Minimum Restricted Diameter Spanning Trees

Refael Hassin and Asaf Levin

Department of Statistics and Operations Research,
Tel-Aviv University, Tel-Aviv 69978, Israel,
{hassin,levinas}@post.tau.ac.il

Abstract. Let $G = (V, E)$ be a requirements graph. Let $d = (d_{ij})_{i,j=1}^{n}$ be a length metric. For a tree T denote by $d_T(i, j)$ the distance between i and j in T (the length according to d of the unique $i - j$ path in T). The restricted diameter of T, D_T, is the maximum distance in T between pair of vertices with requirement between them. The minimum restricted diameter spanning tree problem is to find a spanning tree T such that the minimum restricted diameter is minimized. We prove that the minimum restricted diameter spanning tree problem is NP-hard and that unless $P = NP$ there is no polynomial time algorithm with performance guarantee of less than 2. In the case that G contains isolated vertices and the length matrix is defined by distances over a tree we prove that there exist a tree over the non-isolated vertices such that its restricted diameter is at most 4 times the minimum restricted diameter and that this constant is at least $3\frac{1}{2}$. We use this last result to present an $O(log(n))$-approximation algorithm.

1 Introduction

Let $G = (V, E)$ be a *requirements graph* with $|V| = n$, $|E| = m$. Let $d = (d_{ij})_{i,j=1}^{n}$ be a length metric.

For a tree T denote by $d_T(i, j)$ the distance between i and j in T (the length according to d of the unique $i - j$ path in T).

For a spanning tree T, define the *restricted diameter of* T as

$$D_T = Max_{(i,j)\in E} d_T(i, j).$$

The MINIMUM RESTRICTED DIAMETER SPANNING TREE PROBLEM is to find a spanning tree T that minimizes D_T.

When G is a complete graph the problem is that of finding minimum diameter spanning tree. This problem is solvable in $O(mn + n^2 \log n)$ time (see [7]).

The minimum diameter spanning tree problem is motivated by the following (see for example [9]): we want to find a communication network among the vertices, where the communication delay is measured in terms of the length of a shortest path between the vertices. A desirable communication network is naturally one that minimizes the diameter. To keep the routing protocols simple, often the communication network is restricted to be a spanning tree. The minimum restricted diameter spanning tree problem arises when communication

K. Jansen et al. (Eds.): APPROX 2002, LNCS 2462, pp. 175–184, 2002.

takes place only between a specified collection of pairs of vertices. In such a case it is natural to minimize the maximum communication delay between vertices that need to communicate.

The case, in which G is a clique over a subset $S \subseteq V$ is similarly solved ([13]) by finding the shortest paths tree from the weighted absolute 1-center where the weight of a vertex in S is 1 and the weight of a vertex not in S is 0.

As observed in [1], there are cases where $D_T = \Omega(n) Max_{(i,j)\in E} d_{i,j}$ for any spanning tree T. Therefore, in the analysis of an approximation algorithm for the problem we need to use a better lower bound than $Max_{(i,j)\in E} d_{i,j}$.

We prove that the MINIMUM RESTRICTED DIAMETER SPANNING TREE PROBLEM is NP-hard and that if $P \neq NP$ there is no polynomial time algorithm with performance guarantee of less than 2.

Suppose that $V = V_R \cup V_S$ and $E \subseteq V_R \times V_R$: In [5] it is shown that the *distortion* of tree metric with respect to a steiner tree metric is bounded by a factor of 8. This means that for every spanning tree ST over V there is a tree T over V_R such that for all $i, j \in V_R$ $d_{ST}(i,j) \leq d_T(i,j) \leq 8d_{ST}(i,j)$. By this result we conclude that in particular $D_T \leq 8D_{ST}$. In this paper we will provide a better construction with respect to the restricted diameter criteria, which proves that $D_T \leq 4D_{ST}$. We also show that the best possible bound is at least $3\frac{1}{2}$.

We use this last result to present an $O(\log n)$-approximation algorithm.

