Gluttonous: A Greedy Algorithm for Steiner Forest

Vihang Agarwal, Aditya Modi, Shun Zhang

University of Michigan

April 22, 2019
The Steiner Forest Problem

- $I = (M, D)$ is a Steiner forest instance.
  - A metric space $M = (V, d)$: $V$ is a set of vertices; $d(\cdot, \cdot)$ is a metric.
  - Terminals $D \subseteq \binom{V}{2}$: a collection of pairs of vertices
    (for example, $D = \{\{s_1, \bar{s}_1\}, \{s_2, \bar{s}_2\}, \ldots\}$).

- A solution $F$ to the Steiner forest instance $I$ is a forest:
  for any $\{s, \bar{s}\} \in D$, there exists a tree $T \in F$ such that $\{s, \bar{s}\} \subseteq T$.

- An optimal solution minimizes $\text{cost}(F) = \sum_{T \in F} \text{cost}(T)$.
A primal-dual method (Agrawal et al., 1995; Goemans and Williamson, 1995) provides a 2-approximation algorithm.

2 Greedy algorithms.

1 A paired greedy algorithm (Chen et al., 2010) is a $\Omega(\log n)$-approximation.

2 This work (Gupta and Kumar, 2015) provides a constant-factor approximation algorithm by ignoring the pairing relation.
The Gluttonous Algorithm

1: initialize $C^{(0)}$ to be the trivial clustering; the set of added edges $E' \leftarrow \emptyset$; iteration index $i \leftarrow 0$

2: while there exist active supernodes in $C^{(i)}$ do

3: find active supernodes $S_1, S_2 \in C^{(i)}$ with the minimum $C^{(i)}$-punctured distance.

4: $C^{(i+1)} \leftarrow C^{(i)} \setminus \{S_1, S_2\} \cup \{S_1 \cup S_2\}$.

5: find the shortest path from any $u \in S_1$ and $v \in S_2$ under metric $M \setminus C^{(i)}$; add to $E'$ the edges on this path

6: $i \leftarrow i + 1$

7: end while

8: return a maximal acyclic subgraph of $E'$

Vihang Agarwal, Aditya Modi, Shun Zhang

Gluttonous: A Greedy Algorithm for Steiner Forest
The Gluttonous Algorithm

1: initialize $C^{(0)}$ to be the trivial clustering; the set of added edges $E' \leftarrow \emptyset$; iteration index $i \leftarrow 0$
2: **while** there exist active supernodes in $C^{(i)}$ do
3: find active supernodes $S_1, S_2 \in C^{(i)}$ with the minimum $C^{(i)}$-punctured distance.
4: $C^{(i+1)} \leftarrow C^{(i)} \setminus \{S_1, S_2\} \cup \{S_1 \cup S_2\}$.
5: find the shortest path from any $u \in S_1$ and $v \in S_2$ under metric $M \setminus C^{(i)}$; add to $E'$ the edges on this path
6: $i \leftarrow i + 1$
7: **end while**
8: return a maximal acyclic subgraph of $E'$
The Gluttonous Algorithm

1. initialize $C^{(0)}$ to be the trivial clustering; the set of added edges $E' \leftarrow \emptyset$; iteration index $i \leftarrow 0$
2. while there exist active supernodes in $C^{(i)}$ do
3.  find active supernodes $S_1, S_2 \in C^{(i)}$ with the minimum $C^{(i)}$-punctured distance.
4.  $C^{(i+1)} \leftarrow C^{(i)} \setminus \{S_1, S_2\} \cup \{S_1 \cup S_2\}$.
5.  find the shortest path from any $u \in S_1$ and $v \in S_2$ under metric $M \setminus C^{(i)}$; add to $E'$ the edges on this path
6.  $i \leftarrow i + 1$
7. end while
8. return a maximal acyclic subgraph of $E'$
The Gluttonous Algorithm

