Approximate Graph Embeddings in the Cloud

Highlights of Algorithms 2018

Matthias Rost
Technische Universität Berlin, Internet Network Architectures

Stefan Schmid
Universität Wien, Communication Technologies
Cloud Providers Offer Data Center Resources

Customers  Cloud Data Center (Amazon, Google, ...)

[Diagram of cloud infrastructure with nodes representing customers and data centers, illustrating connectivity and resource allocation.]
Cloud Providers Offer Data Center Resources

Customers Cloud Data Center (Amazon, Google, ...)

‘Classic’ Cloud Computing

- Customer specifies number and ‘size’ of Virtual Machines
- Communication between VMs not modeled
Cloud Providers Offer Data Center Resources

'Classic' Cloud Computing
- Customer specifies number and 'size' of Virtual Machines
- Communication between VMs not modeled

Goal: Virtual Networks (since \( \approx 2006 \))
- Additionally: communication requirements given

Matthias Rost (TU Berlin)  Approximate Graph Embeddings in the Cloud  Highlights of Algorithms 2018
‘Classic’ Cloud Computing

▶ Customer specifies number and ‘size’ of Virtual Machines
▶ Communication between VMs not modeled

Goal: Virtual Networks (since \(\approx 2006\))

▶ Additionally: communication requirements given

The Virtual Network Embedding Problem (VNEP)

▶ Map virtual nodes to substrate nodes
▶ Map virtual edges to paths in the substrate
▶ Respecting capacities & mapping restrictions
‘Classic’ Cloud Computing

- Customer specifies number and ‘size’ of Virtual Machines
- Communication between VMs not modeled

Goal: Virtual Networks (since ≈ 2006)

- Additionally: communication requirements given

The Virtual Network Embedding Problem (VNEP)

- Map virtual nodes to substrate nodes
- Map virtual edges to paths in the substrate
- Respecting capacities & mapping restrictions
‘Classic’ Cloud Computing

- Customer specifies number and ‘size’ of Virtual Machines
- Communication between VMs not modeled

Goal: Virtual Networks (since ≈ 2006)

- Additionally: communication requirements given

The Virtual Network Embedding Problem (VNEP)

- Map virtual nodes to substrate nodes
- Map virtual edges to paths in the substrate
- Respecting capacities & mapping restrictions
‘Classic’ Cloud Computing

- Customer specifies number and ‘size’ of Virtual Machines
- Communication between VMs not modeled

Goal: Virtual Networks (since $\approx$ 2006)

- Additionally: communication requirements given

The Virtual Network Embedding Problem (VNEP)

- Map virtual nodes to substrate nodes
- Map virtual edges to paths in the substrate
- Respecting capacities & mapping restrictions
The Virtual Network Embedding Problem (VNEP)

- Map virtual nodes to substrate nodes
- Map virtual edges to paths in the substrate
- Respecting capacities & mapping restrictions

Virtual Network  Substrate (Physical Network)

Related Work

- VNEP (and related problems) studied intensively in the networking community: > 100 papers.
- VNEP is related to classical problems as, e.g., subgraph isomorphism, but different . . .
- No approximations known for arbitrary virtual networks graphs.
The Virtual Network Embedding Problem (VNEP)

- Map virtual nodes to substrate nodes
- Map virtual edges to paths in the substrate
- Respecting capacities & mapping restrictions

Virtual Network Substrate (Physical Network)

A B
1 1

C D
6

AC B

D

2/2 4/5

0/0 1/1

3/3 1/1

Embedding

Related Work

- VNEP (and related problems) studied intensively in the networking community: > 100 papers.
- VNEP is related to classical problems as, e.g., subgraph isomorphism, but different . . .
- No approximations known for arbitrary virtual networks graphs.
The Virtual Network Embedding Problem (VNEP)

- Map virtual nodes to substrate nodes
- Map virtual edges to paths in the substrate
- Respecting capacities & mapping restrictions

Related Work

- VNEP (and related problems) studied intensively in the networking community: > 100 papers.
- VNEP is related to classical problems as, e.g., subgraph isomorphism, but different . . .
- No approximations known for arbitrary virtual networks graphs.
Related Work

- VNEP (and related problems) studied intensively in the networking community: > 100 papers.
- VNEP is related to classical problems as, e.g., subgraph isomorphism, but different . . .
- No approximations known for arbitrary virtual networks graphs.

Focus: Offline Variant

Setting: Multiple Virtual Network requests are given
Objectives: Maximize profit (admission control) or minimize ‘cost’ s.t. capacity constraints.
Related Work

- VNEP (and related problems) studied intensively in the networking community: \( \geq 100 \) papers.
- VNEP is related to classical problems as, e.g., subgraph isomorphism, but different . . .
- No approximations known for arbitrary virtual networks graphs.

Focus: Offline Variant

- Setting: Multiple Virtual Network requests are given
- Objectives: Maximize profit (admission control) or minimize ‘cost’ s.t. capacity constraints.

