_{\text{CTP}}, E_{\text{CTP}})$ as follows.

For every vertex $v_i \in V$ there is a *vertex gadget* in G_{CTP} that contains a *vertex node* x_0^i, a *connection stub* consisting of eight nodes $x_1^i, \dots, x_8^i$, an *edge stub* for every neighbor v_j of v_i—each edge stub consisting of two nodes $y_1^{i,j}$ and $y_2^{i,j}$—, and edges as shown in Figure 1.

To build the mentioned backbone structure, we add $n-1$ *connection nodes* z^i with $1 \leq i < n$ to G_{CTP}, and for all $1 \leq i < n$ we insert edges between the vertex gadgets of v_i and v_{i+1} and between the gadgets and z^i as shown in Figure 2.

For each edge $(v_i, v_j) \in E$, we add to G_{CTP} an *edge node* $e^{i,j}$ and insert edges between $e^{i,j}$ and the nodes x_0^i, $y_2^{i,j}$, x_0^j, $y_2^{j,i}$ from the vertex gadgets of v_i and v_j. See Figure 3 for an illustration. We call $e^{i,j}$ together with the four edges incident to it the *edge gadget* for $(v_i, v_j) \in E$.

Clearly, G_{CTP} is not a 3-tree power if G contains any edges, because in a 3-tree power T^k that contains at least four nodes every node u has at least three pairwise connected neighbors. To see this, consider the 3-tree root T of T^k: If there is a node v with distance 3 from u in T, then the two vertices between u and v form a clique in T^k together with v and u. Similarly, one can also find

Originally published in *Proc. 11th COCOON*, volume 3595 of *LNCS*, pages 757–766. Springer, 2005.

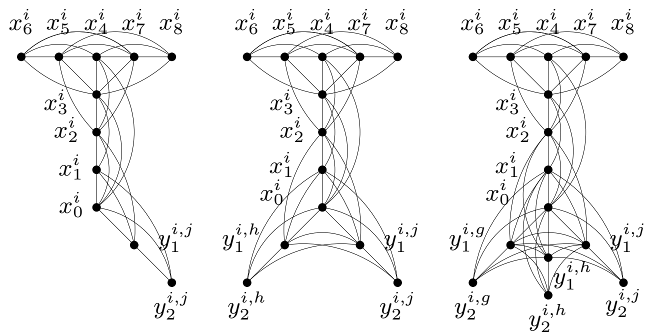

Fig. 1. The vertex gadget of a vertex $v_i \in V$. If v_i has only one neighbor v_j in G, the gadget of v_i has only one edge stub as shown on the left side. The illustrations in the middle resp. on the right side show the gadget of v_i in the case that v_i has two neighbors v_h, v_j resp. three neighbors v_g, v_h, v_j.

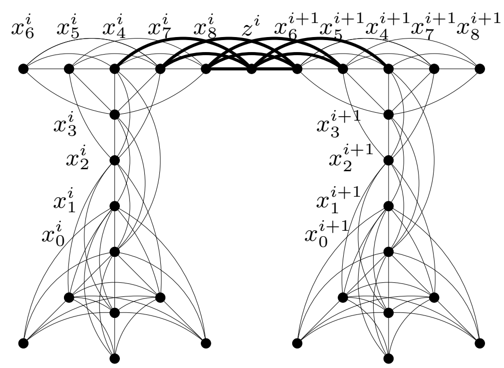

Fig. 2. The vertex gadgets of $v_i, v_{i+1} \in V$, and the connection node z^i. The edges inserted between the two gadgets and between the gadgets and z^i are drawn with bold lines.

three pairwise connected neighbors of u in T^k if all nodes in T are at distance at most 2 from u. However, every edge node $e^{i,j}$ in G_{CTP} has four neighbors with only two edges between them. Figure 4 gives an example for the reduction.

Correctness of the reduction.

Proposition 1. *Let $G = (V, E)$ be an instance of* 3-VERTEX COVER *and let G_{CTP} be the instance of* Δ-CTP3 *constructed from G as described above.*

If $C \subseteq V$ is a vertex cover for G, then G_{CTP} has a solution of size at most $3 \cdot m + 2 \cdot |C|$.

Proof. We prove the proposition by giving a solution of the postulated size for G_{CTP}. Let $C \subseteq V$ be a vertex cover for G, that is, every edge of E has at least one endpoint in C. Then we modify G_{CTP} as follows:

Originally published in *Proc. 11th COCOON*, volume 3595 of *LNCS*, pages 757–766. Springer, 2005.