1: initialize $C^{(0)}$ to be the trivial clustering; the set of added edges $E' \leftarrow \emptyset$; iteration index $i \leftarrow 0$
2: while there exist active supernodes in $C^{(i)}$ do
3: find active supernodes $S_1, S_2 \in C^{(i)}$ with the minimum $C^{(i)}$-punctured distance.
4: $C^{(i+1)} \leftarrow C^{(i)} \setminus \{S_1, S_2\} \cup \{S_1 \cup S_2\}$.
5: find the shortest path from any $u \in S_1$ and $v \in S_2$ under metric $M \setminus C^{(i)}$; add to $E'$ the edges on this path
6: $i \leftarrow i + 1$
7: end while
8: return a maximal acyclic subgraph of $E'$
The Gluttonous Algorithm

1. initialize $C^{(0)}$ to be the trivial clustering; the set of added edges $E' \leftarrow \emptyset$; iteration index $i \leftarrow 0$

2. while there exist active supernodes in $C^{(i)}$ do

3. find active supernodes $S_1, S_2 \in C^{(i)}$ with the minimum $C^{(i)}$-punctured distance.

4. $C^{(i+1)} \leftarrow C^{(i)} \setminus \{S_1, S_2\} \cup \{S_1 \cup S_2\}$.

5. find the shortest path from any $u \in S_1$ and $v \in S_2$ under metric $M \setminus C^{(i)}$; add to $E'$ the edges on this path.

6. $i \leftarrow i + 1$

7. end while

8. return a maximal acyclic subgraph of $E'$

---

Vihang Agarwal, Aditya Modi, Shun Zhang
Gluttonous: A Greedy Algorithm for Steiner Forest
Analysis of Gluttonous

The main result:

**Theorem**

The gluttonous algorithm is a constant-factor approximation for Steiner Forest.

For analysis, we first define *faithfulness*.

**Definition**

A forest $F$ is **faithful** to a clustering $C$ if each supernode $S \in C$ (all vertices in $S$) is (are) contained within a single tree in $F$. 
Analysis of Gluttonous

The analysis proves the bound in the following two steps:

- There exists a solution $F$ that is faithful to $C^g$ and $\text{cost}(F) \leq 2\text{cost}(F^*)$.
- For any solution $F$ that is faithful to $C^g$, $\text{cost}(F^g) \leq O(1) \cdot \text{cost}(F)$.
Bounding the cost of $F$

- Optimal forest solution is $F^*$.  
- Gluttonous returns a forest $F^g$ with a clustering structure.  
- **Idea:** Build the forest $F$ by modifying $F^*$ with two properties.  
  1. $F$ is faithful to the clustering of Gluttonous $F^g$.  
  2. $\text{cost}(F) \leq 2\text{cost}(F^*)$. 

![Diagram of forest and clustering structure](image-url)
Bounding the cost of \( F \)

**Fact:** \( F^* \) is optimal but most likely not faithful.

**Question:** How to ensure faithfulness?

**Idea:** Inductively build \( F \) using \( F^* \) which is faithful to *Gluttonous*.

Assume current \( F \) is faithful to current clustering.

If we connect two unmatched nodes \( u \) and \( v \) in \( S \) and \( S' \):

1. Both are in same tree in \( F \). Unchanged \( F \) is still faithful!
2. Both lie in different \( T_1 \) and \( T_2 \). Bound cost of used path \( P \) as:

\[
\text{cost}(P) \leq \min(d_{T_1}(u, \bar{u}), d_{T_2}(v, \bar{v})) \leq \min(\text{width}(T_1), \text{width}(T_2))
\]

**Next:** \( \text{cost}(F) = \text{sum of all path costs } \text{cost}(P) \).

**Theorem (Low-cost and Faithful \( F \))**

*Sum of all path costs and, therefore, \( F \) is* \( \text{cost}(F^*) + \sum_{T' \in F^*} \text{width}(T') \leq 2 \text{cost}(F^*) \).
Charging the cost of Gluttonous

Result: There exists forest $F$ faithful to $C^g$ with $\text{cost}(F) \leq 2\text{cost}(F^*)$.

Question: How to relate the cost of $F$ to $\text{cost}(F^g)$?

Approach: For faithful $\bar{F}$, charge total $\text{cost}(F^g)$ as $O(1) \cdot \text{cost}(\bar{F})$.

