The Generalized Steiner Cable-Trench Problem and Some Applications

Eric Landquist

Department of Mathematics, Kutztown University of Pennsylvania

elandqui@kutztown.edu

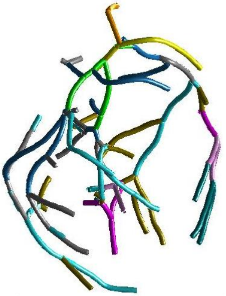
September 1, 2016

Joint work with: Fran Vasko, Greg Kresge, Adam Tal (Kutztown), Yifeng Jiang and Xenophon Papademetris (Yale University)

Outline

- Motivation: Analyzing CT Scans
- Background: The Cable-Trench Problem (CTP)
- The Generalized Steiner Cable-Trench Problem (GSCTP)
- Application to Vascular Image Analysis and Error Correction
- Modified Prim's Algorithms
- Results
- Conclusions
- More Applications
- Future Work and Open Questions

Motivation: Why do we care?



- Given: Data from a Micro-CT scan (combined microscope and CT scan).
- Digitally reconstruct a blood vessel network (vasculature) from a set of points and vessel radii and
- Automatically filter out errors,
- Accurately and efficiently.
- Study vasculature noninvasively for cancer detection, identifying blood clots, etc.

The Problem: Vascular Image Analysis from a CT Scan

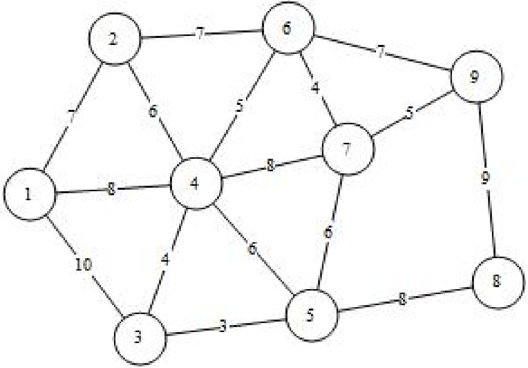
- Example: This scan of a mouse's leg has 27,873 vertices (3D).
- The red dot represents the main artery coming into the leg.
- Can we accurately and efficiently connect the dots to visualize the blood vessel network?

Background: The Cable-Trench Problem

- Suppose you needed to connect each building on a campus to a hub (IT building) with its own (underground) internet cable.
- How should you dig the trenches between buildings to minimize the total cost to dig the trenches and lay the cable?
- If the cables are free, then the solution is a Minimum Spanning Tree (MST).
- If the trenches are free, then the solution is a Shortest Path Tree (SPT).
- In the real world, both cables and trenches have a cost.
- The **Cable-Trench Problem** (CTP) establishes a continuum between the SPT and MST Problems on a weighted graph.

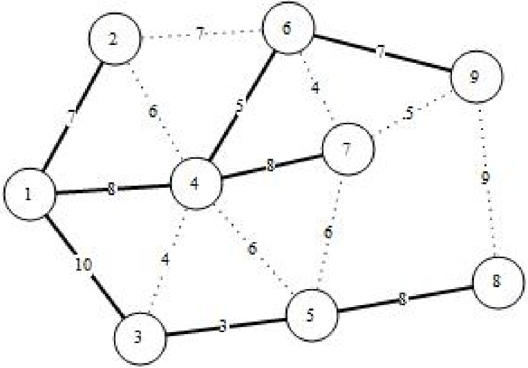
An Example: SPT vs. MST

Circles (nodes, vertices) represent the buildings; Node 1 is the hub.
Edges represent allowable routes for digging trenches.
Numbers (weights) on the edges represent their lengths.



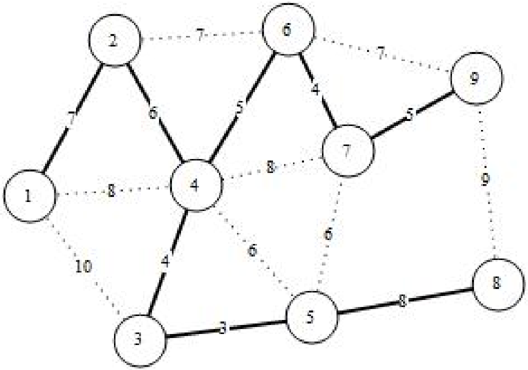
Shortest Path Tree (SPT)

If trenches are free, this minimizes the total cable weight (cost). This is found using **Dijkstra's Algorithm**.