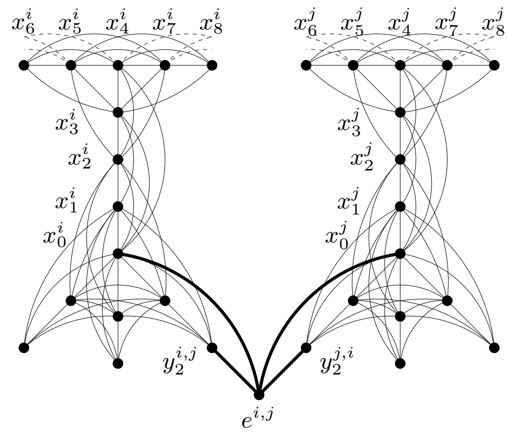

Fig. 3. The edge node $e^{i,j}$ for $(v_i, v_j) \in E$ and the vertex gadgets of v_i and v_j. The edges of the edge gadget are drawn with bold lines.

For every vertex $v_i \in C$ delete the edge (x_0^i, x_4^i) and insert the edge (x_1^i, x_4^i). For every edge $(v_i, v_j) \in E$, at least one of v_i and v_j, say v_i, is in C. Then insert the edge $(e^{i,j}, y_1^{i,j})$ and delete the edges $(e^{i,j}, x_0^j)$ and $(e^{i,j}, y_2^{j,i})$.

This solution has size $m \cdot 3 + |C| \cdot 2$, since we modify two edges in G_{CTP} for every vertex in C and three edges for every edge of E.

The resulting graph has a 3-tree root T with maximum vertex degree 4: Every edge node is connected with exactly one vertex gadget which is modified such that it has a 3-tree root as shown in Figure 4 for the gadget of v_2 (with x_1^2 lying between x_0^2 and x_2^2 in the tree root). If a vertex gadget is disconnected from all edge nodes, it has a 3-tree root like the gadgets of v_1 and v_3 in Figure 4 (with x_0^1 lying between x_1^1 and x_2^1). $\square$

In order to show the reverse direction, we need the following lemma. We omit the lengthy proof.

Lemma 1. *Given a graph $G_{\text{CTP}} = (V_{\text{CTP}}, E_{\text{CTP}})$ constructed as described above, there is an optimal solution E_{opt} for Δ-CTP3 on G_{CTP} that leads to a graph $G_{\text{opt}} = (V_{\text{CTP}}, E_{\text{CTP}} \triangle E_{\text{opt}})$ with the following two properties:*
Edge node property. *Each edge node $e^{i,j}$ which is added into G_{CTP} for edge $(v_i, v_j) \in E$ has only three neighbors in G_{opt}, and these are either $x_0^i, y_1^{i,j}, y_2^{i,j}$ or $x_0^j, y_1^{j,i}, y_2^{j,i}$.*
Vertex node property. *For each vertex node x_0^i with $1 \leq i \leq n$, if there is an edge node $e^{i,j}$ adjacent to x_0^i in G_{opt}, then G_{opt} contains edge (x_1^i, x_4^i) but not edge (x_0^i, x_4^i); otherwise, G_{opt} contains (x_0^i, x_4^i) but not (x_1^i, x_4^i).*

Proposition 2. *Let $G = (V, E)$ be an instance of* 3-Vertex Cover *and let $G_{\text{CTP}} = (V_{\text{CTP}}, E_{\text{CTP}})$ be the instance of Δ-CTP3 constructed from G. If E_{sol} is a solution for G_{CTP}, then G has a vertex cover of size at most $(|E_{\text{sol}}| - 3 \cdot m)/2$.*

Originally published in *Proc. 11th COCOON*, volume 3595 of *LNCS*, pages 757–766. Springer, 2005.