A similar problem, MINIMUM COMMUNICATION SPANNING TREE (MCT), was addressed in other papers. In the MCT problem we are given a requirement graph $G = (V, E)$ and a length matrix d and the goal is to find a spanning tree T that minimizes $\Sigma_{(i,j)\in E} d_T(i,j)$. In [4] a derandomization procedure to Bartal's tree metric construction (see [2], [3]) is used in order to obtain a deterministic $O(\log n \ \log\log n)$-approximation algorithm for the general metric case. In [11] an $O(\log n)$-approximation algorithm is presented for the k-dimensional Euclidean complete graphs where k is a constant.

Another similar problem, MINIMUM ROUTING TREE CONGESTION (MRTC), was addressed in other papers. In the MRTC problem we are given a requirement graph $G = (V, E)$ and a weight function $w : E \rightarrow N$. We want to find a routing tree $T = (V', E')$ (a tree is a routing tree if the leaves of T correspond to V and each internal vertex has degree 3) that minimizes

$$Max_{e\in E'} \Sigma_{(u,v)\in E, u\in S(e), v\notin S(e)} w(u,v)$$

where $S(e)$ is one of the connected components resulting from T by deleting an edge $e \in E'$. In [12] the MRTC is proved to be NP-hard and the special case when G is planar, the problem can be solved optimally in polynomial time. In [10] an $O(\log n)$-approximation algorithm is given for the general case (non-planar graphs).

2 NP-Hardness

Theorem 1. *Unless $P = NP$ there is no polynomial-time approximation algorithm for the restricted diameter spanning tree problem with performance guarantee of less than 2.*

Proof. We describe a reduction from MONOTONE SATISFIABILITY (mSAT) (in which in every clause either all the literals are variables or all the literals are negated variables). mSAT is NP-complete (see [6]). Consider an instance of the mSAT problem composed of variables $x_1, x_2, \ldots, x_n$ and of clauses $c_1, c_2, \ldots c_m$. We construct an instance for the minimum restricted diameter spanning tree problem as follows: Define a vertex set $V = V_1 \cup V_2 \cup V_3$ where $V_1 = \{root\}$, $V_2 = \{x_i, \bar{x}_i, x'_i | 1 \leq i \leq n\}$ and $V_3 = \{c^1_j, c^2_j | 1 \leq j \leq m\}$. Define a length matrix as follows: $d_{root,x_i} = d_{root,\bar{x}_i} = 1 \ \forall i$, $d_{root,c^1_j} = d_{root,c^2_j} = 2 \ \forall j$, $d_{x_i,x'_i} = d_{x'_i,\bar{x}_i} = 2 \ \forall i$, $d_{x_i,c^1_j} = d_{x_i,c^2_j} = 1$ if $x_i \in c_j$, $d_{\bar{x}_i,c^1_j} = d_{\bar{x}_i,c^2_j} = 1$ if $\bar{x}_i \in c_j$ and otherwise d is defined as the shortest path length according to the defined distances. Define a requirement graph $G = (V, E)$ by $E = \{(x_i, x'_i), (x'_i, \bar{x}_i) | 1 \leq i \leq n\} \cup \{(root, c^1_j), (c^1_j, c^2_j), (root, c^2_j) | 1 \leq j \leq m\}$.

The instance for the minimum restricted diameter spanning tree problem has solution which is at most 2 if and only if the mSAT instance is satisfiable. If there is a spanning tree with restricted diameter 2 then for every i the spanning tree must include the edges $(x_i, x'_i), (x'_i, \bar{x}_i)$ as there is no other path of length 2 between x_i and x'_i and between x'_i and $\bar{x}_i$. Therefore, the tree may include only one of the edges $(root, x_i)$ or $(root, \bar{x}_i)$. In order to have a path of length 2 between $root$ and c^1_j, a path of length 2 between $root$ and c^2_j and a path of length 2 between c^1_j and c^2_j the tree must have some i such that all of $(root, x_i), (x_i, c^1_j), (x_i, c^2_j)$ belong to the tree and such that $x_i \in c_j$, or the tree must include $(root, \bar{x}_i), (\bar{x}_i, c^1_j), (\bar{x}_i, c^2_j)$ such that $\bar{x}_i \in c_j$. Therefore, if the tree includes the edge $(root, x_i)$ we set x_i to TRUE, and otherwise to FALSE. Since the tree has paths of length 2 between $root$, c^1_j and c^2_j for every j then c_j must have a literal with TRUE assignment, and therefore the formula is satisfied.

