

# On Improved Bounds for Bounded Degree Spanning Trees for Points in Arbitrary Dimension

Samuel Zbarsky

Received: 5 November 2012 / Revised: 9 December 2013 / Accepted: 28 December 2013 /  
Published online: 15 January 2014  
© Springer Science+Business Media New York 2014

**Abstract** Given points in Euclidean space of arbitrary dimension, we prove that there exists a spanning tree having no vertices of degree greater than 3 with weight at most 1.559 times the weight of the minimum spanning tree. We also prove that there is a set of points such that no spanning tree of maximal degree 3 exists that has this ratio be less than 1.447. Our central result is based on the proof of the following claim:

Given  $n$  points in Euclidean space with one special point  $v$ , there exists a Hamiltonian path with an endpoint at  $v$  that is at most 1.559 times longer than the sum of the distances of the points to  $v$ .

These proofs also lead to a way to find the tree in linear time given the minimal spanning tree.

**Keywords** Minimum spanning trees · Approximation algorithms · Discrete geometry

## 1 Introduction

The minimum spanning tree (MST) problem in graphs is perhaps one of the most basic problems in graph algorithms. An MST is a spanning tree with minimal sum of edge weights. Efficient algorithms for finding an MST are well known.

One variant on the MST problem is the bounded-degree MST problem, which consists of finding a spanning tree satisfying given upper bounds on the degree of each vertex and with minimal sum of edges weights subject to these degree bounds.

In general, this problem is NP-hard [1], so no efficient algorithm exists. However, there are certain achievable results. For undirected graphs, Singh and Lau [2] found a polynomial-time algorithm to generate a spanning tree with total weight no more than that of the bounded-degree MST and with each vertex having degree at most one

---

S. Zbarsky  
Carnegie Mellon University, 2329 McCormick Rd, Rockville, MD 20850, USA  
e-mail: [sa\\_zbarsky@yahoo.com](mailto:sa_zbarsky@yahoo.com)

greater than that vertex's bound. If the graph is undirected and satisfies the triangle inequality, Fekete et al. [3] bounded the ratio of the total weight of the bounded-degree MST to that of any given tree, with a polynomial-time algorithm for generating a spanning tree satisfying the degree constraints and this ratio bound.

The Euclidean case, with vertices being points in Euclidean space and edge weights being Euclidean distances, also has a rich history. We denote (following Chan in [5]) by  $\tau_k^d$  the supremum, over all sets of points in  $d$ -dimensional Euclidean space, of the ratio of the weight of the bounded-degree MST with all degrees at most  $k$  to the weight of the MST with no restrictions on degrees ( $\tau_k^\infty$  is the supremum of  $\tau_k^d$  over all  $d$ ). For  $k = 2$ , the bounded-degree MST problem becomes the Traveling Salesman Problem and  $\tau_2^d = 2$  [3], thus making  $k = 3$  the first unsolved case.

Papadimitriou and Vazirani [1] showed that finding the degree-3 MST is NP-hard. Khuller, Raghavachari, and Young [4] showed that  $1.104 \approx (\sqrt{2} + 3)/4 \leq \tau_3^2 \leq 1.5$  and  $1.035 < \tau_4^2 \leq 1.25$ . Chan [5] improved the upper bounds to 1.402 and 1.143, respectively. Jothi and Raghavachari [6] showed that  $\tau_4^2 \leq (2 + \sqrt{2})/3 \approx 1.1381$ . It holds  $\tau_5^2 = 1$  since there is always an MST with maximal degree 5 or less [7].

These same papers also studied the problem in higher dimensions. Khuller, Raghavachari, and Young [4] gave an upper bound on  $\tau_3^\infty$  of  $5/3 \approx 1.667$ , which Chan [5] improved to  $2\sqrt{6}/3 \approx 1.633$ . These two followed the same approach, proving these bounds on a certain ratio, which we will call  $r$ . The  $r$  is the maximum ratio between the shortest path through a collection of points starting at a special point and the size of a star centered at that point. It is conjectured to actually be 1.5.

Khuller, Raghavachari, and Young [4] showed that  $\tau_3^\infty \leq r$ . This is achieved in linear time as follows:

1. Root the original tree.
2. Treating the root as  $v$ , find a Hamiltonian path with ratio at most  $r$  through its children.
3. Repeat recursively on each child.