Minimum Spanning Tree (MST)

If cables are free, this minimizes the total trench weight (cost).
This is found using **Prim's Algorithm**.



Background: The Cable-Trench Problem

- Let $G = (V, E)$ be a weighted graph, $V = \{v_1, \ldots, v_n\}$, with root node v_1, and t_{ij} the weight of $(v_i, v_j) \in E$.
- $w_c(T)$ is the total weight of the paths, $\mathcal{P}(v_1, v_j)$, from v_1, to each $v_j \in V$ of a spanning tree T of G: **total cable weight**.
- $w_t(T)$ is the total weight of T: **total trench weight**.
- Let $\gamma \geq 0$ be the **per-unit cable cost** and
- $\tau \geq 0$ the **per-unit trench cost**.

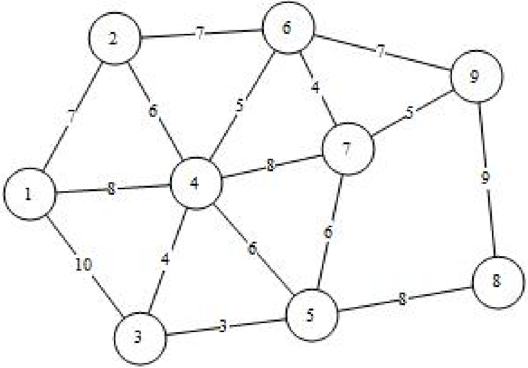
Definition

The **CTP** is the problem of finding a spanning tree $T = (V, E_T)$ of G that minimizes the total cable and trench weight (cost):

$$\gamma w_c(T) + \tau w_t(T) = \gamma \sum_{v_k \in V} \sum_{(v_i, v_j) \in \mathcal{P}(v_1, v_k)} t_{ij} + \tau \sum_{(v_i, v_j) \in E_T} t_{ij} \quad .$$

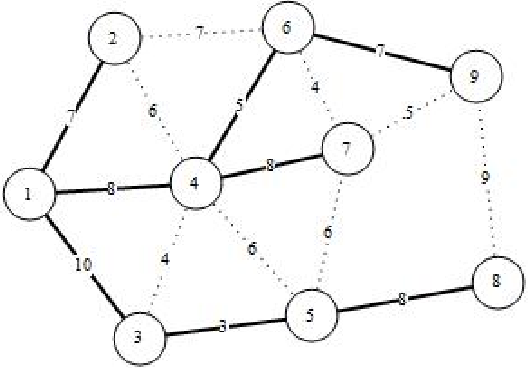
A CTP Example: Revisited

What are the optimal CTP solutions?



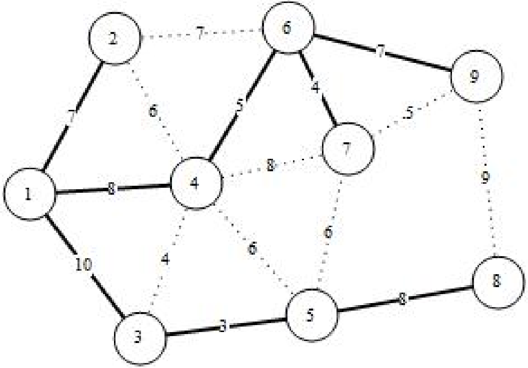
T_1: for $0 \leq \tau/\gamma \leq 1/4$; Cost: $56\tau + 108\gamma$ (SPT)

$\gamma, \tau \geq 0$ are the per-unit cable and trench costs, respectively.



