A 2-approximation algorithm for the SROCT problem*

Bang Ye Wu   Kun–Mao Chao

PROBLEM: Optimal Sum-Requirement Communication Spanning Trees (SROCT)
INSTANCE: A $G = (V, E, w)$ with vertex weight $r : V \to Z_0^+$.
GOAL: Find a spanning tree T of minimum s.r.c. cost.

Recall that the s.r.c. routing cost of a tree T is defined by $C_s(T) = \sum_{u,v}(r(u) + r(v))d_T(u, v)$. Similar to the PROCT problem, the SROCT problem includes the MRCT problem as a special case and is therefore NP-hard. The s.r.c. cost of a tree can also be computed by summing the routing costs of edges. The only difference is the definition of routing load.

Definition 1: Let T be any spanning tree of a graph G, and r a vertex weight function. For any edge $e = (u, v) \in E(T)$, we define the s.r.c. routing load on the edge e to be $l_s(T, r, e) = 2(r(T_u)|T_v| + r(T_v)|T_u|)$, where T_u and T_v are the two subgraphs obtained by removing e from T. The s.r.c. routing cost on the edge e is defined to be $l_s(T, r, e)w(e)$.

Lemma 1: Let T be any spanning tree of a graph $G = (V, E, w)$ and r be a vertex weight function. $C_s(T) = \sum_{e \in E(T)} l_s(T, r, e)w(e)$.

In this section, we focus on the approximation algorithm for an SROCT. For the PROCT problem, it has been shown that an optimal solution for a graph has the same value as the one for its metric closure. In other words, using bad edges cannot lead to a better solution. However, the SROCT problem has no such a property. For example, consider the graph G in Figure 1. The edge (a, b) is not in $E(G)$, and T is a spanning tree of the metric closure of G. All three possible spanning trees of G are Y_1, Y_2 and Y_3. It will be shown that the s.r.c cost of T is less than that of Y_i for $i = 1, 2, 3$.

To compare the s.r.c costs, we can only focus on the coefficient of k in the cost. Note that only vertices a and x have nonzero weights. By Lemma 1, the s.r.c. cost of T can be computed as follows:

$$\begin{aligned}
& C_s(T) \\
= \; & l_s(T, r, (a, b))w(a, b) + l_s(T, r, (a, y))w(a, y) + l_s(T, r, (y, x))w(x, y) \\
= \; & 2(k(4 + 1) + 0(4k))2 + 2(k \times 1 + 4 \times 4k)(1) + 2(5k \times 1 + 4 \times 1)(1) \\
= \; & 64k + \ldots
\end{aligned}$$

Similarly we have $C_s(Y_1) = 66k$, $C_s(Y_2) = 66k$, and $C_s(Y_3) = 90k$. The example illustrates that it is impossible to transform any spanning tree of $\bar{G}$ to a spanning tree of G without increasing the s.r.c cost for some graph G, where $\bar{G}$ is the metric closure of G. But it should be noted that the

*An excerpt from the book "Spanning Trees and Optimization Problems," by Bang Ye Wu and Kun-Mao Chao (2004), Chapman & Hall/CRC Press, USA.