Each vertex then has at most 3 neighbors: two as a child and one as a parent.

We improve previous upper bounds on  $r$ , and thus  $\tau_3^\infty$ , to 1.559. The proof leads to a linear-time algorithm for generating the path and thus the bounded-degree tree. Our approach is based on Chan's but we weigh paths differently and select the number of points to remove when performing the induction based on the distances of points to  $v$ .

We also find, by construction, a nontrivial lower bound of about 1.447 on  $\tau_3^\infty$ .

In Sect. 2, we go over  $r$  a bit more carefully as well as referring to a useful paper and discuss how we will use it. In Sect. 3, we improve the upper bound on  $r$  to 1.559, and in Sect. 6 we improve the lower bound on  $\tau_3^\infty$  to 1.447.

## 2 Preliminaries

The  $r$  is properly defined as follows:

Given point  $v$  and  $m$  points  $a_1, a_2, \dots, a_m$  in a Euclidean space of arbitrary finite dimension, let  $S = \sum_{i=1}^m d(v, a_i)$  and let  $L$  be the length of the shortest possible path

that starts at  $v$  and goes around the other points in some order (it does not go back to  $v$ ). Then  $r$  is the supremum of the possible values of  $L/S$  over all arrangements of points in any number of dimensions. The  $r = 1.5$  is achieved for  $m = 2$  in one dimension by the points  $v = 0, a_1 = 1, a_2 = -1$ .

We use multiple times the results of Young [8] in order to bound certain sums of distances. This paper deals with the maximum of weighted sums (with weights  $w_{i,j}$ ) of lengths between  $n$  points in  $(n - 1)$ -dimensional Euclidean space, given that each point  $a_i$  is specified as being no further than some distance  $l_i$  from the origin.

$$\max\left(\sum_{1 \leq i < j \leq n} w_{i,j} d(a_i, a_j)\right) = \min\left(\sqrt{\sum_{1 \leq i < j \leq n} \frac{w_{i,j}^2}{x_i x_j}} \sqrt{\sum_{i=1}^n l_i^2 x_i} \sqrt{\sum_{i=1}^n x_i}\right) \tag{1}$$

where the maximum is taken over all arrangements of points and the minimum is taken over all nonnegative  $x_i$ .

Furthermore, Young specifies a relationship between the optimal arrangement and the values of  $x_i$  where equality is achieved. Thus one can iteratively approximate the optimal arrangement using the same method as in [9], and then calculate  $x_i$  values from it.

Whenever (1) is used to give an upper bound on some weighted sum of distances, the values for  $x_i$  used are given in Appendix A.

### 3 Main Proof of Upper Bound on $r$

Let  $r^* = 1.559$ . We will prove that  $L \leq r^* S$  (as  $L$  and  $S$  are defined in the Introduction), thus showing that  $r < 1.559$ .

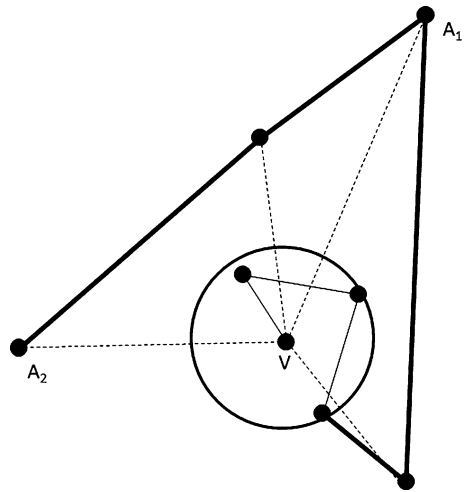
We will prove this by strong induction on the number of points. Given  $m$  vectors  $a_1, a_2, \dots, a_m$  with norms  $d_1 \geq d_2 \geq d_3 \geq \dots \geq d_m > 0$ , respectively, we will try to induct by removing  $a_1, \dots, a_n$  for various values of  $n$ . We will try to traverse the other points, ending at  $a_{n+1}$  or  $a_{n+2}$ . We will then add the removed points, projected onto a sphere, and look at the average length of a path traversing them and ending at  $a_1$  or  $a_2$ . We will then move them in stages, seeing how this average path length changes at each stage, in order to bound the final average path length in terms of the values  $d_k$ . Since the average is an upper bound on the minimum, this gives a linear inequality on the  $d_k$  which is a sufficient condition for the inductive step to work. We then use linear programming to show that one of these inequalities is satisfied and thus that induction is possible. For the algorithm, we will then follow the induction to split the points up into blocks, choose the starting and ending vertex for one block at a time, using brute force to find the shortest path that goes through all the block's points.

