New Complexity Results and Algorithms for Minimum Tollbooth Problem

Soumya Basu
Thanasis Lianeas    Evdokia Nikolova

Department of ECE
The University of Texas at Austin

WINE 2015, Amsterdam
“The combined annual cost of gridlock to these (U.S., U.K., France and Germany) countries is expected to soar to $293.1 billion by 2030...” –INRIX and the CEBR

How to tackle congestion?
Congestion Gridlock

“The combined annual cost of gridlock to these (U.S., U.K., France and Germany) countries is expected to soar to $293.1 billion by 2030…” –INRIX and the CEBR

How to tackle congestion?

- Infrastructure growth is good.
Introduction

Congestion Gridlock

“The combined annual cost of gridlock to these (U.S., U.K., France and Germany) countries is expected to soar to $293.1 billion by 2030...” –INRIX and the CEBR

How to tackle congestion?

- Infrastructure growth is good.
- **But Not Always**: Selfish users lead us to Braess’ Paradox
- **Solution**: Influence the users by placing appropriate Tolls
**Introduction**

**Congestion Gridlock**

“The combined annual cost of gridlock to these (U.S., U.K., France and Germany) countries is expected to soar to $293.1 billion by 2030…” –INRIX and the CEBR

**How to tackle congestion?**

- Infrastructure growth is good.
- **But Not Always:** Selfish users lead us to Braess’ Paradox
- **Solution:** Influence the users by placing appropriate Tolls
- **No Free Lunch:** Each toll comes with large overhead
1 Minimum Tollbooth Problem (MINTB)

2 Complexity Results
   - Single Commodity Network
   - Single Commodity Network with all Edges Used

3 MINTB in Series Parallel Graph
   - Background
   - Algorithm

4 Summary

5 Future Directions
System Model

- Directed graph $G(V, E)$ with single commodity $(s, t)$
- Flow dependent latency, $\mathcal{L} = \{\ell_e(\cdot) : \forall e \in E\}$
- Traffic network, $\mathcal{G} = \{G, (s, t), \mathcal{L}\}$

Social Optimum (SO)

SO flow $o$ is a flow that minimizes social cost.

Nash Equilibrium (NE)

A flow is said to be in NE iff property (1) and (2) holds:

1. All used paths cost $\mathbf{L}$
2. All unused paths cost $\geq \mathbf{L}$
Problem Definition

- **Tolled edge cost**: Under toll $\theta$ and flow $f$, tolled edge cost $c_e = \ell_e(f_e) + \theta_e$
- **Induce flow $f$**: Under the tolled edge cost $f$ is in NE
- **Induced length**: Under NE the tolled cost of used paths

Minimum Tollbooth Problem (MINTB)

- Given traffic network $\mathcal{G}$ and an optimal flow $o$
- Find toll with **minimum support** which induces $o$
Formulated as Mixed integer linear program (MILP). *Hearn et al., 1998*

Heuristics developed: Genetic Algorithm, LP relaxation. *Hardwood et al., ’08; Bai et al., ’09*

In Multi-commodity networks it is **NP-hard**. *Bai et al.,’08*

Design toll for inducing **general flow**. *Harks et al.,’08*

**Figure: Complexity Diagram**
Contributions

- **First APX-hardness:**
  Single commodity networks with linear latencies.

- **MINTB with used edges only:**
  Is MINTB efficiently solvable? **No still NP hard.**

- **First Exact Poly-time Algorithm:**
  Algorithm for **Series-Parallel** graphs.
Theorem

For instances with linear latencies and single commodity, it is NP-hard to approximate the solution of MINTB by a factor of less than 1.1377.

Figure: Complexity Diagram
**MINTB Reduction: Vertex Cover**

**Vertex Cover Instance**

\[ e_k = (i, j) \]

**Figure: MINTB Reduction**

Soumya Basu, UT Austin

New Complexity Results and Algorithms for Minimum Tollbooth Problem
MINTB Reduction: Vertex Cover

Vertex gadget for $v_i$: $G_i$

Vertex gadget for $v_j$: $G_j$

Figure: MINTB Reduction
MINTB Reduction: Vertex Cover

**Vertex Cover Instance**

\[ e_k = (i, j) \]

**Figure: MINTB Reduction**
**MINTB Reduction: Vertex Cover**

Vertex gadget for $v_i$: $G_i$

$e_{4,i} (3)$

$e_{1,i} (1)$

$e_{2,i} (0)$

$e_{3,i} (1)$

Vertex gadget for $v_j$: $G_j$

$e_{4,j} (3)$

$e_{1,j} (1)$

$e_{2,j} (0)$

$e_{3,j} (1)$

Edge Gadget for $e_k$:

$g_{1,k} (0.5)$

$g_{2,k} (0.5)$

$e_k = (i, j)$

Vertex Cover Instance

**Figure: MINTB Reduction**
MINTB Reduction: Vertex Cover

Vertex gadget for \( v_i : G_i \):

Edge Gadget for \( e_k \):

**Figure:** MINTB Reduction
MINTB Reduction: Vertex Cover

- Given Vertex Cover instance $G_{vc}$ with $n_{vc}$ vertices
- Create a Single Commodity network $G$

**Lemma 1**

- A Vertex Cover of size $x$ in $G_{vc} \iff$ Opt-inducing toll with support $n_{vc} + x$ in $G$.

