数理解析研究所講究録
第 1599 巻 2008 年 119-126

完全 k 分木の path distance width について
(On the path distance width of the complete k-ary trees)

群馬大学大学院工学研究科 受川 和幸 (Kazuyuki Ukegawa), 青木 一正 (Kazumasa Aoki),
小澤 恭平 (Kyohei Kozawa), 大舘 陽太 (Yota Otachi), 山崎 浩一 (Koichi Yamazaki)
Department of Computer Science, Gunma University

1 Introduction

Given a connected graph $G = (V, E)$ and a subset of vertices $X_1 \subseteq V$, it is easy to construct the decomposition $D = (X_1, \ldots, X_t)$ such that X_i is the set of vertices of distance $i-1$ from X_1 for each $1 \leq i \leq t$, where t is the largest integer satisfying $X_t \neq \emptyset$. We call X_i's by *levels* and denote the number of levels in D (i.e. t) by $|D|$. We call the decomposition by the *distance structure with initial set* X_1 *of* G, and denote it by $D(X_1, G)$ or simply $D(X_1)$ or more simply D if it is clear from the context. The *width of* D, denoted by $\mathrm{pdw}_D(G)$, is defined as $\max_{0 \leq i \leq t} |X_i|$. The *path distance width of* G, denoted by $\mathrm{pdw}(G)$, is defined as $\min_{X_1 \subseteq V} \mathrm{pdw}_{D(X_1)}(G)$. It is known that even for the trees T the optimization problem computing $\mathrm{pdw}(T)$ is NP-hard and does not admit a PTAS [8, 9]. It is not known whether or not the problem is fixed parameter tractable, i.e., whether there exists an algorithm that solve the decision problem, $\mathrm{pdw}(G) \leq k$ or not, with running time $O(f(k)n^c)$, where c is a constant and independent of k, and f is any function.

The problem computing path distance width is related to the problem computing *bandwidth*. It is easy to see that for any graph G the bandwidth of G is at most $2\mathrm{pdw}(G)-1$. But the difference between them can be arbitrarily large [9]. From the point of view of graph algorithm design the problem computing bandwidth is an attractive subject of research because the problem is known to be a computationally hard problem: the problem is NP-complete even for the trees, has no constant factor approximation, and fixed parameter intractable [7, 1]. It would be important to understand better what makes problems hard. We suspect that the problem computing path distance width is also a hard problem.

For complete k-ary trees, several graph parameters have been studied, such as bandwidth [6], antibandwidth [2], edge-bandwidth [3], harmonious chromatic number [4], vertex boundary-width [5].

Smithline showed that the *density lower bound* determines the bandwidth of the complete k-ary trees [6]. Smithline showed that the *density lower bound* $\mathrm{bw}(G) \geq \left\lceil \frac{|V(G)|-1}{\mathrm{diam}(G)} \right\rceil$ determines the bandwidth of the complete k-ary trees [6], where $\mathrm{diam}(G)$ is the diameter of a connected graph G. The density lower bound is based on the pigeon hole principle. The path distance width also has a lower bound based on the pigeon hole principle, that is, $\mathrm{pdw}(G) \geq \left\lceil \frac{|V(G)|}{\mathrm{diam}(G)+1} \right\rceil$. We refer to the lower bound for $\mathrm{pdw}(G)$ also as density lower bound. It would also be an interesting question whether the density lower bound determines the path distance width of the complete k-ary trees or not. The problem has been left as an open problem [8].

In this paper we consider the path distance width of complete k-ary trees. We first show that path distance width of complete k-ary trees does not coincide with the density lower bound, more precisely, we give a better lower bound for complete k-ary trees. Then we show an upper bound of complete k-ary trees for which the ratio between the upper bound and density lower bound is independent of the depth d.

2 Notations

The *depth of a (sub)tree* is the number of edges in a longest path from the root to a leaf. Let $T_{k,d}$ denote the complete k-ary tree of depth d, T be a subtree of $T_{k,d}$, and D be a distance structure of $T_{k,d}$. T is called *complete* if T is a complete k-ary tree and the leaves of T are also leaves in $T_{k,d}$. A complete subtree T is called an *unfolded subtree in D* if all leaves of T are in the same level ℓ of D for some $1 \le \ell \le |D|$ and the level of the root of T is $\ell -$ (the depth of T). An unfolded subtree T in D is called *maximal* if there is no unfolded subtree in D that properly contains T. We denote the width of level ℓ by $D(\ell) = |X_\ell|$ and the level of a vertex v by $\mathrm{level}_D(v) = \ell$ if $v \in X_\ell$.