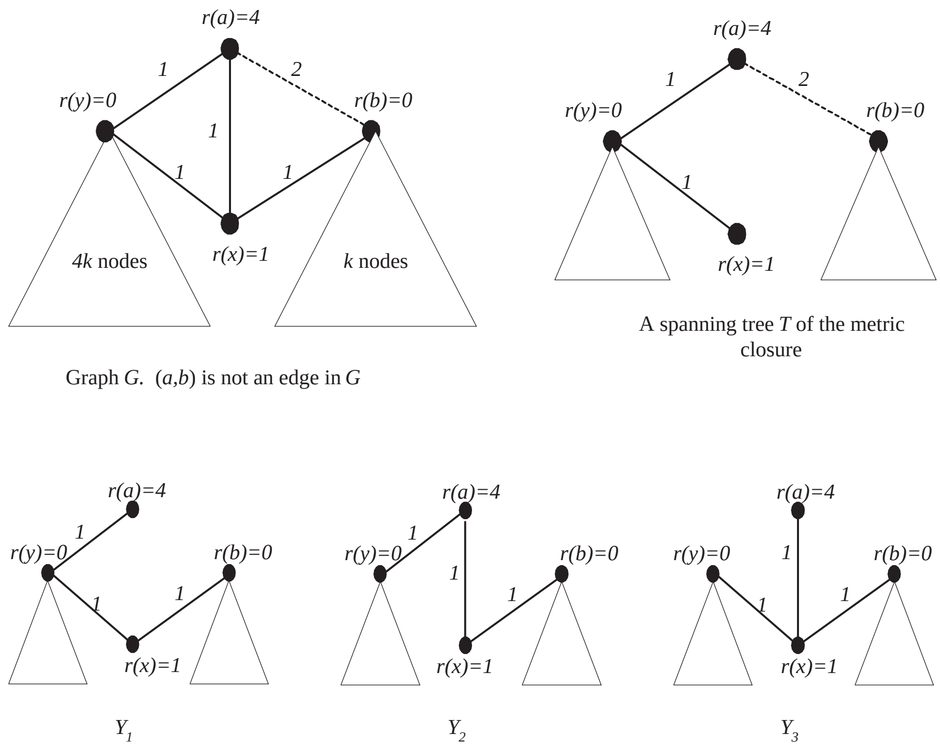

Figure 1: A tree with bad edges may have less s.r.c. cost. The triangles represent nodes of zero weight and connected by zero-length edges.

example does not disprove the possibility of reducing the SROCT problem on general graphs to its metric version.

We shall present a 2-approximation algorithm for the SROCT problem on general graphs. For each vertex v of the input graph, the algorithm finds the shortest-paths tree rooted at v. Then it outputs the shortest-paths tree with minimum s.r.c. cost. We shall show that there always exists a vertex x such that any shortest-paths tree rooted at x is a 2-approximation solution.

In the following, graph $G = (V, E, w)$ and vertex weight r is the input of the SROCT problem. We assume that $|V| = n$, $|E| = m$ and $r(V) = R$.

Lemma 2: Let T be a spanning tree of G. For any vertex $x \in V$,

$$C_s(T) \leq 2 \sum_{v \in V} (nr(v) + R)\, d_T(v, x).$$

Proof:

$$\begin{aligned} C_s(T) &= \sum_{u,v \in V} (r(u) + r(v))\, d_T(u, v) \\ &\leq \sum_{u,v \in V} (r(u) + r(v))\, (d_T(u, x) + d_T(x, v)) \\ &= 2 \sum_{u,v \in V} (r(u) + r(v))\, d_T(u, x) \\ &\leq 2 \sum_{v \in V} (nr(v) + R)\, d_T(v, x). \end{aligned}$$

□

In the following, we use T to denote an optimal spanning tree of the SROCT problem, and use x_1 and x_2 to denote a centroid and an r-centroid of T respectively. Let $P = SP_T(x_1, x_2)$ be the path between the two vertices on the tree. If x_1 and x_2 are the same vertex, P contains only one vertex.

Lemma 3: For any edge $e \in E(P)$, the s.r.c load $l_s(T, r, e) \geq nR$.

Proof: Let T_1 and T_2 be the two subtrees resulting by deleting e from T. Assume that $x_1 \in V(T_1)$ and $x_2 \in V(T_2)$. By the definitions of centroid and r-centroid, $|V(T_1)| \geq n/2$ and $r(T_2) \geq R/2$. Then,

$$\begin{aligned} l_s(T,r,e)/2 &= |V(T_1)|r(T_2) + |V(T_2)|r(T_1) \\ &= |V(T_1)|r(T_2) + (n - |V(T_1)|)\,(R - r(T_2)) \\ &= 2\,(|V(T_1)| - n/2)\,(r(T_2) - R/2) + nR/2 \geq nR/2. \end{aligned}$$

□

The next lemma establishes a lower bound of the minimum s.r.c. cost. Remember that $d_T(v, P)$ denotes the shortest path length from vertex v to path P.

Lemma 4: $C_s(T) \geq \sum_{v \in V} (nr(v) + R)\, d_T(v, P) + nRw(P)$.

