The Generalized Steiner Cable-Trench Problem and Efficient Solutions by Modifying Prim's Algorithm

Eric Landquist[a, *], Francis J. Vasko[a], Gregory Kresge[b], Adam Tal[b], Yifeng Jiang[c], Xenophon Papademetris[c]

[a] Department of Mathematics, Kutztown University, Kutztown, PA 19530, USA
(elandqui@kutztown.edu, vasko@kutztown.edu)
[b] Department of Computer Science, Kutztown University, Kutztown, PA 19530, USA
(gkres121@live.kutztown.edu, atal822@live.kutztown.edu)
[c] Department of Diagnostic Radiology, Yale University School of Medicine, 310 Cedar Street, P.O. Box 208042, New Haven, CT 06520-8042, USA
(jiang1feng@gmail.com, xenophon.papademetris@yale.edu)

ABSTRACT

Vasko et al. (2002) introduced the Cable-Trench Problem (CTP) as a combination of the Shortest Path Tree (cables) and Minimum Spanning Tree (trenches) Problems on a weighted graph. This problem was later generalized (Vasko et al., 2013) to accommodate different weights for "laying cables" and "digging trenches." In this paper, we will define the Generalized Steiner CTP (GSCTP) by requiring the solution tree to contain a given subset of the vertex set. We will describe three efficient modifications of Prim's Algorithm that give nearly optimal results as fast as theoretically possible. The motivation for this work is to correct errors in the problem of vascular image connectivity described by Jiang et al. (2011). Here, we will briefly describe the application of the GSCTP to vascular image analysis and give empirical results for graphs with up to 20,000 vertices.

Key words: cable-trench problem, Steiner tree, generalized cable-trench problem, generalized Steiner cable-trench problem, vascular image analysis, mathematical programming formulation.
*Corresponding author

1. INTRODUCTION

Let $G=(V,E)$ be a connected doubly-weighted graph with vertex set $V=\{v_1,\dots,v_n\}$; root vertex v_1; edge set E; and $s_{ij}\geq 0$ and $t_{ij}\geq 0$ the "cable" and "trench" weights of the edge $(v_i,v_j)\in E$, respectively. Let $F\subseteq V$ be the set of *terminal nodes*, $N\subseteq V$ the set of *nonterminal nodes*, and let γ and τ denote the per-unit cable and trench costs, respectively. We define the *Generalized Steiner Cable-Trench Problem* (GSCTP) as the problem of finding a tree $T=(V_T,E_T)$ such that $\{v_1\}\cup F\subseteq V_T\subseteq V$ and $E_T\subseteq E$ which minimizes $\gamma l_\gamma(T)+\tau l_\tau(T)$, where

$$l_\gamma(T)=\sum_{v_k\in F}\sum_{(v_i,v_j)\in\mathcal{P}(v_1,v_k)} s_{ij} \quad\text{and}\quad l_\tau(T)=\sum_{(v_i,v_j)\in E_T} t_{ij}\ ,$$

are the total cable weight of T and total trench weight in T, respectively, and $\mathcal{P}(v_1,v_k)\subseteq E_T$ is the path in T from v_1 to the terminal node v_k. Nodes in the set $V_T\cap N$ are called *Steiner* nodes. If $N=\emptyset$, then the GSCTP reduces to the Generalized Cable Trench Problem (GCTP), introduced by Vasko et al. (2013). If $s_{ij}=t_{ij}$ for all i and j, then we call this problem the

Steiner Cable-Trench Problem (SCTP). Both the GCTP and the SCTP are extensions of the Cable-Trench Problem (CTP) introduced in (Vasko et al., 2002). Further, if $\gamma > 0$ and $\tau = 0$, then a solution to the CTP is any shortest path spanning tree of G with root vertex v_1. In contrast, if $\tau > 0$ and $\gamma = 0$, then a solution to the CTP is any minimum spanning tree of G.

The name "Cable-Trench" comes from the problem of connecting buildings on a university campus to a building housing the central server such that each building is connected directly to the server with its own cable. The main computer building is represented by v_1 and the remaining campus buildings are represented by the other vertices. The edge set represents the only allowable routes for digging trenches and laying cables between two buildings. A trench may carry more than one cable once it is dug. The cost of digging the trench is τ per unit of length and the cost of the cable required is γ per unit of length. In the Steiner version of the problem, not every building is connected to the central server, but cables may be routed through any node. In the GSCTP, the cable costs are independent of the trench costs. This considers the situation in which, even for the same distance, the cost of digging a trench is more costly for some edges versus others because of soil composition or physical obstacles, for example.