Approach: Randomized Rounding à la Raghavan & Thompson

- Compute opt. ‘convex combinations’ of mappings: \( D_r = \{ ( f_r^k, m_r^k ) \} \) for request \( r \)
  - weight \( \geq 0 \)
  - mapping
- Probabilistically select mapping \( m_r^k \) according to weight \( f_r^k \) for each request \( r \)
  - Yields: approximate solutions of bounded resource augmentations with high probability
Approach: Randomized Rounding à la Raghavan & Thompson

- Compute opt. ‘convex combinations’ of mappings: $\mathcal{D}_r = \{ (f_r^k, m_r^k) \}_{k \text{ for request } r}$
  - Probabilistically select mapping $m_r^k$ according to weight $f_r^k$ for each request $r$
  - Yields: approximate solutions of bounded resource augmentations with high probability

Main Challenge: Computing (Convex Combinations) of Valid Mappings

- Classic LP Formulation yields no meaningful solutions (→ unbounded integrality gap)

Substrate  Request  Classic LP Solution

- Observation: Need to fix confluence targets a priori.
Approach: Randomized Rounding à la Raghavan & Thompson

- Compute opt. ‘convex combinations’ of mappings: \( \mathcal{D}_r = \{ (f_r^k, m_r^k) \}_k \) for request \( r \)
- Probabilistically select mapping \( m_r^k \) according to weight \( f_r^k \) for each request \( r \)
  - Yields: approximate solutions of bounded resource augmentations with high probability

Main Challenge: Computing (Convex Combinations) of Valid Mappings

- Classic LP Formulation yields no meaningful solutions (\( \rightarrow \) unbounded integrality gap)

Obervation: Need to fix confluence targets a priori.
Approach: Randomized Rounding à la Raghavan & Thompson

- Compute opt. ‘convex combinations’ of mappings: \( D_r = \{(f_r^k, m_r^k)\}_{k \geq 0} \) for request \( r \)
- Probabilistically select mapping \( m_r^k \) according to weight \( f_r^k \) for each request \( r \)
  - Yields: *approximate solutions of bounded resource augmentations with high probability*

Main Challenge: Computing (Convex Combinations) of Valid Mappings

- Classic LP Formulation yields no meaningful solutions (\( \rightarrow \) unbounded integrality gap)

Substrate  Request  Classic LP Solution  Extraction Order

- Observation: Need to fix *confluence targets* a priori.
Approach: Randomized Rounding à la Raghavan & Thompson

- Compute opt. ‘convex combinations’ of mappings: $\mathcal{D}_r = \{(f^k_r, m^k_r) \}_{k}^\text{weight} \geq 0$ for request $r$
- Probabilistically select mapping $m^k_r$ according to weight $f^k_r$ for each request $r$
  - Yields: \textit{approximate solutions of bounded resource augmentations with high probability}

Main Challenge: Computing (Convex Combinations) of Valid Mappings

- Classic LP Formulation yields no meaningful solutions ($\rightarrow$ unbounded integrality gap)

Substrate  Request  Classic LP Solution  Extraction Order

- Observation: Need to fix \textit{confluence targets} (here: node $k$) a priori.
Main Challenge: Computing (Convex Combinations) of Valid Mappings
▶ Classic LP Formulation yields no meaningful solutions (→ unbounded integrality gap)
▶ Observation: Need to fix confluence targets a priori.

Main Contributions
▶ LP Formulations for cactus request graphs → first approximation algorithms\(^a\)
▶ Derivation of heuristics & extensive computational evaluation\(^a\)
▶ Extension to arbitrary virtual network topologies → FPT-approximations\(^b\)
▶ FPT required: no poly.-time algorithms for computing valid mappings for general graphs\(^c\)

Main Challenge: Computing (Convex Combinations) of Valid Mappings

- Classic LP Formulation yields no meaningful solutions (→ unbounded integrality gap)
- Observation: Need to fix *confluence targets* a priori.

Main Contributions

- LP Formulations for *cactus request graphs* → first approximation algorithms\(^a\)
- Derivation of *heuristics* & extensive *computational evaluation*\(^a\)
  - Extension to arbitrary virtual network topologies → FPT-approximations\(^b\)
  - FPT required: no poly.-time algorithms for computing valid mappings for general graphs\(^c\)


Main Challenge: Computing (Convex Combinations) of Valid Mappings

- Classic LP Formulation yields no meaningful solutions (→ unbounded integrality gap)
- Observation: Need to fix confluence targets a priori.

Main Contributions

- LP Formulations for cactus request graphs → first approximation algorithms\(^a\)
- Derivation of heuristics & extensive computational evaluation\(^a\)
- Extension to arbitrary virtual network topologies → FPT-approximations\(^b\)
- FPT required: no poly.-time algorithms for computing valid mappings for general graphs\(^c\)


Main Challenge: Computing (Convex Combinations) of Valid Mappings

- Classic LP Formulation yields no meaningful solutions (→ unbounded integrality gap)
- Observation: Need to fix confluence targets a priori.

Main Contributions

- LP Formulations for cactus request graphs → first approximation algorithms
- Derivation of heuristics & extensive computational evaluation
- Extension to arbitrary virtual network topologies → FPT-approximations
- FPT required: no poly.-time algorithms for computing valid mappings for general graphs

---


Main Contributions

▶ LP Formulations for cactus request graphs → first approximation algorithms
▶ Derivation of heuristics & extensive computational evaluation
▶ Extension to arbitrary virtual network topologies → FPT-approximations
▶ FPT required: no poly.-time algorithms for computing valid mappings for general graphs

---


---

Thanks for your attention!