Consider any feasible solution $\bar{F}$. Now,

- For any iteration $t$, build a Steiner forest instance $I_t$ on supernodes.
- We also maintain a forest $F_t$ such that:
  1. $F_t$ is feasible to the instance $I_t$.
  2. $F_t$ maintains the connectivity of $\bar{F}$:
     $u, v$ in same tree in $\bar{F} \Rightarrow S_u, S_v$ in same tree in $F_t$

Approach:

- Start with individual nodes as clustering.
- Set the initial forest solution to $\bar{F}$.
- At any step, when merging two supernodes: merge nodes in $F_t$ ($\text{delete}$ edges to remove cycle).
- To maintain cost, short cut low degree Steiner vertices (inactive supernodes).
Theorem

Given any forest $\bar{F}$, we can charge the cost of gluttonous to at most the cost of $48 \cdot \text{cost}(\bar{F})$.

- Cost of *bought* paths can be charged to the deleted edges.

Theorem (Approximation factor of Gluttonous)

Combining the two results, we can show that the approximation factor of Gluttonous is 96 (constant approximation).

- Choose the faithful clustering from first part as $\bar{F}$.
- $\text{cost}(\bar{F}) \leq 2 \cdot \text{cost}(F^*)$.
- $\text{cost}(F^g) \leq 48 \cdot \text{cost}(\bar{F})$.
- **Note** that the intermediate forest $\bar{F}$ is only considered in analysis.
Cost Sharing Mechanisms

**Intuition:** Informally, a cost-sharing mechanism builds a network connecting agents to their desired source, and allocates the cost incurred among the agents, s.t.

- No group of agents is charged too much precluding the possibility of their being unhappy and trying to leave the system.
Definition: Cost sharing method $\chi$ for the Steiner Forest game. ($I$ is the Steiner Forest instance).

- Seen as a charging function which divides the cost of connecting terminal pairs in $I$ amongst the agents that wish to establish this connection.

Constraints,

- Sum of such cost shares should at least be equal to the total cost of the connections. i.e., $\sum_{(u, \bar{u}) \in D} \chi(I, (u, \bar{u})) \leq \alpha \cdot cost(F^*)$. ($\alpha$-approximate budget-balanced).
- Cost share of any fixed individual agent should never decrease as other agents leave this system. (cross-monotonic)
Strict Cost Shares

- $\chi$ is defined using the timed version of gluttonous algorithm. This is $\gamma$-approximation algorithm.
  - The timed version divides execution into stages $i$ where all supernodes with merging distance in $[2^i, 2^{i+1}]$ are merged instead of the nearest active supernode pair.

- **Getting cost shares:**
  - At each stage increment the cost-share of $(u, \bar{u})$ and $(u', \bar{u}')$ by $\frac{2^{i+1}}{2\gamma}$ ($u$ and $u'$ are active terminals with maximum distance to their mates for a pair of supernodes respectively).
This cost sharing method and its strictness property ensures for an algorithm $A$, if we partition $D$ arbitrarily into $D_1 \cup D_2$, and build a solution $A(D_1)$, then the cost-shares of the terminals in $D_2$ would suffice to augment the solution $A(D_1)$ to one for $D_2$ as well.

Showing strict cost shares for Steiner forest had remained an open problem, and this paper provides a constant factor approximation strict cost share scheme using the structure and analysis for the gluttonous algorithm.
Future Direction

- **Better approximation factor:**
  - 96 and 69 constant factor obtained for greedy and local search algorithms respectively. Best approximation factor available is $2 - \frac{1}{k}$.
  - Thus, the problem of refining the bound and simplifying the analysis is wide open.

- **Dynamic Steiner forest**
  - Here, terminal pairs arrive online and we want to maintain a constant-approximate Steiner Forest while changing the solution by only a few edges in each update.
  - Dynamic Steiner tree algorithms have been based on local search. Thus obtaining such a solution for this generalized version through local search approximation algorithms is another possibility.

- **Stochastic multi-stage Steiner forest**
  - In the stochastic version we are given a distribution over demands and demand set is revealed in the future.
  - The idea is to use cost-sharing schemes to minimize the total expected cost. It would be interesting to show better approximation for these problems. (primal-dual methods give an approximation factor of 5).