We refer the reader to (Vasko, et al., 2013) for a description of further applications and extensions of the CTP. To motivate our definition of the GSCTP, however, we will describe a nontrivial application to vascular image analysis due to Jiang et al. (2011). Briefly, the objective is to digitally represent a network of blood vessels (vasculature) as a tree $T \subseteq G$ in order to assist critical vasculature-related research, including angiogenesis and cancer detection. A massive set of discrete points V, representing the locations of blood vessels, are first detected from 3D medical images, such as CT, and the task is to find the edge set of T which represents the paths of the blood vessels as accurately as possible. In real image analysis scenarios, however, the input invariably contains errors, i.e. false positive vessel points detected from images. One can determine a subset, F, of V to be the leaf nodes of the solution tree, typically those points within the region perfused by the vascular tree. Thus, we let F be the set of leaf nodes and require the solution tree to contain these vertices. Section 3 will further describe the connection of this problem to the GSCTP.

Vasko et al. (2013) explained why the GCTP is NP-complete. Hence, the GSCTP is also NP-complete. Thus, finding optimal solutions to the GSCTP requires exponential time in $n = |V|$ in general. This prohibits one from finding optimal solutions for large examples, such as those encountered in vascular image analysis. Vasko et al. (2013) developed heuristics, requiring $O(n^2)$ arithmetic operations, which find good solutions to the GCTP. In this paper, we adapt these methods to efficiently find nearly optimal solutions to the GSCTP.

In the next section, we will give the mathematical programming formulation for the GSCTP. This will be followed by a brief background on vascular image analysis. Then we will discuss our algorithms and give empirical results for large GSCTPs (with large SCTPs as a special case). The paper will close with some conclusions and areas of future work.

2. MATHEMATICAL PROGRAMMING FORMULATION OF THE GSCTP (MFGSCTP)

The following is a zero-one mixed integer linear programming formulation of the GSCTP.

$$\text{Minimize} \qquad Z = \gamma\left[\sum_{j=1}^{n}\sum_{i=1}^{n} s_{ij}x_{ij}\right] + \tau\left[\sum_{j=1}^{n}\sum_{i=1}^{n} t_{ij}y_{ij}\right] \tag{1}$$

subject to:

$$\sum_{j=2}^{n} x_{1j} = |F| \tag{2}$$

$$\sum_{j=2}^{n} x_{ij} - \sum_{k=1}^{n} x_{ki} = -1 \qquad \text{for all } v_i \in F \text{ and } 2 \le i \le n \tag{3A}$$

$$\sum_{j=2}^{n} x_{ij} - \sum_{k=1}^{n} x_{ki} = 0 \qquad \text{for all } v_i \in N \text{ and } 2 \le i \le n \tag{3B}$$

$$|F| \le \sum_{i=1}^{n}\sum_{j=i+1}^{n} y_{ij} \le n-1 \tag{4}$$

$$|F|y_{ij} - x_{ij} - x_{ji} \ge 0 \qquad \text{for all } i < j \text{ (for all edges)} \tag{5}$$

$$x_{ij} \ge 0 \qquad \text{for all } i, j \tag{6}$$

$$y_{ij} \in \{0,1\} \qquad \text{for all } i < j, \tag{7}$$

where x_{ij} is the number of cables from v_i to v_j (no cables flow back to v_1) and y_{ij} is 1 if a trench is dug between v_i and v_j ($i < j$), and 0 otherwise.

Constraint (2) ensures that exactly one cable leaves the root node for each terminal node. Constraint sets (3A) and (3B) ensure that each node in F is connected by exactly one cable and that no cables terminate at Steiner nodes. Constraint (4) ensures that between $|F|$ and $n-1$ trenches are dug: at least one for each terminal node, but no more than one for each non-root node. Constraint set (5) ensures that cables are not laid unless a trench is dug. Although the x_{ij} (constraint set (6)) are constrained to be nonnegative, they will, in fact, be integers because of their relationship to the trench variables, y_{ij}, which must be either 0 or 1 (constraint set (7)).

In the next section, we will give a brief introduction to the vascular reconstruction problem and discuss how this problem can be formulated as a GSCTP.