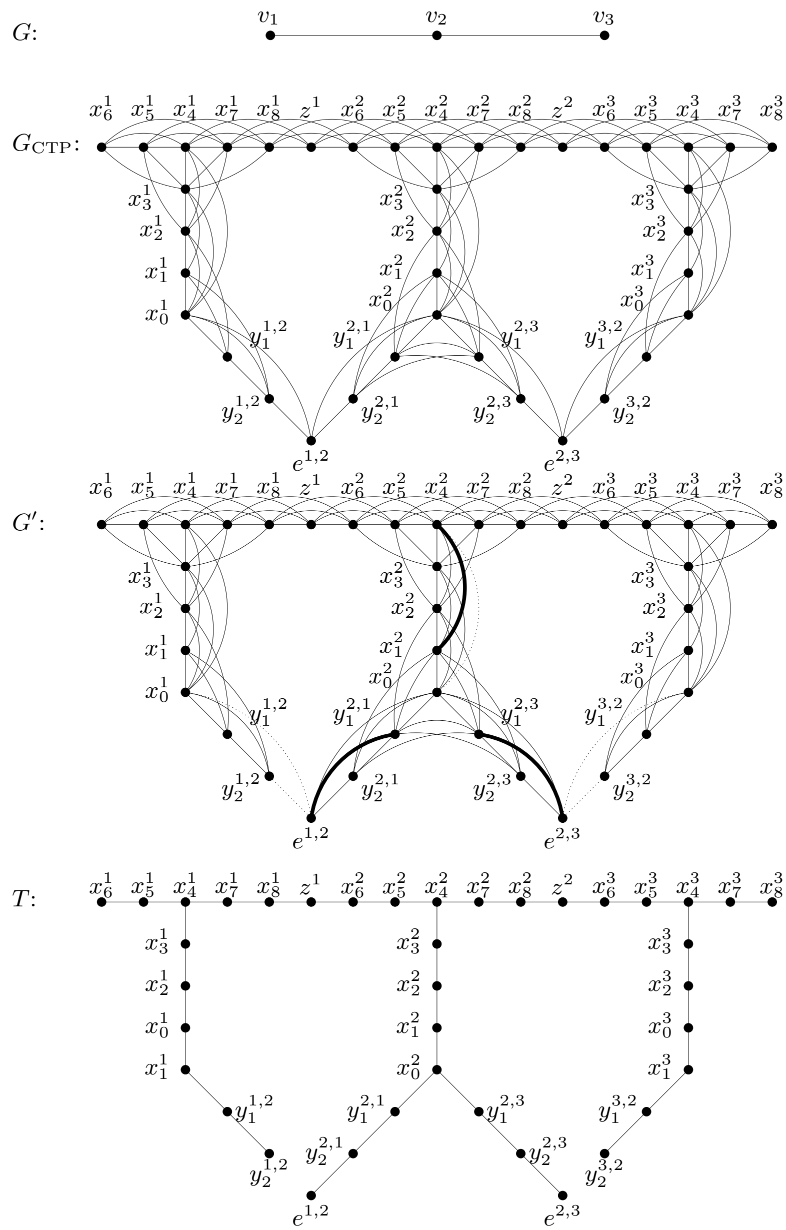

Fig. 4. An example reduction. Graph G is the 3-Vertex Cover instance, graph G_{CTP} is the graph constructed from G. There are three vertex gadgets with vertex nodes x_0^1, x_0^2, x_0^3 for the three vertices in G and two edge gadgets with edge nodes $e^{1,2}, e^{2,3}$ for the two edges of G. Graph G' results from G_{CTP} by deleting five edges and inserting three edges (inserted edges are drawn with bold lines, deleted edges with dotted lines). Graph G' is a 3-tree power with T as its 3-tree root. Note that we need three edge modifications for each edge of G and two edge modifications for the vertex gadget of v_2, which forms a vertex cover for G.

Originally published in *Proc. 11th COCOON*, volume 3595 of *LNCS*, pages 757–766. Springer, 2005.

Proof. Let $E_{\text{sol}} \subseteq V_{\text{CTP}} \times V_{\text{CTP}}$ be an optimal solution for G_{CTP} as described in Lemma 1. In the resulting graph, every edge node $e^{i,j}$ is connected either to $x_0^i, y_1^{i,j}, y_2^{i,j}$ or to $x_0^j, y_1^{j,i}, y_2^{j,i}$ (edge node property of Lemma 1). Hence, for every edge node $e^{i,j}$ there are three edge modifications in E_{sol}, either the deletion of $(e^{i,j}, x_0^i)$, $(e^{i,j}, y_2^{i,j})$ and the insertion of $(e^{i,j}, y_1^{j,i})$, or the deletion of $(e^{i,j}, x_0^j)$, $(e^{i,j}, y_2^{j,i})$ and the insertion of $(e^{i,j}, y_1^{i,j})$.