A *sibling* of a vertex $v \in V(T_{k,d})$ is a vertex that has the same parent as v. A *sibling subtree* T' of a complete subtree T of $T_{k,d}$ is a complete subtree such that the root of T' is a sibling of the root of T.

Let v be a vertex of $T_{k,d}$. The *height* of v, denoted by height(v), is the distance from v to the *nearest* leaf. Note that not the *farthest* leaf. For example the height of a leaf is zero and the height of the root of $T_{k,d}$ is d.

For convenience we denote the following three functions:

- $f(k) = 2(k-1)/k$,
- $\mu(k,d) = \lfloor \log_k (f(k)(d - \lfloor \log_k (f(k)d) \rfloor)) \rfloor$,
- $g(k,m) = (k^m - 1)/f(k) + m + 1$.

3 Lower bound

In this section we give a lower bound for complete k-ary trees. It is easy to derive the following density lower bound:

$$\mathrm{pdw}(T_{k,d}) \ge \left\lceil \frac{|V(T_{k,d})|}{\mathrm{diam}(T_{k,d}) + 1} \right\rceil = \left\lceil \frac{k^{d+1} - 1}{(k-1)(2d+1)} \right\rceil.$$

In this paper, we show a better lower bound,

$$\mathrm{pdw}(T_{k,d}) \ge \left\lceil \frac{|V(T_{k,d})|}{\mathrm{diam}(T_{k,d}) + 1 - 2\mu(k,d)} \right\rceil = \left\lceil \frac{k^{d+1} - 1}{(k-1)(2d+1-2\mu(k,d))} \right\rceil.$$

The following two lemmas are the main tools for deriving our lower bound.

Lemma 3.1. *Let $D = (X_1, \ldots, X_{|D|})$ be a distance structure of $T_{k,d}$ and X_ℓ be a level containing a leaf v of $T_{k,d}$. Then there is a maximal unfolded subtree of depth at least $\lceil (\ell - 3)/2 \rceil$ in D.*

Proof. It is easy to see that there is a unique maximal unfolded subtree T (in D) containing v. We show that T is a desired maximal unfolded subtree. Suppose that $T \ne T_{k,d}$ (Otherwise it is trivial). Because T is maximal, X_1 has a vertex u in a sibling subtree of T. Let d_T denote the depth of T. The distance between v and u is at most $2d_T + 2$ and at least $\ell - 1$, so the lemma holds. □

Lemma 3.2. *Let $D = (X_1, \ldots, X_{|D|})$ be a distance structure of $T_{k,d}$. If $X_{|D|}$ has an internal vertex u of $T_{k,d}$, then $|D| \le d + 1$.*

Proof. Because $u \in X_{|D|}$ is an internal vertex, there are two vertex disjoint paths (except u) from u to X_1. Since two paths have the length at least $|D| - 1$, $T_{k,d}$ has a path of length $2(|D| - 1)$. As $T_{k,d}$ has diameter $2d$, we have that $|D| \le d + 1$. □

From the above lemmas, we can obtain the new lower bound for a distance structure that has a large enough number of levels.

Corollary 3.3. *Let $D = (X_1, \ldots, X_{|D|})$ be a distance structure of $T_{k,d}$. If $|D| \ge d + 2$, then $\mathrm{pdw}_D(T_{k,d}) \ge k^{\lceil (|D|-3)/2 \rceil}$.*

Proof. From Lemma 3.2, $X_{|D|}$ has no internal vertex. So, there is a maximal unfolded subtree T of depth at least $\lceil (|D| - 3)/2 \rceil$ from Lemma 3.1. The lemma follows from the number of T's leaves. □

By combining Corollary 3.3 and the density lower bound formula, we have the following lower bound for a complete k-ary tree $T_{k,d}$, not for a distance structure D.

Corollary 3.4. $pdw(T_{k,d}) \geq \min\left\{\left\lceil \frac{|V(T_{k,d})|}{d+1} \right\rceil, \min_{\ell=d+2}^{2d+1} \max\left\{k^{\lceil(\ell-3)/2\rceil}, \left\lceil \frac{|V(T_{k,d})|}{\ell} \right\rceil\right\}\right\}$.