3. BACKGROUND ON VASCULAR IMAGE ANALYSIS

As noted in the introduction, discrete locations of blood vessels can be detected from 3D medical images. The vessel radii at these points can also be estimated from the images. These points correspond to the vertices, V, of a complete graph $G = (V, E)$, with edge weights determined by Euclidean length, vessel segment volume, or other physiological factors. The problem is to determine the tree $T \subseteq G$ which represents the actual network of blood vessels. This task currently constitutes one of the bottlenecks in quantitative vascular research (Zagorchev, et al., 2010). Previously, the best methods computed the minimum spanning tree of G, but the results heavily depend on manual correction (Bullitt, et al., 1999; Jomier, et al., 2005; Szymczak et al., 2006).

Recently, some methods (Bruyninckx et al., 2010; Jiang et al., 2010; Jiang et al., 2011) have been proposed to utilize further physiological constraints. In particular, Jiang et al. (2011) applied Murray's Minimum Work Principle (1926), which states that vasculature naturally tends to minimize the total work due to blood flow resistance and metabolic support for the blood volume. These two factors are proportional to the cable cost and trench cost in the GCTP, respectively. Based on Murray's Principle, the vessel connection problem can filnally be formulated as a GCTP with cable weights $s_{ij} = l(e_k)/r(e_k)$ and trench weights $t_{ij} = l(e_k)r(e_k)^2$, where $l(e_k)$ is the length of the blood vessel represented by the edge $e_k = (v_i, v_j)$ and $r(e_k)$ is the radius of the vessel at v_j. In this application, if we choose $\gamma = 1$, then $35000 \leq \tau \leq 175000$ is an appropriate range. We refer the reader to (Jiang et al., 2011) for the technical details of their derivation and to (Vasko et al., 2013) for results on the GCTP treatment of this problem.

It is the case, however, that V will invariably contain errors in real image analysis scenarios. However, the leaf nodes of the solution tree can be determined. We therefore extend the GCTP to the Steiner tree equivalent, the GSCTP, by defining the set of terminal nodes, F, to be the set of these leaf nodes. The set of errors from the vessel point detection is therefore a subset of the non-terminal nodes, N. To solve the resulting large GSCTP, we present a new modification of Prim's Algorithm.

4. A SIMPLE HEURISTIC FOR SOLVING LARGE GSCTPs

We extend the Modified Prim's Algorithm (MOD_PRIM) of Vasko et al. (2013) for the GCTP, to the Generalized Steiner Modified Prim's Algorithm (GSMPRIM) for the GSCTP. Like Prim's and Dijkstra's Algorithms, GSMPRIM adds vertices and edges to the solution tree in a greedy manner, where the "cost" of an edge is the sum of the trench cost of that edge, the cable cost from the root node, v_1, to the terminal node of the edge, and a third "benefit" term based on a node's proximity to terminal nodes. When the solution tree contains each terminal node, GSMPRIM then "trims" all leaf nodes in N, together with the associated terminal branches, since the corresponding edges contribute unnecessary costs to the total trench length. (Such nodes are presumably errors in the vascular image application.)

Two solution strategies are contained in this algorithm. The first strategy (*USE_BENEFIT* = *FALSE* in Algorithm 1) executes MOD_PRIM until all terminal nodes are included, and then trims unnecessary leaf nodes. The second strategy (*USE_BENEFIT* = *TRUE* in Algorithm 1) uses the concept of a benefit function to encourage selection of terminal nodes as well as Steiner nodes adjacent to a relatively large number of terminal nodes. With this strategy, each terminal node not in the solution tree is given 2 benefit points. Each node gets an additional benefit point for each terminal node it is adjacent to or "near." The benefit value of a node is the number of benefit points multiplied by the ratio τ/γ, a multiplier supplied to the algorithm, and the maximum value of a fixed edge metric $|\cdot|$: cable weight, trench weight, etc. The benefit value is then subtracted from the cable cost and trench cost that a vertex and associated edge would contribute to the tree. The algorithm is greedy, selecting nodes of lowest cost.

Algorithm 1: GSMPRIM: Generalized Steiner MOD_PRIM Algorithm

Input: $G = (V, E)$, a doubly-weighted connected graph with vertex set $V = \{v_1\} \dot\cup F \dot\cup N = \{v_1, \dots, v_n\}$, where F is the set of terminal nodes and N is the set of nonterminal nodes; edge set E; cable and trench edge weights s_{ij} and t_{ij}, respectively, for edge (v_i, v_j); edge metric $|\cdot|$; per-unit cable cost γ; per-unit trench cost τ; *USE_BENEFIT* $\in \{TRUE, FALSE\}$; and *MULTIPLIER*.