We start by defining  $a_k = 0$  and  $d_k = 0$  for all  $k > m$ . Introducing these new points does not affect the distance sum or the traversing path length, as the traversing path can go to them first.

We will prove the following claim:

**Claim 1** *There exist two paths  $P_1$  and  $P_2$  ending at  $a_1$  and  $a_2$ , respectively, such that the average of the lengths of these paths is at most  $r^* S$*

**Fig. 1** The thick segments contribute to  $L(a_n, \dots, a_1)$ ; the dotted segments contribute to  $S_n$



This clearly implies that  $L \leq r^*S$ .

We will proceed by strong induction on  $m$ . To induct, remove  $a_1$  through  $a_n$  (where  $n \geq 3$  may vary), use the inductive hypothesis to find two paths  $P_1$  and  $P_2$  through the other  $m - n$  points, ending at  $a_{n+1}$  and  $a_{n+2}$ , respectively. We will then try to find four paths  $Q_{11}, Q_{12}, Q_{21}, Q_{22}$  with path  $Q_{ij}$  going from  $a_{n+i}$  to  $a_j$  and going through all points  $a_1, \dots, a_n$ , so that the average length of these four paths is at most  $r^*(\sum_{i=1}^n d_i)$ .

We will assume that this is impossible, generate a set of conditions on the values  $d_n$ , then prove that one of the conditions must be violated.

### 3.1 Given $n \geq 3$

In this section, we will assume  $n > 3$  to be a given value. We will select it in Sect. 4.

Let  $S_n = \sum_{i=1}^n d_i$ . Let  $L(u_{n+1}, \dots, u_1)$  be the shortest length of a path  $u_{s_{n+1}}, \dots, u_{s_1}$  where  $s$  is a permutation of  $1, \dots, n + 1$  so that  $s_{n+1} = n + 1$  and  $\{s_1, s_2\} = \{1, 2\}$  (see Fig. 1). Let  $\overline{L(u_{n+1}, \dots, u_1)}$  be the average length over all such paths  $u_{s_{n+1}}, \dots, u_{s_1}$ . Then

$$\begin{aligned}
 L(u_{n+1}, \dots, u_1) &= \frac{1}{n-1}d(u_1, u_2) + \frac{1}{2(n-1)}d(u_1, u_{n+1}) + \frac{1}{2(n-1)}d(u_2, u_{n+1}) \\
 &+ \sum_{i=3}^n \frac{3}{2(n-1)}d(u_1, u_i) + \sum_{i=3}^n \frac{3}{2(n-1)}d(u_2, u_i) \\
 &+ \sum_{i=3}^n \frac{1}{n-1}d(u_i, d_{n+1}) + \sum_{3 \leq i < j \leq n} \frac{2}{n-1}d(u_i, u_j). \tag{2}
 \end{aligned}$$

We wish to find upper bounds on  $\overline{L(a_{n+1}, \dots, a_1)}$  and  $\overline{L(a_{n+2}, a_n, a_{n-1}, \dots, a_1)}$ . For  $1 \leq i \leq n$ , let

$$D_{n,i,1} = L\left(a_{n+1}, \dots, a_{i+1}, a_i, \frac{d_i}{d_{i-1}}a_{i-1}, \dots, \frac{d_i}{d_1}a_1\right)$$

$$- L\left(a_{n+1}, \dots, a_{i+1}, \frac{d_{i+1}}{d_i} a_i, \frac{d_{i+1}}{d_{i-1}} a_{i-1}, \dots, \frac{d_{i+1}}{d_1} a_1\right) \tag{3}$$

and let

$$D_{n,1} = L\left(a_{n+1}, \frac{d_{n+1}}{d_n} a_n, \dots, \frac{d_{n+1}}{d_1} a_1\right). \tag{4}$$

$D_{n,i,2}$  and  $D_{n,2}$  are defined identically, except that  $a_{n+1}$  and  $d_{n+1}$  are replaced with  $a_{n+2}$  and  $d_{n+2}$ . For  $k = 1$  or  $k = 2$ ,

$$\overline{L(a_{n+k}, a_n, a_{n-1}, \dots, a_1)} = D_{n,k} + \sum_{i=1}^n D_{n,i,k}.$$

Intuitively, we are setting all points at distance  $d_{n+k}$ , then moving out  $n$  points to distance  $d_n$ , then moving out  $n - 1$  points, and so on.