We will state the above corollary in the closed form. From a basic calculation we have the following lemma.

Lemma 3.5. $\min_{\ell=d+2}^{2d+1} \max\left\{k^{\lceil(\ell-3)/2\rceil}, \left\lceil \frac{|V(T_{k,d})|}{\ell} \right\rceil\right\} \geq \left\lceil \frac{|V(T_{k,d})|}{2d+1-2\mu(k,d)} \right\rceil$.

For $d \geq 1$, we have the following lemma.

Lemma 3.6. $\left\lceil \frac{|V(T_{k,d})|}{d+1} \right\rceil \geq \left\lceil \frac{|V(T_{k,d})|}{2d+1-2\mu(k,d)} \right\rceil$.

Now, we are ready to state the lower bound in closed form.

Theorem 3.7. $pdw(T_{k,d}) \geq \left\lceil \frac{|V(T_{k,d})|}{2d+1-2\mu(k,d)} \right\rceil$.

Proof. From Corollary 3.4, Lemma 3.5, and 3.6. □

4 Upper bound

We have a naive upper bound $\mathrm{pdw}(T_{k,d}) \leq k^d/2$ (a half of the number of leaves). But this upper bound is so far from our lower bound $\approx k^d/(2(d-\log_k d))$. In fact, the ratio $\frac{k^d/2}{k^d/(2(d-\log_k d))}$ depends on the depth d. So it would be nice to have an upper bound for which the ratio is independent of the depth d. In this section, for even numbers $k \geq 4$, we will show a better upper bound which ensure that the ratio is independent of the depth d. For odd numbers k, a similar upper bound can be derived by performing a similar calculation as in the even case. However, in the detailed calculation, the odd case is much more troublesome than the even case. The reason why the even case can be handed easily is that $g(k,m)$ takes a nonnegative integer for any nonnegative integer m if k is an even natural number. So in what follows, we will consider a k-ary tree $T_{k,d}$ for an even number $k \geq 4$. For convenience let m be the number such that $g(k,m) \leq d = g(k,m) + a < g(k,m+1)$. Then $\mu(k,d) = m$ from Lemma Appendix A.1.

To show our upper bound, we give a transformation F from (k,d) to $X^* \subseteq V(T_{k,d})$ such that

1. $\mathrm{pdw}_{D(X^*)}(T_{k,d})$ is close to the lower bound for $\mathrm{pdw}(T_{k,d})$, and
2. $\mathrm{pdw}_{D(X^*)}(T_{k,d})$ can be estimated accurately.

We also show that the ratio $\frac{\mathrm{pdw}_{D(X^*)}(T_{k,d})}{\text{lower bound in Theorem 3.7}}$ is at most k^2+1 (independent of the depth). Our $F(k,d)$ consists of vertices of height zero or one. This makes estimation of $\mathrm{pdw}_{D(F(k,d))}(T_{k,d})$ feasible. To describe F, we need a concept of "*release*": We first consider the set X^0 of all vertices of height zero and one both as an initial set; Then we *release* (remove) vertices from X^0 step by step; Then finally we have the desired set X^*. The procedure of release i.e. F is described in Figure 2. We will call F by Improvement.

Now we make a detail explanation of the transformation Improvement. Let D^0 the distance structure $D(X^0)$ .It is easy to see that

$$D^0(\ell) = \begin{cases} k^d + k^{d-1}, & (\ell = 1) \\ k^{d-\ell}, & (2 \leq \ell \leq d) \\ 0. & (\text{otherwise}) \end{cases}$$

As explained before, to obtain the initial set X^*, we release some vertices in X^0 step by step. Let $X^* \subseteq X \subseteq X^0$ be an initial set that appears in the process of the step by step releasing. A complete subtree T of $T_{k,d}$ is *unreleased at* X if $\{v \mid v \in V(T), \mathrm{height}(v) = 0 \text{ or } 1\} \subseteq X$. Let β be a BFS ordering of $V(T_{k,d})$ started from the root of $T_{k,d}$. For two unreleased complete subtrees T_1 and T_2 at X (not necessarily be of the same size), T_1 *is to the left of* T_2 if $\min\{\beta(v) \mid v \in V(T_1)\} < \min\{\beta(v) \mid v \in V(T_2)\}$. An unreleased complete subtree T is the *leftmost of depth* j *at* X if T is of depth j, T is unreleased at X, and there is no unreleased complete subtree T' of depth j such that T' is to the left of T.