Output: A tree $T = (V_T, E_T) \subseteq G$, such that $\{v_1\} \cup F \subseteq V_T$ which solves the GSCTP.

GSMPRIM(G):

    $V_T \coloneqq \{v_1\}$
    $E_T \coloneqq \{\}$
    $d_{tree} \coloneqq d_{root} \coloneqq \{\infty, \infty, \dots, \infty\}$
    $cost \coloneqq \{\infty, \infty, \dots, \infty\}$
    $Pre \coloneqq \{1, 1, \dots, 1\}$
    $w_b \coloneqq 0$
    If *USE_BENEFIT*
        $b \coloneqq \{0, 0, \dots, 0\}$ // This is the benefit vector.
        $w_b \coloneqq (\tau/\gamma) \cdot MULTIPLIER \cdot \max_{e \in E} |e|$
    For $1 < i \leq n$ // Initialize the cost and benefit vectors.
        If $e \coloneqq (v_1, v_i) \in E$
            $d_{tree}[i] \coloneqq t_{1i}$
            $d_{root}[i] \coloneqq s_{1i}$
            $cost[i] \coloneqq \tau d_{tree}[i] + \gamma d_{root}[i]$
        If *USE_BENEFIT*
            If $v_i \in F$
                $b[i] \coloneqq 2$
            For all $i < j \leq n$ such that $v_j \in F$ AND $(v_i, v_j) \in E$
                $b[i] \coloneqq b[i] + 1$
    While $\{v_1\} \cup F \not\subseteq V_T$ // This is the main loop.
        $m \coloneqq \text{index}(\ \min\{cost[i] - w_b b[i] : v_i \in V \setminus V_T\}\ )$
        $V_T \coloneqq V_T \cup \{v_m\}$
        $E_T \coloneqq E_T \cup \{(v_{Pre[m]}, v_m)\}$
        If *USE_BENEFIT* AND $v_m \in F$
            For all $v_i \in V \setminus V_T$ such that $(v_m, v_i) \in E$
                $b[i] \coloneqq b[i] - 1$
        For all $v_i \in V \setminus V_T$ such that $e \coloneqq (v_m, v_i) \in E$
            If $cost[i] > \tau t_{mi} + \gamma(d_{root}[m] + s_{mi})$
                $d_{tree}[i] \coloneqq t_{mi}$
                $d_{root}[i] \coloneqq d_{root}[m] + s_{mi}$
                $cost[i] \coloneqq \tau d_{tree}[i] + \gamma d_{root}[i]$
                $Pre[i] \coloneqq m$
    While there exist leaf nodes $w \in N$ // Trim the leaves.
        If $w \in N$ is a leaf node and $e = (v, w) \in E_T$
            $V_T \coloneqq V_T \setminus \{w\}$
            $E_T \coloneqq E_T \setminus \{e\}$
    Return $T = (V_T, E_T)$

Since we add a new vertex and edge with each iteration of the main while loop, GSMPRIM will terminate. Moreover, GSMPRIM requires quadratic time and space.

Theorem 1: Let $n = |V|$, the number of vertices of G. GSMPRIM requires $O(n^2)$ bits of storage and $O(n^2)$ arithmetic operations.

Proof: Notice that the calculation of w_b takes time $O(|E|)$. At worst, G is a complete graph, in which case $|E| = O(n^2)$. Also, updating the arrays b, d_{tree}, and d_{root} only contribute a constant factor to the running time of GSMPRIM, compared with that of Prim's Algorithm. The last loop, trimming the non-terminal leaf nodes, only takes time $O(|N|) = O(n)$. Therefore, GSMPRIM has the same asymptotic running time of Prim's Algorithm, namely $O(n^2)$. Likewise, storage is dominated by $|E| = O(n^2)$. $\square$

Since the storage requirement of a complete graph is $\Theta(n^2)$, the running time of any algorithm solving the GSCTP cannot be $o(n^2)$ in general. Thus, GSMPRIM is asymptotically the fastest possible algorithm for the GSCTP on a general graph.