Suppose that the formula can be satisfied and consider the tree composed of the edges $(x_i, x'_i), (x'_i, \bar{x}_i)$ for $i = 1, \ldots, n$, the edge $(root, x_i)$ for every TRUE assigned variable, and the edge $(root, \bar{x}_i)$ for every FALSE assigned variable. Every clause c_j must include a literal with TRUE value. Pick one of them and add an edge between this literal and c^1_j, and an edge between this vertex and c^2_j. Then the restricted diameter of this spanning tree is 2.

We note that for this problem if there is no solution of cost 2 then the minimum cost is at least 4. Therefore, distinguishing between 2 and 4 is NP-complete. Therefore, if $P \neq NP$ there is no polynomial time approximation algorithm with performance guarantee better than 2.

3 The Role of Steiner Points

In this section we assume that $V = V_R \cup V_S$ where $V_R \cap V_S = \emptyset$ and $E \subseteq V_R \times V_R$, ($V_S$ the Steiner point set and V_R the regular point set). Denote by ST the minimum restricted diameter spanning tree (a spanning tree over V). We will prove that there is a spanning tree T over V_R such that $D_T \leq 4D_{ST}$ and that there are cases where $D_T \geq 3\frac{1}{2}D_{ST}$.

Theorem 2. *There exists a spanning tree T that can be computed in polynomial time, such that $D_T \leq 4D_{ST}$.*

Proof. It is sufficient to prove the claim under the assumption that G is connected. If G is not connected then we can construct a tree over every component with restricted diameter at most $4D_{ST}$ and connect these trees arbitrary. Therefore, w.l.o.g. assume that G is connected.

W.l.o.g. we assume that all the leaves of ST are in V_R. This is so since we can remove any leaf $u \in V_S$ of ST from G without affecting D_{ST}.

For a vertex $p \in V$ denote by $t(p) = argmin_{u \in V_R} d_{ST}(u,p)$ ($t(p) = p$ if $p \in V_R$). Define a set of vertices U by the following procedure:

1. Arbitrarily choose $r \in V_S$. Add r to U, root ST at r. Set $i = 0$, and label r 'unvisited' $level(r) = 0$.
2. While there is an 'unvisited' non-leaf vertex w do:
   – Pick an arbitrary non-leaf vertex $w \in U$ with label 'unvisited' and $level(w) = i$ (if there is no such vertex set $i = i + 1$):
   – Mark w 'visited'.
   – Along every path going down the tree from w to a leaf of ST that does not include a vertex $v \in U$ $level(v) = i$ there is a vertex, u, which is the last vertex along this path such that $d_{ST}(t(u), t(w)) \leq D_{ST}$. Add u to U with label 'unvisited' and $level(u) = i + 1$.

Step 2 is well defined (the vertex u exists): For every $w \in U$ and v a son of w, since G is connected, there is a requirement crossing the cut corresponding to $ST \setminus \{(w,v)\}$, and therefore $d_{ST}(t(v), t(w)) \leq D_{ST}$.

We now define a spanning tree T. For every $v \in V_R \setminus \{t(r)\}$ there is a vertex $u \in U$ on the path from v to r such that $t(u) \neq v$ and we pick the closest vertex u to v satisfying these conditions. We add to the edge set of T the edge $(v, t(u))$. This defines a spanning tree T. This is so as there are $|V_R| - 1$ edges and every vertex $v \in V_R$ is connected to a vertex $t(u)$ and $t(u)$ is either $t(r)$ or it is connected to a vertex $t(w)$ such that w is an ancestor of u in ST. Therefore, all the vertices are connected by paths to $t(r)$ and T is connected.

Consider an edge $(v, t(u))$ in T. By step 2 u is on the path between v and r and between v and u there is no other vertex $p \in U$ unless $v = t(p)$. By the construction of U $d_{ST}(v, t(u)) \leq D_{ST}$.