As every edge node $e^{i,j}$ is adjacent to exactly one of the vertex nodes x_0^i and x_0^j in the resulting graph,

$$C := \{v_i \subseteq V \mid x_0^i \text{ is adjacent to at least one edge node}\}$$

is clearly a vertex cover for G. The vertex node property of Lemma 1 implies that for every vertex node x_0^i that is adjacent to at least one edge node, the solution E_{sol} contains two edge modifications, namely the deletion of (x_0^i, x_4^i) and the insertion of (x_1^i, x_4^i). Hence, there can be at most $(|E_{\text{sol}}| - 3 \cdot m)/2$ such vertex nodes, and the size of C, which consists of the corresponding vertices in V, is bounded from above by $(|E_{\text{sol}}| - 3 \cdot m)/2$. □

Theorem 1. *Δ-CTPk is* NP*-complete for $k = 3$ and $\Delta \geq 4$.*

To generalize to Δ-CTPk for $k > 3$, we use a straightforward extension of the construction used for the case $k = 3$. The gadget for a vertex $v_i \in V$ then consists of a vertex node x_0^i, $3k - 1$ nodes $x_1^i, \ldots, x_{3k-1}^i$ (the connection stub), and $k - 1$ nodes $y_1^{i,j}, \ldots, y_{k-1}^{i,j}$ for each neighbor v_j of v_i (the nodes of the edge stubs). For each edge $(v_i, v_j) \in E$ there is an edge node $e^{i,j}$ with edges to $x_0^i, y_{k-1}^{i,j}, x_0^j$ and $y_{k-1}^{j,i}$. All ideas and proofs used for $k = 3$ also hold for $k > 3$, which leads to the following theorem:

Theorem 2. *Δ-CTPk is* NP*-complete for $k > 3$ and $\Delta \geq 4$.*

4 Δ-CTP2 is NP-complete for $\Delta \geq 4$

In this section we show the NP-completeness of Δ-CTP2 for $\Delta \geq 4$. The reduction is also from 3-Vertex Cover. Compared to the reduction in Section 3, the only difference lies in the edge gadget. In the construction in Section 3, a decisive point was that in the edge gadget with edge node $e^{i,j}$ any optimal solution *needs* to disconnect exactly one of the two vertex gadgets corresponding to vertices v_i and v_j from $e^{i,j}$. More precisely, the vertex gadget for the covering vertex (or the vertex gadget for exactly one arbitrary covering vertex if there are two) stayed connected with $e^{i,j}$ whereas the vertex gadget for the other vertex became disconnected. In case of CTP2, however, with this construction it is no longer obvious that an optimal solution needs to disconnect exactly one of the two vertex gadgets. That is why we introduce a somewhat more complicated edge gadget, where we basically replace the *one* edge node $e^{i,j}$ by a clique of five nodes.

To present the refined construction and demonstrate its correctness, we employ forbidden subgraphs as shown in Figure 5. No 2-tree power has any of these as vertex-induced subgraph. A proof of the following lemma can be found in [5].

Originally published in *Proc. 11th COCOON*, volume 3595 of *LNCS*, pages 757–766. Springer, 2005.

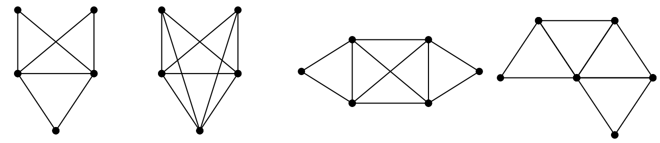

Fig. 5. Four forbidden induced subgraphs for 2-tree powers.

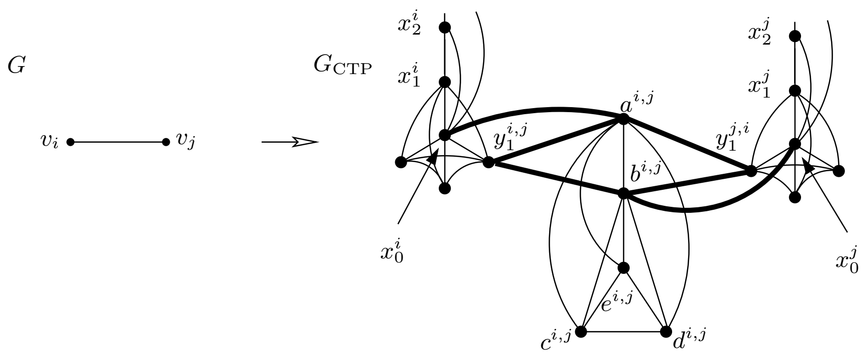

Fig. 6. The edge gadget used in the reduction from 3-Vertex Cover to Δ-CTP2 is a 5-nodes clique consisting of nodes $a^{i,j}, b^{i,j}, c^{i,j}, d^{i,j}, e^{i,j}$. The edges inserted between the edge gadget and the two vertex gadgets are drawn as bold lines. Nodes $y_1^{i,j}, y_1^{j,i}$ and $a^{i,j}, b^{i,j}$ together with each of $c^{i,j}, d^{i,j}, e^{i,j}$ form a forbidden induced subgraph G_1 as shown in Figure 5.