5. TWO ENHANCEMENTS OF GSMPRIM

Aside from the use of the benefit function, we experimented with two enhancements of GSMPRIM—one semi-greedy and deterministic and the other partially stochastic. These two enhancements were initially developed for the GCTP in (Vasko, et al, 2013). In both cases, we did not use the benefit function in order to better compare the effectiveness of these two modifications to that of the benefit function. The semi-greedy enhancement is a multi-pass modification of GSMPRIM that we call SG-GSMPRIM. Specifically, SG-GSMPRIM constructs candidate solution trees by first sorting the cost of the first candidate edge of the tree in increasing order. The first GSCTP tree generated is the tree found by GSMPRIM (with no benefit function). The second solution tree selects the second lowest cost edge connecting the root node to another vertex as the first edge and then uses GSMPRIM from that point on. The k^{th} lowest cost edge connecting the root node to another node is the first edge selected for the k^{th} solution tree and then continues using GSMPRIM. If the root node has degree m, then SG-GSMPRIM can generate up to m distinct trees. The SG-GSMPRIM solution is the tree of lowest cost from these trees. Note that the running time of SG-GSMPRIM is the same as GSMPRIM, namely $O(n^2)$.

The second heuristic is also a multi-pass modification of GSMPRIM, which we call PS-GSMPRIM. This heuristic selects the first $m < n - 1$ edges of a candidate solution tree stochastically, and then selects the remaining edges in the greedy manner. Let $[p_1, p_2, \dots, p_k]$ be a discrete probability distribution. For each of the first m edges selected for inclusion into a solution, the edge contributing the i^{th} lowest total cost is selected with probability p_i, for $1 \leq i \leq k$. After any number of passes, the lowest cost tree is chosen as the solution tree. For a large GSCTP, we expect the trees that are generated to be distinct. Note that the asymptotic running time of PS-GSMPRIM is also $O(n^2)$.

6. RESULTS

In this section, we first compare GSMPRIM to optimal solutions for small graphs. Then we experiment with very large graphs generated from vascular image data, comparing the results of GSMPRIM with or without the benefit function. For the application to the problem of vascular imaging, we further compare GSMPRIM, SG-GSMPRIM, and PS-GSMPRIM.

In Tables 1 and 2, we compare results from GSMPRIM, first using no benefit and then using benefit multipliers, to optimal solutions found via MFGSCTP on the SCTP (i.e., the cable and trench weights and the edge metric for GSMPRIM are the same) for $\gamma = 1$ and four values of τ. The respective tables summarize the results from a 9-vertex graph (Example 4 from (Vasko et al., 2002)) and a 21-vertex graph based on data taken from a large (27,873-vertex) vascular connectivity data set. For the 21-vertex graph, we used edge weights given by the Euclidean distance between points, i.e., $r(e) = 1$ for all $e \in E$. Table 1 gives the results using the multiplier 1, the best multiplier for that graph, and Table 2 summarizes the best multiplier from the set {0.00001, 0.0001, 0.001, 0.01, 0.1, 0.2, …, 1.5, 2, 2.5, 3}. Percent deviations are from the optimal solution.

Table 1: 9-vertex SCTP

τ	Optimum	GSMPRIM Mult.=0	% Dev. from Opt.	GSMPRIM Mult.=1	% Dev. from Opt.	%Impr. over Mult.=0
0.1	55.54	55.54	0	55.54	0	0
1	84	85.8	2.14	84	0	2.10
5	201.6	249.4	23.7	204.2	1.29	18.1
30	922.6	1134.4	22.5	929.2	0.72	18.1

Table 2: 21-vertex SCTP

τ	Optimum	GSMPRIM Mult.=0	% Dev. from Opt.	Best Mult. Avg.	% Dev. from Opt.	%Impr. over Mult.=0
0.1	20,956,278	20,956,278	1.60	1.5	0.26	1.21
1	33,115,762	33,115,762	8.62	1.0	2.58	5.65
5	79,694,210	79,694,210	13.9	0.6	2.75	10.0
30	362,538,620	362,538,620	13.3	0.5	2.65	9.55

In Table 3, we used the same graphs and terminal nodes as in Table 2, but we used cable and trench weights based on Murray's Principle, thus making these GSCTPs. Here, we included much larger values of τ because of the application to vascular image analysis. Because we are particularly interested in the application to vascular image analysis, we compared results from GSMPRIM, SG-GSMPRIM, and PS-GSMPRIM to the optimum solutions. For GSMPRIM, we used the same benefit multipliers as for Table 2, but we took $|e| = l(e)$. For SG-GSMPRIM, we recorded the percent improvement of the best solution over GSMPRIM based on 20 iterations. Finally, we ran PS-GSMPRIM 30 times with the probability distribution [1/3, 2/9, 2/9, 1/9, 1/9]; the first 15 iterations chose the first 5 non-root nodes stochastically, and the remaining 15 iterations chose the first 10 non-root nodes stochastically.