To complete the proof we prove that if $d_{ST}(u,v) \leq D_{ST}$ for $u, v \in V_R$ then the path between u and v in T has at most 4 edges. We assume otherwise. Consider T as a rooted tree at $t(r)$. We use the following observations:

– For $u, v \in V_R$ a path in T between u and v goes up the tree until their common ancestor and then goes down the tree.
– Suppose that the path from u to v in T consists of at least 5 edges. W.l.o.g. assume that it goes up the tree in at least 3 edges $(u, t(a))$, $(t(a), t(b))$, $(t(b), t(c))$.
– In ST the path between u and v goes through b. Therefore, $d_{ST}(u,v) = d_{ST}(u,b) + d_{ST}(b,v)$.
– $v \in V_R$ and by the definition of $t(b)$ $d_{ST}(v,b) \geq d_{ST}(t(b),b)$.
– In Step 2 a vertex is added to U only if it is the last vertex on the path to a leaf satisfying the conditions. Therefore, $d_{ST}(t(b), u) > D_{ST}$.

These observations together with the triangle inequality yield a contradiction to the definition of D_{ST}:

$$d_{ST}(u,v) = d_{ST}(u,b)+d_{ST}(b,v) \geq d_{ST}(u,b)+d_{ST}(t(b),b) \geq d_{ST}(u,t(b)) > D_{ST}.$$

Remark 1. Figure 1 shows that the construction in Theorem 2 cannot lead to a better asymptotic ratio. In this example $V_S = \{a,b,c,d\}$, and G contains the edges $\{(u,t(a)),\ (u,t(d)),\ (u,v),\ (v,t(a)),\ (v,t(d)),\ (t(a),t(d)),\ (t(a),t(b)),\ (t(d),t(b))\}$. T contains the edges $\{(u,t(a)),\ (t(a),t(b)),\ (t(b),t(d)),\ (t(d),v)\}$, and therefore $D_{ST} = 2+2\epsilon$ and $D_T = 8+2\epsilon$.

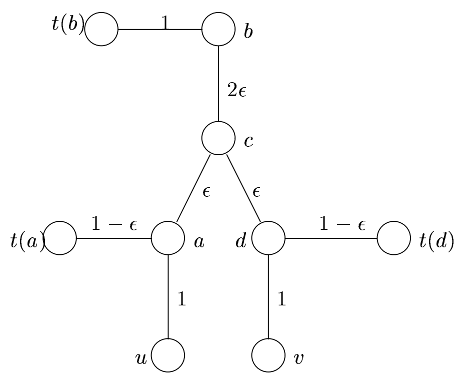

Fig. 1. Bad example for the construction in Theorem 2

We now prove a lower bound on the best possible constant in Theorem 2.

Theorem 3. *For any $\epsilon > 0$ there exists a requirement graph $G = (V,E)$ such that, $V = V_R \cup V_S$ $E \subseteq V_R \times V_R$, a metric d and a Steiner tree ST over V such that for any tree T over V_R, $D_T \geq (3\frac{1}{2} - \epsilon)D_{ST}$.*

Proof. Consider the following family of instances: ST contains a rooted (at a vertex *root*) complete binary tree with at least $16K^2$ levels, and the length of these edges is 1. V_S consists of the vertices of this binary tree. Every vertex of V_S is connected in ST to a distinct pair of vertices of V_R by edges of length $K - \frac{\alpha}{2}$. We will start at *root* with $\alpha = 2$ and every $16K$ levels we will increase α by 2. $D_{ST} = 2K$ and there is a requirement between a pair of vertices from V_R if and only if the distance between them in ST is at most $2K$. The metric will be defined as the distance in ST between the vertices, that is, $d_{ij} = d_{ST}(i,j)$ for every $i,j \in V$. See for example Figure 2.

For $u,v \in V_R$, $(u,v) \in E$ if and only if $d_{ST}(u,v) \leq 2K$. Therefore $D_{ST} = 2K$.

Due to space limitations the proof that for every spanning tree T $D_T > (3\frac{1}{2} - \epsilon)2K$ is omitted.