We will now find values  $B_{n,i}$  and  $B_n$  independent of the arrangement of  $a_1, a_2, \dots$ , satisfying

$$\begin{aligned} B_{n,i}(d_i - d_{i+1}) &\geq D_{n,i,1}, \\ B_n d_n &\geq D_{n,1}. \end{aligned}$$

The corresponding equations (substituting  $a_{n+2}$  for  $a_{n+1}$  and  $d_{n+2}$  for  $d_{n+1}$ ) will then hold for  $D_{n,i,2}$  and  $D_{n,2}$ .

### 3.1.1 $B_n$

Define  $g(n)$  as the maximum value of

$$\begin{aligned} &d(u_1, u_2) + \frac{1}{2}d(u_1, u_{n+1}) + \frac{1}{2}d(u_2, u_{n+1}) + \sum_{j=3}^n \frac{3}{2}d(u_1, u_j) \\ &+ \sum_{j=3}^n \frac{3}{2}d(u_2, u_j) + \sum_{j=3}^n d(u_j, u_{n+1}) + \sum_{3 \leq j < k \leq n} 2d(u_j, u_k) \end{aligned}$$

over unit vectors  $u_1, \dots, u_{n+1}$ .

We use Eq. (1) to obtain upper bounds on  $g(n)$ , which we then use to find numerical values for  $B_n$ .

Substituting in (4) and (2), we get that

$$D_{n,1} \leq d_n \frac{g(n)}{n - 1},$$

and similarly for  $D_{n,2}$ . Thus we can set

$$B_n = \frac{g(n)}{n - 1}.$$

3.1.2  $B_{n,i}$

For  $i < j < k$ ,

$$d(a_j, a_k) - d(a_j, a_k) = 0. \tag{5}$$

For  $j \leq i < k$ ,

$$d\left(\frac{d_i}{d_j} a_j, a_k\right) - d\left(\frac{d_{i+1}}{d_j} a_j, a_k\right) \leq d_i - d_{i+1}. \tag{6}$$

For  $k < j \leq i$ ,

$$d\left(\frac{d_i}{d_j} a_j, \frac{d_i}{d_j} a_k\right) - d\left(\frac{d_{i+1}}{d_j} a_j, \frac{d_{i+1}}{d_j} a_k\right) \leq (d_i - d_{i+1})d\left(\frac{a_j}{d_j}, \frac{a_k}{d_k}\right). \tag{7}$$

Define also  $f(i)$  as the maximum value of

$$d(u_1, u_2) + \sum_{j=3}^i \frac{3}{2}d(u_1, u_j) + \sum_{j=3}^i \frac{3}{2}d(u_2, u_j) + \sum_{3 \leq j < k \leq i} 2d(u_j, u_k)$$

over unit vectors  $u_1, \dots, u_i$ .

We use Eq. (1) to obtain upper bounds on  $f(i)$ , which we then use to find numerical values for  $B_{n,i}$ .

For  $i > 2$ , substituting (5), (6), (7), and (2) into (3), we get that

$$\begin{aligned} D_{n,i,1} &\leq (d_i - d_{i+1}) \frac{f(i)}{n-1} \\ &\quad + (d_i - d_{i+1}) \left( \frac{1 + 3(n-i) + (i-2) + 2(n-i)(i-2)}{n-1} \right) \\ D_{n,i,1} &\leq (d_i - d_{i+1}) \left( \frac{f(i)}{n-1} + 2 \frac{(n-i)(i-1)}{n-1} + 1 \right), \end{aligned}$$

and similarly for  $D_{n,i,2}$ . So we set

$$B_{n,i} = \frac{f(i)}{n-1} + 2 \frac{(n-i)(i-1)}{n-1} + 1.$$

For  $i = 2$ , the same substitution gives  $B_{n,2} = 3$ . For  $i = 1$ , the same substitution gives  $B_{n,1} = 1.5$ .