Table 3: 21-vertex GSCTP

τ	GSMPRIM Mult.=0 % Dev./Opt.	Best Benefit Mult.	Benefit %Impr. / Mult. 0	SG-GSMPRIM %Impr. / Mult. 0	PS-GSMPRIM %Impr. / Mult. 0
0.01	4.19	0.00001	0.0003	0	0
1	4.19	0	0	0.00009	0
5	4.19	0	0	0.006	0
10	4.19	0	0	0.006	0
100	4.06	0	0	0.006	0
10000	5.41	0.001	6.51	8.43	3.58

50000	16.2	0.001	4.39	3.18	6.33
100000	16.2	0.001	1.76	2.54	6.75
150000	15.2	0.001	3.53	2.34	6.80

As expected, GSMPRIM and its variants do not find the optimal solution to the GSCTP in general, but they efficiently find very good solutions. The results in Tables 1 and 2 indicate that GSMPRIM with no benefit gives good solutions to the SCTP for relatively small values of τ and that when the benefit function is used, we obtain nearly optimal solutions for all values of τ. In fact, in Table 2, optimal solutions were found via GSMPRIM four times: once for each value of τ, with multiplier 0.5, 0.9, 0.4, and 0.4, respectively. On the other hand, with the application to vascular image analysis, none of the methods provided significant improvement over GSMPRIM without the benefit function for $\tau \leq 100$. In these cases, however, GSMPRIM with no benefit gave results within 10.5% of the optimal value and for one set of terminal vertices, even gave the optimal tree for all values of τ. For $\tau \geq 10000$, each algorithm provided significant improvements, between 1.76% and 8.43%, over GSMPRIM with no benefit. In these cases, no single algorithm consistently gave the best results, but PS-GSMPRIM was the best of the three.

To test GSMPRIM and its variants on larger graphs, we generated seven large graphs with 501, 1001, 2501, 5001, 10001, 15001, and 20001 vertices. As for the 21-vertex graphs, we used Euclidean distances for edge weights for the SCTPs and the weights s_{ij} and t_{ij} based on Murray's Principle for the GSCTPs. To mimic a real world situation in which terminal nodes are nodes that are certain to be in the solution tree, 90% of the non-root vertices were chosen to be terminal nodes in each case.

We used GSMPRIM on each of the seven SCTP graphs and five values of τ with the same benefit multipliers as above. Table 4a gives the percent improvement of GSMPRIM using the benefit function over GSMPRIM without the benefit function. And Table 4b gives the benefit multiplier that yielded the best result in each case. We note that these graphs are much too large to obtain optimum results.

Table 4a: SCTP Euclidean Distances
Percent improvement of GSMPRIM with benefit over GSMPRIM with no benefit

Vertices	$\tau = 0.01$	$\tau = 1$	$\tau = 5$	$\tau = 10$	$\tau = 100$	Average
501	0.01	3.13	7.33	15.2	0	5.13
1001	0.02	4.89	18.5	27.7	23.8	15.0
2501	0.02	4.18	25.4	23.5	19.2	14.5
5001	0.00008	2.60	0.85	14.1	2.65	4.04
10001	0	2.33	4.47	11.4	10.4	5.72
15001	0	2.49	4.72	8.25	4.40	3.97
20001	0.00004	2.21	5.56	3.52	0	2.26
Average	0.007	3.12	9.55	14.81	8.64	7.23

Table 4b: SCTP Euclidean Distances
Best benefit multiplier for GSMPRIM

Vertices	$\tau = 0.01$	$\tau = 1$	$\tau = 5$	$\tau = 10$	$\tau = 100$
501	0.3	0.03	0.08	0.02	0
1001	0.5	0.03	0.08	0.09	0.01
2501	0.2	0.2	0.06	0.4	0.01

5001	0.09	0.01	0.03	0.01	0.01
10001	0	0.01	0.03	0.5	0.02
15001	0	0.01	0.08	0.3	0.07
20001	0.06	0.07	0.1	0.06	0

In contrast to the 21-vertex examples, the best multipliers were relatively small. Nevertheless, with the exception of $\tau = 0.01$, significant results were obtained in almost every case, with improvements up to 27.7% over GSMPRIM with no benefit.

Next, we experimented with large subsets of the vascular image data set. As with the 21-vertex examples, there were no significant results for $\tau \leq 100$ because of the nature of the application, so we only report results for $\tau \geq 10000$. Tables 5a and 5b give analogous results to those in Tables 4a and 4b, but for the large GSCTP graphs.