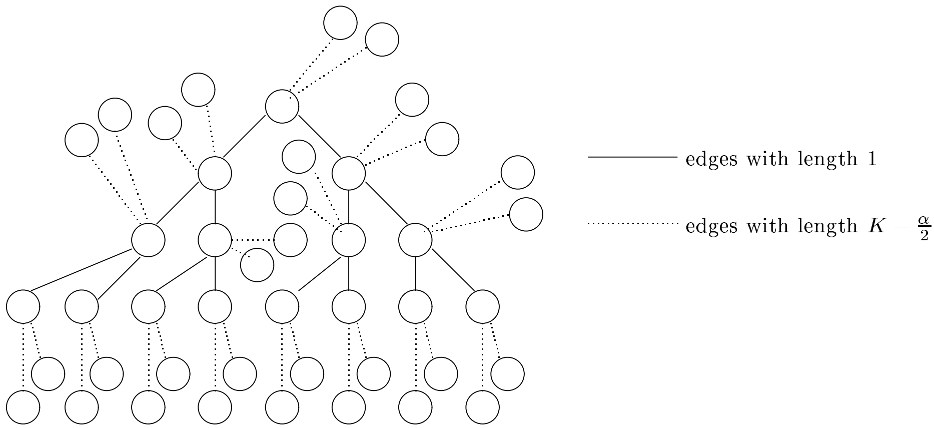

Fig. 2. The structure of ST

Theorem 4. *Suppose that $G = (V, S \times S)$ for a subset $S \subseteq V$. Then Theorem 2 and Theorem 3 hold with both constants 4 and $3\frac{1}{2}$ replaced by 2.*

Proof. The upper bound of 2 is a result of the following argument: Let r be a regular vertex in ST. D_{ST} is at least the distance between r and any of the non-steiner vertices in ST. Take a tree T which is a star with a center in r. Then by the triangle inequality its cost is at most twice the cost of ST.

The lower bound is shown by the following example: Let ST be a star with the length of all the edges be 1. Let S be defined as the set of all the star leaves. All the non-ST edges has length 2 and any tree T not containing the star's center has cost at least 4 which is $2D_{ST}$.

4 Approximation Algorithm

In this section we provide an $O(\log n)$ approximation algorithm for the MINIMUM RESTRICTED DIAMETER SPANNING TREE PROBLEM.

Denote by T^* a tree that achieves the optimal restricted diameter and denote $D_{T^*} = D^*$.

The following lemma identifies a lower bound and an upper bound on D^*.

Lemma 1. *Assume that G is connected. Let MST be a minimum spanning tree with respect to the length matrix d, and let $(i, j) \in MST$ be the longest edge in MST. Then $d_{ij} \leq D^* \leq (n-1)d_{ij}$.*

Proof. A path in MST contains at most $n - 1$ edges. Since (i, j) is the longest edge in MST it follows that, $D^* \leq D_{MST} \leq (n-1)d_{ij}$. To see the lower bound let (I, J) be the cut induced by $MST \setminus \{(i, j)\}$. Since G is connected it contains an edge in (I, J), therefore, $D^* \geq min\{d_{k,l} | k \in I, l \in J\} = d_{ij}$.

Corollary 1. *Let* $A = \{d_{ij}, 2d_{ij}, 4d_{ij}, 8d_{ij}, \ldots, 2^{\lceil \log n \rceil} d_{ij}\}$ *then there exists* $\bar{D} \in A$ *such that* $D^* \leq \bar{D} \leq 2D^*$.

We will present an algorithm that for a given test value D' either finds a spanning tree T' such that $D_{T'} = O(D' \log n)$ or concludes that $D^* > D'$. By applying this algorithm for every $D' \in A$ we get an $O(\log n)$-approximation.

We will use the following *decomposition procedure* with a fixed vertex u and test value D'. Identify the vertices $V^u(D') = \{v \in V | d_{u,v} \leq D'\}$. To simplify notations we will denote $V^u = V^u(D')$. Let $E(V^u) = E \cap (V^u \times V^u)$. Let $C^1, C^2, \ldots, C^r$ be the connected components of $G \setminus E(V^u)$ which are not singletons (see Figure 3).