The transformation IMPROVEMENT uses a procedure $\text{release}_i(j)$. The procedure $\text{release}_i(j)$ removes some vertices from X in the following way: (1) Let T_j be the leftmost unreleased complete subtree of depth j, (2) select an arbitrary vertex $v \in V(T_j)$ with $\text{height}(v) = i$, (3) release vertices $X \cap V(T) \setminus \{v\}$ from X. In our transformation $i \in \{0, 1\}$. The results of transformations $\text{release}_0(3)$ and $\text{release}_1(3)$ are depicted in Fig. 1.

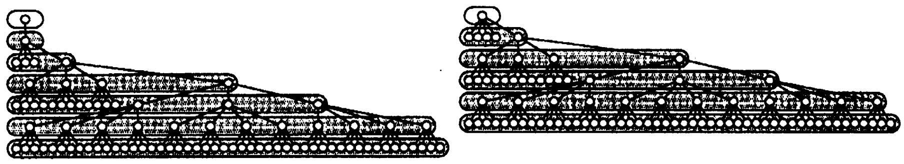

Fig. 1 Examples of the transformations ($k = 4$).

Now we are ready to state the whole transformation IMPROVEMENT. This transformation starts from the initial structure D^0, and output the resultant structure $D^* = D(X^*)$. We apply the long sequence of the transformations release_0 and release_1 alternatively. See Fig. 2 for the complete definition of IMPROVEMENT.

```
Transformation IMPROVEMENT(k, d)
    Build the initial structure D^0;
    Fix a BFS ordering β started from the root;
    release_0(d − μ(k, d) + 1);
    repeat k − 1 times
      release_1(d − μ(k, d)); release_0(d − μ(k, d));
    end
    for j = d − μ(k, d) − 1 to 3k/2 + a + 1 do
      repeat (k − 1)k^(d−μ(k,d)−j−1) times
        release_1(j); release_0(j);
      end
    end
    Output the resultant structure D*;
```

Fig. 2 The transformation IMPROVEMENT.

We should show that IMPROVEMENT is applicable, that is, there exists an unreleased complete subtree of depth j whenever $\text{release}_i(j)$ ($i \in \{0, 1\}$) is called in IMPROVEMENT. First we estimate the number of complete subtrees that are used in IMPROVEMENT.

Proposition 4.1. *Let $q_i^k(j, d)$ be the number of times $\text{release}_i(j)$ is called in* IMPROVEMENT(k, d), *and J be the set $\{j \mid 3k/2 + a + 1 \le j \le d - \mu(k, d) - 1\}$. Then,*

$$q_i^k(j, d) = \begin{cases} 1, & (i = 0,\ j = d - \mu(k, d) + 1) \\ k - 1, & (i = 0, 1,\ j = d - \mu(k, d)). \\ (k - 1)k^{d-\mu(k,d)-j-1}, & (i = 0, 1,\ j \in J). \\ 0. & (otherwise) \end{cases}$$

Lemma 4.2. *There exists an unreleased complete subtree of depth j whenever $\text{release}_i(j)$ ($i \in \{0, 1\}$) is called in* IMPROVEMENT.

Proof. In this proof we denote a complete subtree of depth $d - \mu(k, d) - 1$ by ***unit***. We count how many unit IMPROVEMENT consumes. Note that $T_{k,d}$ contains $k^{\mu(k,d)+1}$ disjoint units and that for each iteration of **for** loop, IMPROVEMENT consumes $2(k - 1)k^{d-\mu(k,d)-j-1}$ released complete subtrees of depth j. We will refer to the set of released complete subtrees as *block*. From Proposition 4.1, the number of used complete subtrees of depth j is,

$$\begin{cases} 1, & (j = d - \mu(k,d) + 1) \\ 2(k-1), & (j = d - \mu(k,d)) \\ 2(k-1)k^{d-\mu(k,d)-j-1}, & (3k/2 + a + 1 \le j \le d - \mu(k,d) - 1) \\ 0. & (\text{otherwise}) \end{cases}$$