Table 5a: GSCTP Vascular Application Percent improvement of GSMPRIM with benefit over GSMPRIM with no benefit					
Vertices	$\tau = 10{,}000$	$\tau = 50{,}000$	$\tau = 100{,}000$	$\tau = 150{,}000$	Average
501	0.15	9.90	17.7	28.8	14.1
1001	5.22	14.6	28.3	23.5	17.9
2501	11.9	26.0	27.2	23.4	22.1
5001	2.13	7.38	15.6	14.7	9.95
10001	1.75	1.05	6.84	7.86	4.37
15001	0	1.56	1.59	0.68	0.96
20001	3.11	1.03	0.58	0	1.18
Average	3.47	8.79	14.0	14.1	10.1

Table 5b: GSCTP Vascular Application Best benefit multiplier for GSMPRIM				
Vertices	$\tau = 10{,}000$	$\tau = 50{,}000$	$\tau = 100{,}000$	$\tau = 150{,}000$
501	0.00001	0.0001	0.0001	0.0001
1001	0.01	0.0001	0.0001	0.0001
2501	0.01	0.01	0.01	0.01
5001	0.0001	0.00001	0.0001	0.0001
10001	0.001	0.001	0.01	0.0001
15001	0	0.02	0.0001	0.0001
20001	0.01	0.0001	0.01	0

The results were most significant for the smaller graphs and, like the SCTP data, for the larger values of τ. With only a few exceptions, significant results were obtained, up to 28.8% improvement over SGMPRIM with no benefit function. In general, the multipliers in these examples were even smaller than those in Table 4b. One interesting observation is that for each $\tau \geq 50{,}000$, the averages are considerably larger than the corresponding averages of the 21-vertex examples in Table 3.

Because of our interest in the application to vascular image analysis, we applied SG-GSMPRIM and PS-GSMPRIM to the GSCTP to compare with the results in Table 5a in order to determine the best heuristic of the three. For Table 6, we recorded the percent improvement of the best solution with 30 iterations of SG-GSMPRIM over GSMPRIM with no benefit function. For the results in Table 7, we give the results of our experiments with PS-GSMPRIM using the probability distribution [1/3, 2/9, 2/9, 1/9, 1/9]; 15 iterations were on the first 6% of the non-root nodes and another 15 iterations were on the first 10% of the non-root nodes.

Table 6: GSCTP Vascular Application Percent improvement of SG-GSMPRIM over GSMPRIM with no benefit					
Vertices	$\tau = 10{,}000$	$\tau = 50{,}000$	$\tau = 100{,}000$	$\tau = 150{,}000$	Average
501	2.08	6.12	2.58	1.79	3.14
1001	1.42	2.52	11.5	3.42	4.72
2501	4.52	2.60	3.00	2.79	3.23
5001	0.72	3.31	3.53	1.10	2.17
10001	0.73	1.45	1.57	0.88	1.16
15001	0.70	0.98	0.85	0.79	0.83
20001	1.97	0.73	0.64	0.60	0.99
Average	1.73	2.53	3.39	1.62	2.32

Table 7: GSCTP Vascular Application Percent improvement of PS-GSMPRIM over GSMPRIM with no benefit					
Vertices	$\tau = 10{,}000$	$\tau = 50{,}000$	$\tau = 100{,}000$	$\tau = 150{,}000$	Average
501	1.16	6.07	2.85	11.6	5.42
1001	2.00	3.66	9.55	7.44	5.66
2501	2.91	2.57	3.95	1.55	2.75
5001	1.00	3.86	1.33	1.42	1.90
10001	0.49	1.12	4.85	1.46	1.98
15001	0.36	1.36	1.33	1.77	1.20
20001	1.46	1.22	1.10	1.69	1.37
Average	1.34	2.84	3.56	3.85	2.90

In contrast to the 21-vertex examples, the use of the benefit function produced the best results in most cases: 21 of the 28 total cases, while SG-GSMPRIM and PS-GSMPRIM were preferred in 3 and 4 of the 28 graphs, respectively. GSMPRIM with the benefit function was the best algorithm of the three for each of the four values of τ, when averaged over the seven vertex sets. In fact, using the Wilcoxon Signed-Rank Test, we determined with 95% confidence that the percentage improvement for GSMPRIM with the benefit function over GSMPRIM without the benefit function was significantly better than the percentage improvement for SG-GSMPRIM and also for PS-GSMPRIM.