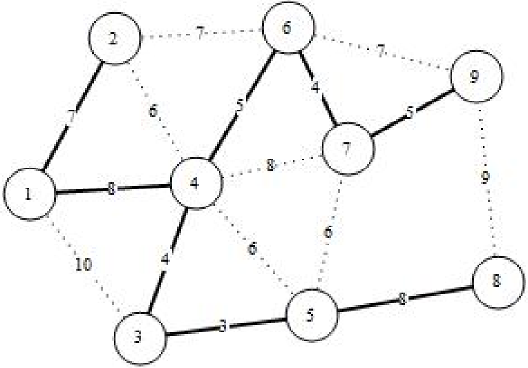
T_2: for $1/4 \leq \tau/\gamma \leq 1$; Cost: $52\tau + 109\gamma$

$\gamma, \tau \geq 0$ are the per-unit cable and trench costs, respectively.



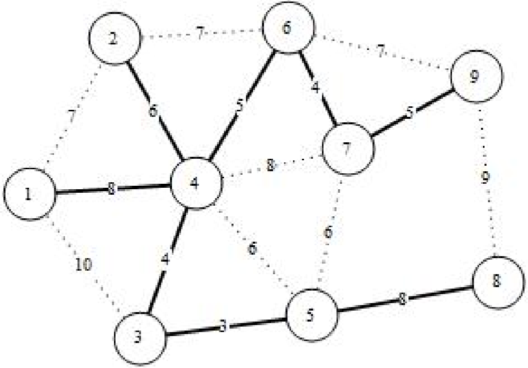
T_3: for $1 \leq \tau/\gamma \leq 7$; Cost: $44\tau + 117\gamma$

$\gamma, \tau \geq 0$ are the per-unit cable and trench costs, respectively.



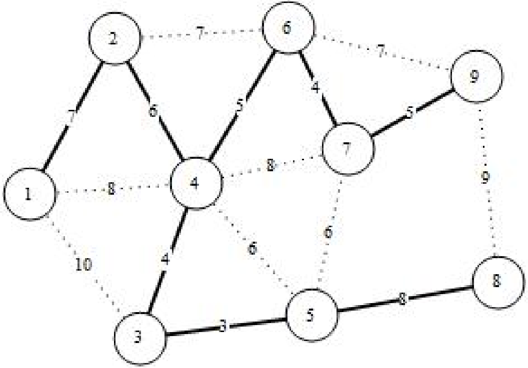
T_4: for $7 \leq \tau/\gamma \leq 28$; Cost: $43\tau + 124\gamma$

$\gamma, \tau \geq 0$ are the per-unit cable and trench costs, respectively.



T_5: for $28 \leq \tau/\gamma$; Cost: $42\tau + 152\gamma$ (MST)

$\gamma, \tau \geq 0$ are the per-unit cable and trench costs, respectively.



A Generalized Cable-Trench Problem

- What if the edge weights don't measure both the trench and cable weights, due to rocks, obstacles, overhead costs, etc.?
- For $(v_i, v_j) \in E$, let s_{ij} be the **cable weight** of (v_i, v_j) and
- t_{ij} the **trench weight** of (v_i, v_j).
- $w_c(T)$ and $w_t(T)$ are the total cable weight and total trench weight of the tree T.
- $\gamma, \tau \geq 0$ are the per-unit cable and trench costs, respectively.

Definition

The **Generalized Cable-Trench Problem** (GCTP) is the problem of finding a spanning tree T of G that minimizes

$$\gamma w_c(T) + \tau w_t(T) = \gamma \sum_{v_k \in V} \sum_{(v_i,v_j)\in\mathcal{P}(v_1,v_k)} s_{ij} + \tau \sum_{(v_i,v_j)\in E_T} t_{ij} \ .$$

A Generalized Steiner Cable-Trench Problem

- What if we only require the solution tree T to contain a given subset, $F \subseteq V$?
- F is called the set of **terminal nodes**,
- and $N = V \setminus (F \cup \{v_1\})$ the set of **nonterminal nodes**.
- $w_c(T)$ and $w_t(T)$ are the total cable weight and total trench weight of the tree T.
- $\gamma, \tau \geq 0$ are the per-unit cable and trench costs.