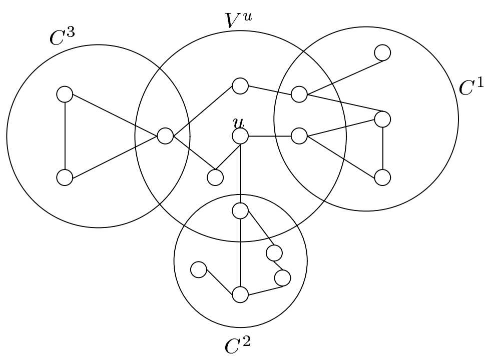

Fig. 3. Decomposition procedure at u

Definition 1. *A* D'*-center is a vertex* $u \in V$ *such that the decomposition procedure with* u *and* D' *forms connected components with at most* $\frac{n}{2}$ *vertices.*

Lemma 2. *Assume* $D' \geq D^*$*, then a* D'*-center* u *exists.*

Proof. Every tree has a centroid, a vertex whose deletion leaves no subtree containing more than $\frac{n}{2}$ vertices (proved by Jordan in 1869, see for example [8]). Let u be a centroid of T^*. We will show that u is a D^*-center (and therefore it is also a D'-center for every $D' \geq D^*$). In T^* every path connecting vertices from $V \setminus V^u$ that belong to different sides of $T^* \setminus \{u\}$ is of length at least $2D^*$. For vertices w and v that belong to distinct sides of $T^* \setminus \{u\}$, $d_{T^*}(w, v) \geq 2D^*$. Therefore, every connected component C^i of $G \setminus E(V^u)$ is fully contained in some side of $T^* \setminus \{u\}$. It follows that $|C^i| \leq \frac{n}{2}$.

We propose to approximate the problem by Algorithm Restricted_Diameter (Figure 4). l is the recursive level of the algorithm ($0 \leq l \leq \lceil \log n \rceil$). The clusters define the connected components in a partial solution (a forest) obtained by previous levels. The application of Algorithm Restricted_Diameter in Figure 4 with parameters $\bar{D}$, $l = 0$, G, d and $V_i' = \{v_i\}$ $\forall i$ will result an $O(\log n)$-approximation.

The algorithm first finds a D'-center u and adds to the solution the edges connecting u to $V^u \setminus \{u\}$ without closing cycles with previously existing edges. It then solves recursively for every connected component of $G \setminus V^u$. The algorithm uses the information in the partition into clusters to ensure that the solution in each phase doesn't contain cycles. It returns the union of the solutions resulted from the recursive calls and the edges connecting u to $V^u \setminus \{u\}$ without edges that close cycles.

For a formal statement of the algorithm, we need the following notations. For a graph $G' = (V', E')$ and $U \subseteq V'$ the induced subgraph of G' over U will be denoted by $G'(U)$ and the length matrix induced by $U \times U$ will be denoted by $d(U)$.

Denote by $D(V_j')$ the T-diameter (regular diameter and not restricted one) of cluster V_j' in the final solution tree T $(D(V_j') = Max_{v,w \in V_j'} d_T(v,w))$. The following holds throughout the algorithm:

Lemma 3.
1) Each component returned by the decomposition procedure at level l of Algorithm Restricted_Diameter has size at most $\frac{n}{2^l}$.
2) Let C be a component returned by the decomposition procedure at level l. Then,

$$\Sigma_{j:V_j' \cap C \neq \emptyset} D(V_j') \leq 2lD'.$$

Proof. The first property holds by induction because u is chosen in each iteration to be a D'-center in a graph induced by a component of the previous level.

The second property holds also by induction over the levels of iterations as follows:

For level $l = 0$ all the clusters are singletons and therefore, have zero diameter and the property holds.

Assume the property holds for the previous levels and we will prove it for l:

The only affected clusters in iteration l are the ones that intersect V^u. These clusters are all replaced with a new cluster that has diameter of at most $2D'$ plus the sum of all the diameters of the original clusters. Therefore, the property holds for l as well.

Theorem 5. *Algorithm Restricted_Diameter applied with D'-values which result from binary search over A is an $8 \log n + 4$-approximation algorithm for the minimum restricted diameter spanning tree problem. Its running time is $O(n^3 loglogn)$.*

Proof. Consider a pair of vertices with requirement between them. They both belong to V^u in some level l of the algorithm. Let C be their component in this level.

By Lemma 3 the diameter of their cluster which is an upper bound over their distance in T is at most $2lD$ and $l \leq \log n + 1$.

The presentation of the algorithm assumes the graph is connected but for non-connected graphs vertices from other components may only serve as Steiner points and therefore, by Theorem 2, after multiplying by another factor of 4 one obtains the desired result.