If there do not exist four paths  $Q_{11}, Q_{12}, Q_{21}, Q_{22}$ , then the average length of a path is too great, namely

$$\begin{aligned} &\frac{1}{2}(\overline{L(a_{n+1}, \dots, a_1)} + \overline{L(a_{n+2}, a_n, a_{n-1}, \dots, a_1)}) > r^* S_n \\ &\frac{d_{n+1} + d_{n+2}}{2} B_n + \left( d_n - \frac{d_{n+1} + d_{n+2}}{2} \right) B_{n,n} + \sum_{i=1}^{n-1} (d_i - d_{i+1}) B_{n,i} > \sum_{i=1}^n r^* d_n. \end{aligned}$$

### 3.2 $n = 3$

If  $d_4 \leq 0.541d_3$ , then, by (1),

$$L\left(a_4, a_3, \frac{d_3}{d_2}a_2, \frac{d_3}{d_1}a_1\right) \leq 4.677d_3 = 3r^*d_3.$$

Then, since  $B_{3,2} = 3 < 2r^*$  and  $B_{3,1} = 1.5 < r^*$  as in the last section,

$$\overline{L(a_4, a_3, a_2, a_1)} \leq r^*(d_1 + d_2 + d_3).$$

Similarly,

$$\overline{L(a_5, a_3, a_2, a_1)} \leq r^*(d_1 + d_2 + d_3),$$

so the induction works. Thus for  $n = 3$  we have the constraint  $d_4 > 0.541d_3$ , which is stronger than the one obtained for  $n = 3$  in the previous section.

## 4 Choosing $n$

We obtained linear constraints for various values of  $n \leq 10$ . These, together with the constraints  $d_i \geq d_{i+1}$ , make a linear program (given in Appendix B), which is unsatisfiable. Thus one of the constraints must not hold, so the induction works for some  $n$ .

## 5 Algorithm

We repeatedly use the inductive step to obtain a sequence of indices  $0 = n_0 < n_1 < n_2 < \dots$ . At a stage  $j$ , we remove  $n_j - n_{j-1}$  points. The intermediate ending points are then of the form  $n_j + k_j$  where each  $k_j$  is 1 or 2. Since we are only using  $n \leq 10$ , we can find all the paths  $Q_{11}, Q_{12}, Q_{21}, Q_{22}$  by brute force in linear time. Now, for both possible values of  $k_1$ , we find which value of  $k_0$  gives the shorter path. Then, for both possible values of  $k_2$ , we find which value of  $k_1$  will make the total path after  $a_{n_2+k_2}$  shorter. We repeat until we get to some  $n_j > m$  at which point we have two paths and choose the shorter one. This whole algorithm is linear.

## 6 Lower Bound on Degree-3 Tree Ratios

Denote by  $\sigma$  the sum of edge weights of the minimal spanning tree and by  $\sigma_3$  the sum of edge weights of a minimal degree-3 tree. Denote by  $(x_1, x_2, \dots, x_n)$  the coordinates of a point in  $n$  dimensions.

In six dimensions, let  $O$  be the origin and let  $v_1, v_2, \dots, v_7$  be the vertices of a simplex with center at  $O$  and radius  $\sqrt{6}$ . Let the coordinates of  $v_i$  be  $(v_{i,1}, v_{i,2}, v_{i,3}, v_{i,4}, v_{i,5}, v_{i,6})$ . Note that  $d(v_i, v_j) = \sqrt{2 * 7/6} \sqrt{6} = \sqrt{14}$ .

Now, given natural  $N$  and  $0 < \alpha < 1$ , take the following tree in  $7N$  dimensions:

1. The origin,  $O$ , is the root.
2. Its  $N$  children are  $p_1, p_2, \dots, p_N$ . The  $p_i$  have coordinates 0 except  $x_{7i} = 1 - \alpha$ .
3. Each  $p_i$  has seven children,  $q_{i,1}, q_{i,2}, \dots, q_{i,7}$ . The coordinates of  $q_{i,j}$  are all 0 except  $x_{7i} = 1$  and, for  $k$  from 1 to 6,  $x_{7i-k} = v_{j,k}$ .

Then  $q_{i,1}, q_{i,2}, \dots, q_{i,7}$  form a simplex with the center distance  $\alpha$  from  $p_i$  and with each vertex distance  $\sqrt{6}$  from the center.