Definition

The **Generalized Steiner Cable-Trench Problem** (GSCTP) is the problem of finding a subtree $T = (V_T, E_T)$ of G such that $\{v_1\} \cup F \subseteq V_T$ that minimizes

$$\gamma w_c(T) + \tau w_t(T) = \gamma \sum_{v_k \in F} \sum_{(v_i,v_j) \in \mathcal{P}(v_1,v_k)} s_{ij} + \tau \sum_{(v_i,v_j) \in E_T} t_{ij} \ .$$

Vascular Image Analysis and Murray's Principle (1926)

- Good solution (previous state-of-the-art): Find the MST of the graph from a Micro-CT scan via Prim's Algorithm, using vessel segment volume as the edge weight.
- Better idea: Incorporate Murray's Minimum Work Principle.
- Work in a circulatory system is primarily due to two factors:
  1. Overcome blood flow resistance (friction; inversely proportional to vessel radius), and
  2. Metabolic support for the blood volume (proportional to vessel segment volume).
- The body minimizes the total work due to these two factors.
- Applied by Jiang, et al. to improve on the MST model. (2011)

The GCTP Model for Vascular Image Analysis

- "Cables:" Work to overcome friction
- "Trenches:" Work due to metabolic support
- Notation:
  - $r(e_k)$ = Radius of vessel segment $e_k = (v_i, v_j) \in E$
  - $\ell(e_k)$ = Length of vessel segment $e_k \in E$
- Total Cable Length:

$$w_c(T) = \sum_{j=2}^{n} \sum_{e_k \in \mathcal{P}(v_1, v_j)} \frac{\ell(e_k)}{r(e_k)} \text{ , so } s_{ij} = \frac{\ell(e_k)}{r(e_k)}$$

- Total Trench Length:

$$w_t(T) = \sum_{k=1}^{n-1} \ell(e_k) \cdot r^2(e_k) \text{ , so } t_{ij} = \ell(e_k) \cdot r^2(e_k)$$

- Find edges $e_1, \ldots, e_{n-1} \in E$ to minimize the total work:

$$\gamma w_c(T) + \tau w_t(T) = \gamma \sum_{j=2}^{n} \sum_{e_k \in \mathcal{P}(v_1, v_j)} \frac{\ell(e_k)}{r(e_k)} + \tau \sum_{k=1}^{n-1} \ell(e_k) \cdot r^2(e_k)$$

- $\gamma = 1$ and $35000 \leq \tau \leq 175000$ for this application.

Error-correction in Vascular Imaging

- Problem: There are errors in the imaging process.

Error-correction in Vascular Imaging

- Problem: There are errors in the imaging process.
- Current methods require tedious manual correction.

Error-correction in Vascular Imaging

- Problem: There are errors in the imaging process.
- Current methods require tedious manual correction.
- Can we automate error-correction using a CTP-based model?

Error-correction in Vascular Imaging

- Problem: There are errors in the imaging process.
- Current methods require tedious manual correction.
- Can we automate error-correction using a CTP-based model?
- Information: The actual leaf nodes can be determined.

Error-correction in Vascular Imaging

- Problem: There are errors in the imaging process.
- Current methods require tedious manual correction.
- Can we automate error-correction using a CTP-based model?
- Information: The actual leaf nodes can be determined.
- Solution: Let F be the set of leaf nodes of T.

Error-correction in Vascular Imaging

- Problem: There are errors in the imaging process.
- Current methods require tedious manual correction.
- Can we automate error-correction using a CTP-based model?
- Information: The actual leaf nodes can be determined.
- Solution: Let F be the set of leaf nodes of T.
- The other (**Steiner**) nodes are optional.

Error-correction in Vascular Imaging

- Problem: There are errors in the imaging process.
- Current methods require tedious manual correction.
- Can we automate error-correction using a CTP-based model?
- Information: The actual leaf nodes can be determined.
- Solution: Let F be the set of leaf nodes of T.
- The other (**Steiner**) nodes are optional.
- Vascular imaging error-correction is a GSCTP!

Error-correction in Vascular Imaging

- Problem: There are errors in the imaging process.
- Current methods require tedious manual correction.
- Can we automate error-correction using a CTP-based model?
- Information: The actual leaf nodes can be determined.
- Solution: Let F be the set of leaf nodes of T.
- The other (**Steiner**) nodes are optional.
- Vascular imaging error-correction is a GSCTP!
- Next problem: Find good solutions to large GSCTP instances using efficient heuristics.