For the deepest complete subtree, k^2 units are used. For the complete subtrees of depth $d - \mu(k,d)$, $2k(k-1)$ units are used. For the complete subtrees of depth j ($3k/2 + a + 1 \le j \le d - \mu(k,d) - 1$), $2(k-1)$ units are used in each block. So the number of units that are used in IMPROVEMENT is:

$$\begin{aligned} &k^2 + 2k(k-1) + 2(k-1)\left((d - \mu(k,d) - 1) - (3k/2 + a + 1) + 1\right) \\ &= k^2 + 2k(k-1) + 2(k-1)\left(g(k,m) - m - 1 - 3k/2\right) \\ &= k^2 + 2k(k-1) + 2(k-1)\left(\left(\frac{k(k^m-1)}{2(k-1)} + m + 1\right) - m - 1 - 3k/2\right) \\ &= k^{m+1} = k^{\mu(k,d)+1}. \end{aligned}$$

This number and the number of units contained in $T_{k,d}$ are the same. □

We consider the difference between D^0 and D^*. Recall that vertices in an initial set are in level 1, not 0.

Proposition 4.3. *For any vertex $v \notin X^0$, $level_{D^0}(v) = height(v)$.*

Proposition 4.4. *For any vertex v, $level_{D^0}(v) \le level_{D^*}(v)$.*

Lemma 4.5. *If a vertex $v \notin X^0$ has a descendant u such that $u \in X^*$ and $height(u) = 1$, then $level_{D^0}(v) = level_{D^*}(v)$.*

Proof. From Proposition 4.3 and Proposition 4.4, $\text{height}(v) \le \text{level}_{D^*}(v)$. It is easy to see that $\text{level}_{D^*}(v) \le \text{dist}(u,v) + 1 = \text{height}(v)$. Thus $\text{level}_{D^*}(v) = \text{height}(v) = \text{level}_{D^0}(v)$. □

Corollary 4.6. *For a vertex v, if $level_{D^0}(v) \ne level_{D^*}(v)$ then v is in a complete subtree that was released by the algorithm* IMPROVEMENT.

Proof. Let T_v be a complete subtree with root v, and $h1_v$ be the set $\{u \mid u \text{ is descendant } u \text{ of } v, \text{height}(u) = 1\}$. From Lemma 4.5, $X^* \cap h1_v = \emptyset$. This means that for each $u \in h1_v$ there is a unique released complete subtree $T(u)$ such that $T(u)$ contains u and $T(u)$ was released by IMPROVEMENT. $T(u) = T(w)$ for any $u, w \in h1_v$ implies that v is in a complete subtree released by IMPROVEMENT.

Now we show that there is no other case. Suppose for contradiction that there are two complete subtrees $T(u), T(w)$ such that $T(u), T(w)$ are contained in T_v. Note that $T(u), T(w) \subseteq T_v$. Without loss of generality, we can assume that $T(w)$ and $T(u)$ were released consecutively. Then one of $T(u)$ and $T(w)$ was released by applying release$_1$. This contradicts $X^* \cap h1_v = \emptyset$. □

From Corollary 4.6, we can estimate the width of levels in the resultant structure. Let $C^k(j,\ell)$ be the width of level ℓ in unreleased complete subtree of depth j, i.e., the width of level ℓ in the distance structure $D(A, T_{k,j})$, where A is the vertices of height 1 or 2 in $T_{k,j}$. And let $S_i^k(j,\ell)$ be the width of level ℓ in complete subtree of depth j released by release$_i$, i.e., the width of level ℓ in the distance structure $D(\{v\}, T_{k,j})$, where v is a vertex of height 1 in $T_{k,j}$. Then

$$D^*(\ell) = D^0(\ell) + \sum_{i \in \{0,1\}} \sum_{j=3k/2+a+1}^{d-\mu(k,d)+1} q_i^k(j,d)\left(S_i^k(j,\ell) - C^k(j,\ell)\right).$$

For convenience we denote $P_i^k(d,\ell)$ and $C^k(d,\ell)$ such that

$$P_i^k(d,\ell) = \sum_{j=3k/2+a+1}^{d-\mu(k,d)+1} q_i^k(j,d)\cdot S_i^k(j,\ell),$$

$$C^k(d,\ell) = D^0(\ell) - \sum_{i\in\{0,1\}}\sum_{j=3k/2+a+1}^{d-\mu(k,d)+1} q_i^k(j,d)\cdot C^k(j,\ell).$$

Then we can restate the width of D^*.