It is easy to check that

$$\begin{aligned}
 d(p_i, p_h) &= \sqrt{2}(1 - \alpha) \quad \text{for } i \neq h, \\
 d(q_{i,j}, q_{i,k}) &= \sqrt{14} = d(q_{i,j}, q_{h,k}) \quad \text{for } j \neq k, h \neq i, \\
 d(p_i, q_{i,j}) &= \sqrt{6 + \alpha^2}, \\
 \sigma &= N(1 - \alpha + 7\sqrt{6 + \alpha^2}).
 \end{aligned}$$

Then we can pick

$$\alpha = -1 - \sqrt{7} + \sqrt{4 + 4\sqrt{7}},$$

which gives  $d(q_{i,j}, q_{h,k}) + d(p_i, p_h) = 2d(p_i, q_{i,j})$ .

Then we can define function  $c$  on the vertices so that  $c(O) = 0, c(p_i) = d(p_i, p_h)/2$  and  $c(q_{i,j}) = d(q_{i,j}, q_{h,k})/2$ . In that case, the length of edge  $AB$  is at least  $c(A) + c(B)$ , so  $c$  can be thought of as a half-edge length. Then, since there are  $8N + 1$  vertices, there are  $8N$  edges, so there is a total of  $16N$  edge endpoints. At most 3 of them contribute 0 to  $\sigma_3$ , at most  $3N$  contribute  $(1 - \alpha)/\sqrt{2}$ , and the remainder contribute  $\sqrt{14}/2$ . Thus,

$$\begin{aligned}
 \sigma_3 &\geq 3N \left( \frac{1}{2} \sqrt{2}(1 - \alpha) \right) + (13N - 3) \left( \frac{1}{2} \sqrt{14} \right), \\
 \frac{\sigma_3}{\sigma} &= \frac{3N(\frac{1}{2}\sqrt{2}(1 - \alpha)) + (13N - 3)(\frac{1}{2}\sqrt{14})}{N(1 - \alpha + 7\sqrt{6 + \alpha^2})}, \\
 \lim_{N \rightarrow \infty} \frac{\sigma_3}{\sigma} &= \frac{3(\frac{1}{2}\sqrt{2}(1 - \alpha)) + 13(\frac{1}{2}\sqrt{14})}{1 - \alpha + 7\sqrt{6 + \alpha^2}} \approx 1.4473.
 \end{aligned}$$

Thus  $\tau_3^\infty \geq 1.447$ .

**Acknowledgements** I thank Samir Khuller for suggesting that I work on this problem and Timothy Chan for improved notation and organization.



### Appendix A: Values of $x_i$

**Table 1**  $x_j$  values used to bound  $f(i)$

$i$	$x_1$ and $x_2$	$x_3$ through $x_i$
3	2.127480103088468	2.715029663803688
4	3.2023557495551507	4.175556640172782
5	4.270167577054796	5.608618419590356
6	5.335126162486634	7.033301794415261
7	6.3986555212789265	8.454218195486414
8	7.461367172755974	9.873101560726544
9	8.52356722480373	11.290758818589284
10	9.585425903496056	12.707618366991161

**Table 2**  $x_j$  values used to bound  $g(n)$

$n$	$x_1$ and $x_2$	$x_3$ through $x_n$	$x_{n+1}$
3	2.4556264573869506	3.5140460449331314	1.5613009117434562
4	3.5424450202354296	4.920230571592636	2.294026685501083
5	4.618609731491003	6.336229610465761	3.0154193383617174
6	5.689328832275783	7.753335975414664	3.7315531287091606
7	6.757011330006688	9.170224016158656	4.4448690694127775
8	7.822844123284092	10.58670954888685	5.15650608100577
9	8.88747045789415	12.002823667602273	5.867063400774457
10	9.951267362449125	13.41863151261787	6.576885724382338

**Table 3**  $x_j$  values for the  $n = 3$  case

$x_1$	$x_2$	$x_3$	$x_4$
1.2840665853752833	1.2840665853752833	1.8003074954981302	1.0528095728981612

### Appendix B: Linear Program

This is the infeasible linear program one achieves. The coefficients on the left are strictly greater (by at least 0.0001) than the actual values one would calculate. Since they are all multiplied by positive values, this takes care of round-off error in calculating coefficients.