Lemma 2. *If a graph G has a 2-tree root, then G does not contain the subgraphs shown in Figure 5 as induced subgraphs.*

We use G_1 in Figure 5 to construct the edge gadget. The other three forbidden induced subgraphs are not directly used in the reduction but will be used in the proof of Lemma 3.

Since subgraph G_1 in Figure 5 is a forbidden induced subgraph for 2-tree powers, we need at least one edge modification to edit G_1 into a graph having a 2-tree root. Based on this observation, the edge gadget for $(v_i, v_j) \in E$ consists of five nodes which form a clique. Moreover, edges are inserted to connect two nodes of this edge gadget to the vertex gadgets of v_i and v_j to form induced subgraphs G_1, see Figure 6 for an illustration. Thus, if the 3-Vertex Cover instance G contains edges, then G_{CTP} is not a 2-tree power.

With exception of the edge gadget, the rest of the reduction is the same as the one in Section 3. The proof of the following lemma is very similar to the proofs of Propositions 1 and 2.

Lemma 3. *Given a* 3-Vertex Cover *instance $G = (V, E)$. Let G_{CTP} denote the graph constructed as described above. There is a vertex cover of ℓ vertices iff G_{CTP} can be transformed into a 2-tree power by $3 \cdot m + 2 \cdot \ell$ edge modifications.*

Originally published in *Proc. 11th COCOON*, volume 3595 of *LNCS*, pages 757–766. Springer, 2005.

Theorem 3. Δ-CTP2 *is NP-complete for* $\Delta \geq 4$.

5 Conclusion

Showing NP-completeness of Closest k-Tree Power for $k \geq 2$ and maximum vertex degree four, we basically settled the open question of Tsukiji and Chen [14] and strengthened results of Kearney and Corneil [8]. Only the case with maximum vertex degree three is left open. We conjecture that by a further refinement of our type of reduction NP-completeness can be also shown here. Moreover, it would be interesting to study the complexities of the graph modification problems when one only allows either adding or deleting edges. Finally, investigating the polynomial-time approximability or fixed-parameter tractability of the proven NP-complete problems is a task for future research. Fixed-parameter tractability for closely related leaf root problems is shown in [4,5].

References

1. A. Brandstädt, V. B. Le, and J. P. Spinrad. *Graph Classes: a Survey.* SIAM Monographs on Discrete Mathematics and Applications, 1999.
2. Z.-Z. Chen, T. Jiang, and G. Lin. Computing phylogenetic roots with bounded degrees and errors. *SIAM Journal on Computing*, 32(4):864–879, 2003.
3. Z.-Z. Chen and T. Tsukiji. Computing bounded-degree phylogenetic roots of disconnected graphs. In *Proc. 30th WG*, volume 3353 of *LNCS*, pages 308–319. Springer, 2004.
4. M. Dom, J. Guo, F. Hüffner, and R. Niedermeier. Error compensation in leaf root problems. In *Proc. 15th ISAAC*, volume 3341 of *LNCS*, pages 389–401. Springer, 2004. Long version to appear in *Algorithmica*.
5. M. Dom, J. Guo, F. Hüffner, and R. Niedermeier. Extending the tractability border for closest leaf powers. In *Proc. 31st WG*, LNCS. Springer, 2005. To appear.
6. M. R. Garey, D. S. Johnson, and L. J. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1(3):237–267, 1976.
7. T. Jiang, G. Lin, and J. Xu. On the closest tree kth root problem. Manuscript, Department of Computer Science, University of Waterloo, 2000.
8. P. E. Kearney and D. G. Corneil. Tree powers. *Journal of Algorithms*, 29(1):111–131, 1998.
9. M. Křivánek and J. Morávek. NP-hard problems in hierarchical-tree clustering. *Acta Informatica*, 23(3):311–323, 1986.
10. L. C. Lau and D. G. Corneil. Recognizing powers of proper interval, split, and chordal graphs. *SIAM Journal on Discrete Mathmatics*, 18(1):83–102, 2004.
11. Y. L. Lin and S. S. Skiena. Algorithms for square roots of graphs. *SIAM Journal on Discrete Mathematics*, 8(1):99–118, 1995.
12. R. Motwani and M. Sudan. Computing roots of graphs is hard. *Discrete Applied Mathematics*, 54(1):81–88, 1994.
13. N. Nishimura, P. Ragde, and D. M. Thilikos. On graph powers for leaf-labeled trees. *Journal of Algorithms*, 42(1):69–108, 2002.
14. T. Tsukiji and Z.-Z. Chen. Computing phylogenetic roots with bounded degrees and errors is hard. In *Proc. 10th COCOON*, volume 3106 of *LNCS*, pages 450–461. Springer, 2004.