Proposition 4.7. $D^*(\ell) = P_0^k(d,\ell) + P_1^k(d,\ell) + C^k(d,\ell)$.

We can easily verify that the following two lemmas hold (See also Fig. 1).

Lemma 4.8.

$$S_0^k(j,\ell) = \begin{cases} k^{\lfloor(\ell-1)/2\rfloor}, & (1\le\ell\le j+1)\\ k^{\lfloor(\ell-1)/2\rfloor} - k^{\ell-j-2}, & (j+2\le\ell\le 2j+1)\\ 0. & (otherwise)\end{cases}$$

Lemma 4.9.

$$S_1^k(j,\ell) = \begin{cases} 1, & (\ell=1)\\ k+1, & (\ell=2)\\ k^{\lfloor\ell/2\rfloor}, & (3\le\ell\le j)\\ k^{\lfloor\ell/2\rfloor} - k^{\ell-j-1}, & (j+1\le\ell\le 2j)\\ 0. & (otherwise)\end{cases}$$

Now, we have the exact value of $S_i^k(j,\ell)$. In what follows, more simpler upper bound is sufficient.

Corollary 4.10.

$$S_0^k(j,\ell) \le \begin{cases} k^{\lfloor(\ell-1)/2\rfloor}, & (1\le\ell\le 2j+1)\\ 0, & (otherwise)\end{cases}$$

$$S_1^k(j,\ell) \le \begin{cases} k+1, & (\ell=1,2)\\ k^{\lfloor\ell/2\rfloor}, & (3\le\ell\le 2j)\\ 0. & (otherwise)\end{cases}$$

Lemma 4.11.

$$P_0^k(d,\ell) \le \begin{cases} 2k^{d-\mu(k,d)}, & (1\le\ell\le 2d-2\mu(k,d)-1)\\ k^{d-\mu(k,d)}, & (\ell = 2d-2\mu(k,d))\\ k^{d-\mu(k,d)+1}, & (2d-2\mu(k,d)+1\le\ell\le 2d-2\mu(k,d)+3)\\ 0. & (otherwise)\end{cases}$$

Lemma 4.12.

$$P_1^k(d,\ell) \le \begin{cases} 2k^{d-\mu(k,d)} - k^{d-\mu(k,d)-1}, & (1\le\ell\le 2d-2\mu(k,d)-1)\\ k^{d-\mu(k,d)+1} - k^{d-\mu(k,d)}, & (\ell = 2d-2\mu(k,d))\\ 0. & (otherwise)\end{cases}$$

Lemma 4.13.

$$P_0^k(d,\ell) + P_1^k(d,\ell) \le \begin{cases} k^{d-\mu(k,d)+1} - k^{d-\mu(k,d)-1}, & (1\le\ell\le 2d-2\mu(k,d)-1)\\ k^{d-\mu(k,d)+1}, & (2d-2\mu(k,d)\le\ell\le 2d-2\mu(k,d)+3)\\ 0. & (otherwise)\end{cases}$$

Proposition 4.14.

$$C^k(j,\ell) = \begin{cases} k^j + k^{j-1}, & (\ell = 1) \\ k^{j-\ell}, & (2 \leq \ell \leq j) \\ 0. & (otherwise) \end{cases}$$

Lemma 4.15.

$$C^k(d,\ell) \leq \begin{cases} k^{d-\mu(k,d)-1}, & (3k/2 + a + 2 \leq \ell \leq d) \\ 0. & (otherwise) \end{cases}$$

Corollary 4.16. $pdw_{D^*}(T_{k,d}) \leq k^{d-\mu(k,d)+1}$.

Theorem 4.17. $pdw(T_{k,d}) \leq k^{d-\mu(k,d)+1}$.

Theorem 4.18. *The ratio between upper bound in Theorem 4.17 and lower bound in Theorem 3.7 is at most* $k^2 + 1$ *for* $d \geq k^2$.

5 Conclusion

We showed upper and lower bounds for path distance width of complete k-ary trees for even numbers $k \geq 4$. By performing a similar calculation, it can be shown that for $k = 2$ (i.e. complete binary trees) and $m \geq 3$

$$2^{2^m} \leq \mathrm{pdw}(T_{2,2^m+m}) \leq \frac{17}{16} 2^{2^m}.$$

References

[1] H.L. Bodlaender, M.R. Fellows, and M.T. Hallett. Beyond *NP*-completeness for problems of bounded width: Hardness for the *W* hierarchy (extended abstract). In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing*, pages 449–458, 1994.