To see the complexity of the algorithm note that using binary search over A we test only $O(\log\log n)$ D' values. It remains to show that each value can be tested in $O(n^3)$ time. To see this note that finding a center can be done in $O(n^3)$ time (by trying all the vertices as candidates to be a center, each candidate u is tested using *BFS* on $G \setminus E(V^u)$ in $O(n^2)$ time). Denote by n_i the number of vertices in C^i then the running time of a test value satisfies the following recursive

Restricted_Diameter
  input
  Integers D' *and* l.
  Connected graph $G' = (V', E')$.
  Length matrix d.
  A partition of V' *into clusters,* $P = \{V'_1, V'_2, \ldots, V'_k\}$.
  returns
  A spanning tree T.
  begin
  $T := (V', \emptyset)$.
  $V'' := \emptyset$.
  if $|V'| = 1$
    then
      return T.
    else
      $u :=$ *a* D'*-center of* G (*if it does not exist conclude* $D' < D^*$).
      for every $i = 1, 2, \ldots, k$
        if $V^u \cap V'_i \setminus \{u\} \neq \emptyset$
          then
            Choose $u_i \in V^u \cap V'_i,\ u_i \neq u$.
            $T := T \cup \{(u, u_i)\}$.
            $V'' := V'' \cup V'_i$.
            $P := P \setminus V'_i$.
          end if
      $P := P \cup V''$.
      $C^1, C^2, \ldots C^r :=$ *components returned by the decomposition procedure when applied to* (G', D', u).
      for every $i \in \{1, 2, \ldots, r\}$
        $P_i := \{U \cap C^i | U \in P,\ U \cap C^i \neq \emptyset\}$.
        $T_i :=$ *Restricted_Diameter*$(D', l+1, G(C^i), d(C^i), P_i)$.
      $T := T \cup T_1 \cup \cdots \cup T_r$.
    end if
  return T.
  end *Restricted_Diameter*

Fig. 4. Algorithm Restricted_Diameter

relation: $T(n) \leq cn^3 + \Sigma_{i=1}^{r} T(n_i)$ where $n_i \leq \frac{n}{2}$ $\forall i$ and $\Sigma_{i=1}^{r} n_i \leq n$. $\Sigma_{i=1}^{r} T(n_i)$ is maximized when $r = 2$ and $n_1 = n_2 = \frac{n}{2}$, and therefore $T(n) \leq 2cn^3$.

References

1. N. Alon, R. Karp, D. Peleg and D. West, "A graph theoretic game and its application to the k-server problem", *SIAM J. Comput.*, **24**, 78-100, 1995.
2. Y. Bartal, "Probabilistic approximation of metric spaces and its algorithmic applications", *Proceedings of FOCS 1996*, pages 184-193.
3. Y. Bartal, "On approximating arbitrary metrics by tree metrics", *Proceedings of STOC 1998*, pages 161-168.
4. M. Charikar, C. Chekuri, A. Goel and S. Guha, "Rounding via trees: deterministic approximation algorithms for group steiner trees and k-median", *Proceedings of STOC 1998*, pages 114-123.
5. A. Gupta, "Steiner points in tree metrics don't (really) help", *Proceeding of SODA 2001*, pages 220-227.
6. M. R. Garey and D. S. Johnson, "Computers and intractability: a guide to the theory of NP-completeness", W.H. Freeman and Company, 1979.
7. R. Hassin and A. Tamir, "On the minimum diameter spanning tree problem", *IPL*, **53**, 109-111, 1995.
8. F. Harary, "Graph theory", Addison-Wesley, 1969.
9. J.-M. Ho, D. T. Lee, C.-H. Chang, and C. K. Wong, "Minimum diameter spanning trees and related problems", *SIAM J. Comput.*, **20 (5)**, 987-997, 1991.
10. S. Khuller, B. Raghavachari and N. Young, "Designing multi-commodity flow trees", *IPL*, **50**, 49-55, 1994.
11. D. Peleg and E. Reshef, "Deterministic polylog approximation for minimum communication spanning trees", *Proceedings of ICALP 1998*, pages 670-681.
12. P. Seymour and R. Thomas, "Call routing and the rat catcher", *Combinatorica*, **14**, 217-241, 1994.
13. A. Tamir, *Private communication*, 2001.