Error-correction in Vascular Imaging

- Problem: There are errors in the imaging process.
- Current methods require tedious manual correction.
- Can we automate error-correction using a CTP-based model?
- Information: The actual leaf nodes can be determined.
- Solution: Let F be the set of leaf nodes of T.
- The other (**Steiner**) nodes are optional.
- Vascular imaging error-correction is a GSCTP!
- Next problem: Find good solutions to large GSCTP instances using efficient heuristics.
- Assumption: Optimal GSCTP solutions yield images that are very close to the actual vascular network.

Solving the Cable-Trench Problem

- The CTP and its generalizations are NP-hard.
- The obvious exceptions are $\tau = 0$ (SPT) and $\gamma = 0$ (MST).
- Vasko et al. developed a heuristic that finds good solution trees for all τ and γ.
- For large graphs in which τ and γ are known, this is unnecessary and inefficient.
- Determining optimal solutions of large instances of the CTP and GSCTP is computationally infeasible.

Must go faster. Must go faster.

Modified Prim's Algorithm for the GCTP (`MPrim`)

- Combines Dijkstra's (SPT) and Prim's (MST) Algorithms.
- Let $G = (V, E)$ and initialize $T = (\{v_1\}, \{\})$.
- Let v_t represent a vertex in T.
- Iteratively add the vertex $v_k \in V$ and edge $e_k = (v_t, v_k) \in E$ to T so that the added cost, $d(v_k, v_t)$, of
  - "digging the trench" from vertex v_t to v_k plus
  - "laying the cable" from the root node v_1 to v_k is minimized.
- Cost function:

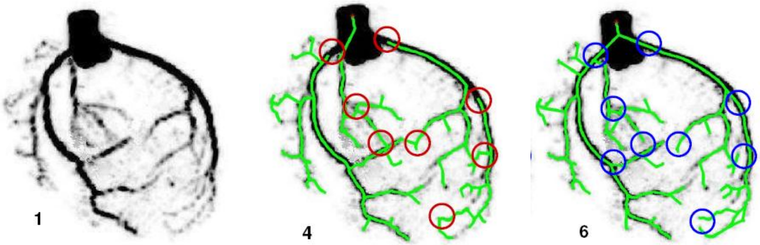
$$d(v_k, v_t) = \text{ cable cost } + \text{ trench cost}$$
$$d(v_k, v_t) = \gamma \sum_{e_i \in \mathcal{P}(v_1, v_k)} w_c(e_i) + \tau w_t(e_k)$$

- **Theorem**: Quadratic running time and space, i.e. $O\left(n^2\right)$.

The Steiner `MPrim` Algorithm for GSCTP: `SMPrim`

- We can further modify `MPrim` for the GSCTP.
- Two key changes:
  1. Stop when the solution tree, T, contains all terminal nodes, F.
  2. "Trim" any leaf nodes of T that are in N, the set of nonterminal nodes.
- **Theorem**: Quadratic running time and space.
- `MPrim` and `SMPrim` give good, but not always optimal, solutions.

MST Model vs. GCTP Model



- The right-most image used `MPrim`.
- Blue circles indicate improvements.
- There is still room for improvement. Can we do better?

Moral of the story thus far

"I like fast algorithms. They're kind of like sports cars for nerds." (Nate Wentzel)

Variations of MPrim and SMPrim

- All variations require quadratic time and space.

1. Modify the cost function to encourage the inclusion of terminal nodes and Steiner nodes adjacent to terminal nodes.

- Why? Certain Steiner nodes are good branching-off points to reach multiple terminal nodes.

2. Semi-Greedy (SG) approach: Insert the j^{th} best edge on the first step, $j > 1$, and proceed greedily.
3. Partially-Stochastic (PS) approach: Insert the first 5-10% of the edges randomly and then proceed in a greedy manner. For random selections, consider the best few candidate edges. Give higher priority to better candidate edges.

Ideas on modifying the cost function

- Define a **benefit** function $b : V \to \mathbb{N}_0$.
- If $v \in F$, $b(v) = 2 + \#$ terminal vertices "near" v.
- If $v \in N$, $b(v) = \#$ terminal vertices "near" v.
- Define a positive multiplier M and let
  $W = (\tau/\gamma)M \cdot \max_{e \in E}\{|e|\}$,
  for some edge metric or weight $|\cdot|$.
- Modify the cost function in `SMPrim` to
  $d(v_k, v_t) =$ cable cost $+$ trench cost $- W \cdot b(v_t)$.
- Run `SMPrim` with many different values of M.