[2] T. Calamoneri, A. Massini, L'. Török, and I. Vrt'o. Antibandwidth of complete k-ary trees. *Electronic Notes in Discrete Mathematics*, 24:259–266, 2006.

[3] T. Calamoneri, A. Massini, and I. Vrt'o. New results on edge-bandwidth. *Theoretical Computer Science*, 307:503–513, 2003.

[4] K. Edwards. The harmonious chromatic number of complete r-ary trees. *Discrete Mathematics*, 203:83–99, 1999.

[5] Y. Otachi and K. Yamazaki. A lower bound for the vertex boundary-width of complete k-ary trees. *Discrete Mathematics*, to appear.

[6] L. Smithline. Bandwidth of the complete k-ary tree. *Discrete Mathematics*, 142:203–212, 1995.

[7] W. Unger. The complexity of the approximation of the bandwidth problem (extended abstract). In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, pages 82–91, 1998.

[8] K. Yamazaki. On approximation intractability of the path-distance-width problem. *Discrete Applied Mathematics*, 110:317–325, 2001.

[9] K. Yamazaki, H.L. Bodlaender, B. De Fuiter, and D.M. Thilikos. Isomorphism for graphs of bounded distance width. *Algorithmica*, 24(2):105–127, 1999.

Appendix A Relationship between $\mu(k,d)$ and $g(k,m)$

Lemma Appendix A.1. *Let d be an integer such that $g(k,m) \leq d < g(k,m+1)$. Then $\mu(k,d) = m$ for any integer $m \geq 0$.*

Proof. As $g(k,m)$ is a monotonically increasing function, it is sufficient to show that $\mu(k,g(k,m)) = m$ and $\mu(k,g(k,m+1)-1) = m$.

First we show $\mu(k, g(k, m)) = m$. From Proposition Appendix A.3,

$$\lfloor \log_k(f(k)g(k,m)) \rfloor = \lfloor \log_k (k^m - 1 + f(k)(m+1)) \rfloor = m.$$

Thus,

$$\begin{aligned}\mu(k, g(k,m)) &= \lfloor \log_k (f(k)(g(k,m) - \lfloor \log_k(f(k)g(k,m)) \rfloor)) \rfloor \\ &= \lfloor \log_k (f(k)(g(k,m) - m)) \rfloor = \lfloor \log_k (k^m - 1 + f(k)) \rfloor \\ &= m. \qquad (\because \text{Proposition Appendix A.2})\end{aligned}$$

Next, we show $\mu(k, g(k, m+1) - 1) = m$. From Proposition Appendix A.4,

$$\lfloor \log_k(f(k)(g(k, m+1) - 1)) \rfloor = \left\lfloor \log_k(k^{m+1} - 1 + f(k)(m+1)) \right\rfloor = m + 1.$$

So, we have

$$\begin{aligned}\mu(k, g(k, m+1) - 1) &= \lfloor \log_k (f(k)(g(k, m+1) - 1 - \lfloor \log_k(f(k)(g(k, m+1) - 1)) \rfloor)) \rfloor \\ &= \lfloor \log_k (f(k)\,(g(k, m+1) - m - 2)\,) \rfloor = \left\lfloor \log_k(k^{m+1} - 1) \right\rfloor \\ &= m.\end{aligned}$$

□

Proposition Appendix A.2. $1 \leq f(k) < 2$.

Proposition Appendix A.3. $k^m \leq k^m - 1 + f(k)(m+1) < k^{m+1}$.

Proof. From Proposition Appendix A.2, $k^m - 1 + f(k)(m+1) \geq k^m + m \geq k^m$. Assume for contradiction that $k^{m+1} \leq k^m - 1 + f(k)(m+1)$. It implies $(k-1)k^m < f(k)(m+1)$. Then we have the following contradiction.

$$\begin{aligned}(k-1)k^m &< f(k)(m+1) = \frac{2(k-1)(m+1)}{k}, \\ k^{m+1} &< 2(m+1), \\ m &< \log_k(m+1).\end{aligned}$$

□

Proposition Appendix A.4. $k^{m+1} \leq k^{m+1} - 1 + f(k)(m+1) < k^{m+2}$.

Proof. The proof is almost the same as the proof of Proposition Appendix A.3. □