**Theorem**

It is NP hard to approximate Minimum Vertex Cover to within a factor of $1.3606$.

*Dinur et al 2005*
Are Unused Edges the Troublemaker?

- **Motivation:** Less overhead present in edge removal.
- **Model:** Unused edges in social optimum flow are removed.

**Theorem**

For instances with *linear latencies*, it is **NP-hard** to solve MINTB even if *all edges are used* by the optimal solution.

Reduction follows from *PARTITION* problem.
Complexity Diagram

**MINTB on General Graphs**

APX Hard

General Graphs with Used edges

NP Hard

Figure: Complexity Diagram
Series Parallel Graphs (SP)

An SP graph is created by starting from a directed edge and inductively connecting two graphs in series or in parallel.
Series Parallel Graphs (SP)

An SP graph is created by starting from a directed edge and inductively connecting two graphs in series or in parallel.

Figure: SP Graph and Parse Tree
An SP graph is created by starting from a directed edge and inductively connecting two graphs in series or in parallel.
Algorithm 1

**Input:** A parallel link network

**Output:** Induce $L$ with min support

1. Sort edges
2. Append length $\ell_{\text{end}} = \infty$
Algorithm 1

Input: A parallel link network
Output: Induce $L$ with min support

1. Sort edges
2. Append length $\ell_{\text{end}} = \infty$
3. Max used edge cost $\ell_{\text{max}}$

Figure: Example: $\ell_{\text{max}} = \ell_1$, $i_0 = 1$
Algorithm 1

**Input:** A parallel link network

**Output:** Induce $L$ with min support

1. Sort edges
2. Append length $l_{end} = \infty$
3. Max used edge cost $l_{max}$
4. Create list $\{(\eta, l)\}$
   - Toll on edges $1, \ldots, \eta$
   - Maximally induce $l$

Figure: Example: $l_{max} = l_1$, $i_0 = 1$
**Algorithm 1**

**Input:** A parallel link network  
**Output:** Induce \( L \) with min support 

1. Sort edges  
2. Append length \( \ell_{end} = \infty \)  
3. Max used edge cost \( \ell_{max} \)  
4. Create list \( \{(\eta, \ell)\} \)  
   - Toll on edges \( 1, \ldots, \eta \)  
   - Maximally induce \( \ell \)  
5. Place toll to induce \( L \)

**Figure:** Example: Placing tolls
Induce length $L$

- Series comb. $G_1(S)G_2$: Induce $L_1(\leq L)$ in $G_1$ and $L - L_1$ in $G_2$.
- Parallel comb. $G_1(P)G_2$: Induce $L$ in $G_1$ and $G_2$.

Monotonicity Lemma

In a $SP$ graph $G$ with maximum used path length $\ell_{max}$,

- We can induce length $L \iff L \geq \ell_{max}$
- $L$ is **induced optimally** with support $T$
  \[ \Rightarrow \ell \leq L \text{ can be induced optimally with support } t \leq T \]
Series Parallel in $P$: Extension to $SP$

**MakeList: Bottom-up List creation: Dynamic Programming**

1) **Start** from PL  2) **Combine** list in series/parallel  3) **Recur** till root

---

**Figure: Step1: Example Run of MINTB Algorithm**

Soumya Basu, UT Austin  New Complexity Results and Algorithms for Minimum Tollbooth Problem
Series Parallel in $P$: Extension to $SP$

Placetoll: Top-down Induction of length: Traceback

1) **Start** from root.  
2) **Branch** in series/parallel.  
3) **Recur** till PL.

**Figure:** Step2: Example Run of MINTB Algorithm
The MINTB on a SP graph with $|E| = m$, is solved optimally in time $O(m^3)$. 

Theorem
Theorem

The MINTB on a SP graph with $|E| = m$, is solved optimally in time $O(m^3)$.

Presence of Braess Structure:
- The Monotonicity Lemma breaks.
- Edges to induce length $3 = 3$.
- Edges to induce length $4 = 2$.

Figure: Counter
Summary

**MINTB** on General Graphs and Multicommodity Network

**NP Hard**

*Figure: Complexity Diagram Prior to the Work*
Summary

- Reduction from Vertex Cover Problem
- APX hardness of 1.1377
- Exploits Unused Edges in an Optimal Flow

Complexity Results

- APX Hard
- General Graphs with Used edges
- NP Hard
- Series-Parallel Graphs

Algorithm

- Absence of Braess Structure
- Monotonicity Property
- Tree Decomposition

- Dynamic Programming
- Bottom Up List Creation
- Top Down Toll Placement

Soumya Basu, UT Austin
New Complexity Results and Algorithms for Minimum Tollbooth Problem
Future Directions

- Is the APX hardness result **tight**?
  - **YES** Find matching approximation algorithms.
  - **NO** Give tighter APX hardness results.

- Design Practical Algorithms:
  - Improved heuristics with **performance guarantee**.
  - Faster algorithms for **large-scale traffic networks**.

- What happens while **Taxing Sub-networks**?
Future Directions

- Is the APX hardness result **tight**?
  - **YES** Find matching approximation algorithms.
  - **NO** Give tighter APX hardness results.

- Design Practical Algorithms:
  - Improved heuristics with **performance guarantee**.
  - Faster algorithms for **large-scale traffic networks**.

- What happens while **Taxing Sub-networks**?

**Questions?**