Proof: For any vertex u, we define $SB(u)$ to be the set of vertices in the same branch of u. Note that $|SB(u)| \leq n/2$ and $r(SB(u)) \leq R/2$ for any vertex u by the definitions of centroid and r-centroid.

$$\begin{aligned} C_s(T) &= \sum_{u,v \in V} (r(u) + r(v))\, d_T(u,v) \\ &= 2 \sum_{u,v \in V} r(u) d_T(u,v) \\ &\geq 2 \sum_{u \in V} \sum_{v \notin SB(u)} r(u)\,(d_T(u,P) + d_T(v,P)) \\ &\quad + 2 \sum_{u,v \in V} r(u) w(SP_T(u,v) \cap P). \end{aligned} \tag{1}$$

For the first term in (1),

$$\begin{aligned} & 2 \sum_{u \in V} \sum_{v \notin SB(u)} r(u)\,(d_T(u,P) + d_T(v,P)) \\ = \; & 2 \sum_{u \in V} \sum_{v \notin SB(u)} r(u) d_T(u,P) + 2 \sum_{u \in V} \sum_{v \notin SB(u)} r(u) d_T(v,P) \\ \geq \; & \sum_{u \in V} nr(u) d_T(u,P) + 2 \sum_{v \in V} \sum_{u \notin SB(v)} r(u) d_T(v,P) \\ \geq \; & \sum_{u \in V} nr(u) d_T(u,P) + \sum_{v \in V} R d_T(v,P) \\ = \; & \sum_{v \in V} (nr(v) + R)\, d_T(v,P). \end{aligned} \tag{2}$$

For the second term in (1),

$$
\begin{aligned}
& 2\sum_{u,v\in V} r(u)w(SP_T(u,v)\cap P) \\
= \; & 2\sum_{u,v\in V} r(u)\left(\sum_{e\in SP_T(u,v)\cap P} w(e)\right) \\
= \; & \sum_{e\in E(P)}\left(2\sum_{v} r\left(\{u|e\in E(SP_T(u,v))\}\right)\right)w(e) \\
= \; & \sum_{e\in E(P)} l_s(T,r,e)w(e) \\
\geq \; & nRw(P). \qquad \text{(by Lemma 3)}
\end{aligned} \tag{3}
$$

The result follows (1), (2), and (3). □

The main result of this section is stated in the next theorem.

Theorem 5: There exists a 2-approximation algorithm with time complexity $O(n^2 \log n + mn)$ for the SROCT problem.

Proof: Let Y^* and Y^{**} be the shortest-path trees rooted at x_1 and x_2 respectively. Also, for any $v \in V$, let $h_1(v) = w(SP_T(v, x_1) \cap P)$ and $h_2(v) = w(SP_T(v, x_2) \cap P)$. By Lemma 2,

$$
\begin{aligned}
C_s(Y^*)/2 \; & \leq \; \sum_{v\in V}(nr(v)+R)\,d_{Y^*}(v,x_1) \\
& \leq \; \sum_{v\in V}(nr(v)+R)\,(d_T(v,P)+h_1(v)).
\end{aligned} \tag{4}
$$

Similarly

$$
C_s(Y^{**})/2 \leq \sum_{v\in V}(nr(v)+R)\,(d_T(v,P)+h_2(v)). \tag{5}
$$

Since $h_1(v) + h_2(v) = w(P)$ for any vertex v, by (4) and (5), we have

$$
\begin{aligned}
& \min\{C_s(Y^*), C_s(Y^{**})\} \\
\leq \; & \left(C_s(Y^*) + C_s(Y^{**})\right)/2 \\
\leq \; & \sum_{v\in V}(nr(v)+R)\,(2d_T(v,P)+h_1(v)+h_2(v)) \\
= \; & \sum_{v\in V}(nr(v)+R)\,(2d_T(v,P)+w(P)) \\
= \; & 2\sum_{v\in V}(nr(v)+R)\,d_T(v,P)+2nRw(P) \\
\leq \; & 2C_s(T). \qquad \text{(by Lemma 4)}
\end{aligned}
$$

We have proved that there exists a vertex x such that any shortest-paths tree rooted at x is a 2-approximation solution. Since it takes $O(n \log n + m)$ time to construct a shortest-paths tree rooted at a given vertex and the s.r.c cost of a tree can be computed in $O(n)$ time, a 2-approximation solution of the SROCT problem can be found in $O(n^2 \log n + mn)$ time by constructing a shortest-paths tree rooted at each vertex and choosing the one with minimum s.r.c cost. □