`SMPrim` Results on 21-vertex graph examples

Results are averaged over 5 graphs with randomly chosen sets of 4 to 7 terminal nodes.

`SMPrim` results are percentage deviations from the optimal solutions.

Other results are percentage improvements over `SMPrim`.

τ	SMPrim	M	w/ Benefit	SG	PS
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	6.51	8.43	3.58
100000	16.2	0.001	1.76	2.54	6.75
150000	15.2	0.001	3.53	2.34	6.80

`SMPrim` results on large graphs

Percentage improvement with use of the benefit function over `SMPrim`.
90% of the vertices were chosen at random to be terminals.

n	$\tau = 10,000$	$\tau = 50,000$	$\tau = 100,000$	$\tau = 150,000$
501	0.15	9.90	17.7	28.8
1001	5.22	14.6	28.3	23.5
2501	11.9	26.0	27.2	23.4
5001	2.13	7.38	15.6	14.7
10001	1.75	1.05	6.84	7.86
15001	0	1.56	1.59	0.68
20001	3.11	1.03	0.58	0
25001	4.59	4.35	2.15	0

SG SMPrim results on large graphs

Percentage improvement of Semi-Greedy SMPrim over SMPrim.

n	$\tau = 10,000$	$\tau = 50,000$	$\tau = 100,000$	$\tau = 150,000$
501	2.08	6.12	2.58	1.79
1001	1.42	2.52	11.5	3.42
2501	4.52	2.6	3.00	2.79
5001	0.72	3.31	3.53	1.10
10001	0.73	1.45	1.57	0.88
15001	0.70	0.98	0.85	0.79
20001	1.97	0.73	0.64	0.60
25001	1.31	0.93	0.90	0.85

PS `SMPrim` results on large graphs

Percentage improvement of Partial-Stochastic `SMPrim` over `SMPrim`.

n	$\tau = 10000$	$\tau = 50000$	$\tau = 100000$	$\tau = 150000$
501	1.16	6.07	2.85	11.6
1001	2.00	3.66	9.55	7.44
2501	2.91	2.57	3.95	1.55
5001	1.00	3.86	1.33	1.42
10001	0.49	1.12	4.85	1.46
15001	0.36	1.36	1.33	1.77
20001	1.46	1.22	1.10	1.69
25001	0.63	1.30	1.42	1.77

Summary of Results

- Use of the benefit function had the best results in 24 of the 32 cases.
- The Semi-Greedy approach was best in 3 of the 32 cases.
- The Partial-Stochastic approach was best in 5 of the 32 cases.
- The benefit function results were statistically significant.

Conclusions

- We accurately modeled a vascular imaging problem as a new graph-theoretic problem.
- We were able to accurately eliminate errors without manual correction.
- We found very good suboptimal solutions to GSCTPs using `SMPrim` and various enhancements.
- If more accurate results are desired to instances of the GSCTP, `SMPrim` and its variants yield excellent “seeds” to initialize various metaheuristics.

Other Applications

- Multi-Commodity CTP: Allow cables and pipes for multiple utilities in a common trench network. (Schwarze and Lalla-Ruiz)
- Connect an array of radio telescopes. (Girard, Zyma et al.)
- Construction of logging roads and sawmills. (Marianov et al.)
- Construction of irrigation canals and wells. (Marianov et al.)
- Layout of wireless and wired access networks. (Nielsen et al.)
- Carpooling.
- Other infrastructure development?

Future Work and Open Questions

- Experiment with Teaching-Learning-Based Optimization and other iterative algorithms.
- Use Lagrangean relaxation to determine good lower bounds for solutions to the GSCTP.
- More rigorous statistical analysis.
- Are there other fundamental approaches?
- Theoretical considerations:
  - How close to optimal is `SMPrim` guaranteed to be?
  - When are we guaranteed optimal solutions?

Danke schön!

elandqui@kutztown.edu