Variables:  $d_1, \dots, d_{12}$

$$d_i \geq d_{i+1} \quad \text{for } 1 \leq i \leq 11$$

$$d_1 = 1$$

$$d_4 \geq 0.541d_3$$

$$4.6568(d_3 - d_4) + 5.9188(d_4 - d_5/2 - d_6/2) + 3.2034(d_5 + d_6) \\ \geq 1.559(3d_3 + d_4)$$

$$4.7426(d_3 - d_4) + 6.1891(d_4 - d_5) + 7.3417(d_5 - d_6/2 - d_7/2) \\ + 3.9078(d_6 + d_7) \geq 1.559(3d_3 + d_4 + d_5)$$

$$4.7941(d_3 - d_4) + 6.3513(d_4 - d_5) + 7.6734(d_5 - d_6) \\ + 8.7608(d_6 - d_7/2 - d_8/2) + 4.6125(d_7 + d_8) \geq 1.559(3d_3 + d_4 + d_5 + d_6)$$

$$4.8285(d_3 - d_4) + 6.4595(d_4 - d_5) + 7.8945(d_5 - d_6) + 9.1341(d_6 - d_7) \\ + 10.1782(d_7 - d_8/2 - d_9/2) + 5.3177(d_8 + d_9) \\ \geq 1.559(3d_3 + d_4 + d_5 + d_6 + d_7)$$

$$4.853(d_3 - d_4) + 6.5367(d_4 - d_5) + 8.0525(d_5 - d_6) + 9.4006(d_6 - d_7) \\ + 10.5814(d_7 - d_8) + 11.5946(d_8 - d_9/2 - d_{10}/2) + 6.0232(d_9 + d_{10}) \\ \geq 1.559(3d_3 + d_4 + d_5 + d_6 + d_7 + d_8)$$

$$4.8714(d_3 - d_4) + 6.5946(d_4 - d_5) + 8.1709(d_5 - d_6) + 9.6006(d_6 - d_7) \\ + 10.8837(d_7 - d_8) + 12.0203(d_8 - d_9) + 13.0105(d_9 - d_{10}/2 - d_{11}/2) \\ + 6.729(d_{10} + d_{11}) \geq 1.559(3d_3 + d_4 + d_5 + d_6 + d_7 + d_8 + d_9)$$

$$4.8857(d_3 - d_4) + 6.6397(d_4 - d_5) + 8.2631(d_5 - d_6) + 9.7561(d_6 - d_7) \\ + 11.1189(d_7 - d_8) + 12.3514(d_8 - d_9) + 13.4538(d_9 - d_{10}) \\ + 14.4259(d_{10} - d_{11}/2 - d_{12}/2) + 7.4351(d_{11} + d_{12}) \\ \geq 1.559(3d_3 + d_4 + d_5 + d_6 + d_7 + d_8 + d_9 + d_{10})$$

## References

1. Papadimitriou, C.H., Vazirani, U.V.: On two geometric problems related to the traveling salesman problem. *J. Algorithms* **5**, 231–246 (1984)
2. Singh, M., Lau, L.: Approximating minimum bounded degree spanning trees to within one of optimal. In: *Proc. of the 39th STOC* (2007)
3. Fekete, S.P., Khuller, S., Klemmstein, M., Raghavachari, B., Young, N.: A network-flow technique for finding low-weight bounded-degree trees. *J. Algorithms* **24**, 310–324 (1997)
4. Khuller, S., Raghavachari, B., Young, N.: Low-degree spanning trees of small weight. In: *Proc. of the 26th Ann. ACM Symp. on Theory of Computing* (1994)

5. Chan, T.: Euclidean bounded-degree spanning tree ratios. In: Proc. of the Nineteenth Annual Symposium on Computational Geometry, pp. 11–19 (2003)
6. Jothi, R., Raghavachari, B.: Degree-bounded minimum spanning trees. In: Proc. of the 16th Canadian Conf. on Computational Geometry, CCCG, pp. 192–195 (2004)
7. Monma, C., Suri, S.: Transitions in geometric minimum spanning trees. *Discrete Comput. Geom.* **8**, 265–293 (1992)
8. Young, N.: A Bound on the Sum of Weighted Pairwise Distances of Points Constrained to Balls. Technical report 1103, Cornell University Operations Research and Industrial Engineering (1994)
9. Berman, J., Hanes, K.: Optimizing the arrangement of points on the unit sphere. *Math. Comput.* **140**, 1006–1008 (1977)