Now while SGMPRIM with the benefit yielded the best results for the smallest five graphs, when averaged over the four values of τ, PS-GSMPRIM was the best method on average for the largest two graphs. In particular, it was the best method for the 20001-vertex graph with $\tau \geq 50{,}000$, the scenarios that most closely resemble the actual application to vascular image analysis. In each of these cases, the best tree was found when we selected the first 2000 vertices stochastically.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we defined the GSCTP as the Steiner tree equivalent of the GCTP. Furthermore, we developed three highly efficient, specifically quadratic-time, approximation algorithms, based on Prim's Algorithm, that are capable of finding near-optimal solutions to very large GSCTPs. While the graphs we used for this paper had up to 20,000 vertices, the proposed algorithms can handle graphs with more than 100,000 vertices even on a typical computer by taking advantage of the fact that the adjacency matrix of the graph is sparse. In particular, GSMPRIM with the benefit function produced statistically significant results for GSCTPs over the other two methods. However, PS-GSMPRIM was the best method for those graphs most similar to graphs that would be seen in the application to vascular image analysis. Our motivation for defining the GSCTP and developing efficient algorithms for solving large GSCTPs is to algorithmically automate error correction in the application to vascular image analysis, where the ultimate goal is to facilitate biomechanical and physiological analysis of the vascular tree.

Since the CTP and its variants are a relatively new and unexplored area of study, there are several avenues to consider for future work, both theoretically and experimentally. In particular, we will consider the effectiveness of genetic algorithms, and experiment with modifications to the benefit function and PS-GSMPRIM. We will also consider the other formulation of the Steiner tree problem in which T contains V, but with the possibility of additional (Steiner) nodes being added.

ACKNOWLEDGMENTS

The authors thank Dr. Albert Sinusas and Zhenwu Zhuang, Yale University School of Medicine, for providing vascular image data for our experiments.

REFERENCES

Bruyninckx P., Loeckx D., Vandermeulen D., Suetens P., 2010. Segmentation of liver portal veins by global optimization. In: Karssemeijer N., Summers R.M. (Eds.), *Proc. of SPIE7624, Medical Imaging 2010*. SPIE. pp. 76241Z.

Bullitt E., Aylward S., Liu A., Stone J., Mukherji S., Coffey C., Gerig G., Pizer S., (1999). 3D graph description of the intracerebral vasculature from segmented MRA and tests of accuracy by comparison with X-ray angiograms. In: Kuba A., Sámal M., Todd-Pokropek A. (Eds.), *LNCS Vol. 1613: IPMI*. Springer, Berlin. pp. 308–321.

Jiang Y., Zhuang Z., Sinusas A., Papademetris X., 2010. Vascular tree reconstruction by minimizing a physiological functional cost. In: *CVPRW-MMBIA*. IEEE. pp. 178–185.

Jiang Y., Zhuang Z.W., Sinusas A.J., Staib L.H., Papademetris X., 2011. Vessel connectivity using Murray's hypothesis. In: Fichtinger G., Martel A.L., Peters T.M. (Eds.), *LNCS Vol. 6893: MICCAI 2011*. Springer, Berlin. pp. 528–536.

Jomier J., Ledigarcher V., Aylward S., 2005. Automatic vascular tree formation using the Mahalanobis distance. In: Duncan J., Gerig G. (Eds.), *LNCS Vol. 3750: MICCAI 2005*. Springer, Berlin. pp. 806–812.

Murray C., 1926. The physiological principle of minimum work. I. The vascular system and the cost of blood volume. *Proceedings of the Naional Academy of Sciences* 12(3), 207–214.

Szymczak A., Stillman A., Tannenbaum A., Mischaikow K., 2006. Coronary vessel trees from 3D imagery: a topological approach. *Medical Image Analysis* 10(4), 548–559.

Vasko F., Barbieri R., Rieksts B., Reitmeyer K., Stott K., 2002. The cable trench problem: combining the shortest path and minimum spanning tree problems. *Computers and*

Operations Research 29(5), 441–458.

Vasko F., Landquist E., Kresge G., Tal A., Jiang Y., Papademetris X., 2013. A simple and efficient strategy for solving large generalized cable-trench problems. *Computers and Operations Research*, submitted.

Zagorchev L., Oses P., Zhuang Z.W., Moodie K., Mulligan-Kehoe M., Simons M., Couffinhalt T., 2010. Micro computed tomography for vascular exploration. *Journal of Angiogenesis Research* 2(1), 7.