Approximate Graph Embeddings in the Cloud

Technische Universität Berlin

Matthias Rost
Internet Network Architectures
Technical University of Berlin

Stefan Schmid
Communication Technologies
University of Vienna

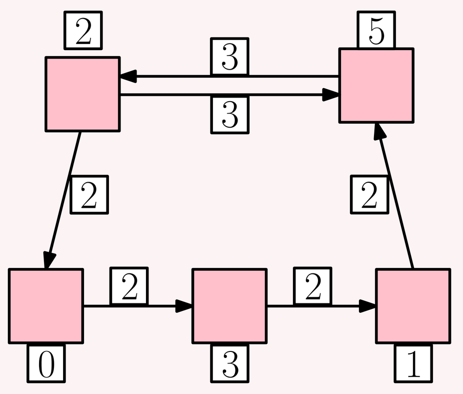
universität wien

The Virtual Network Embedding Problem

Substrate $G_S = (V_S, E_S)$

Represents **physical network**

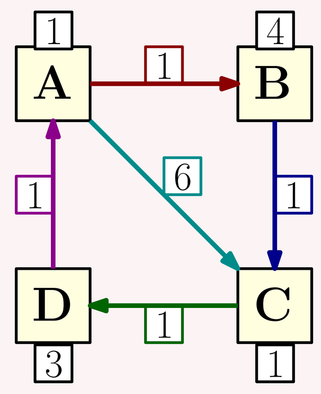
- capacities $d_S : G_S \to \mathbb{R}_{\geq 0}$



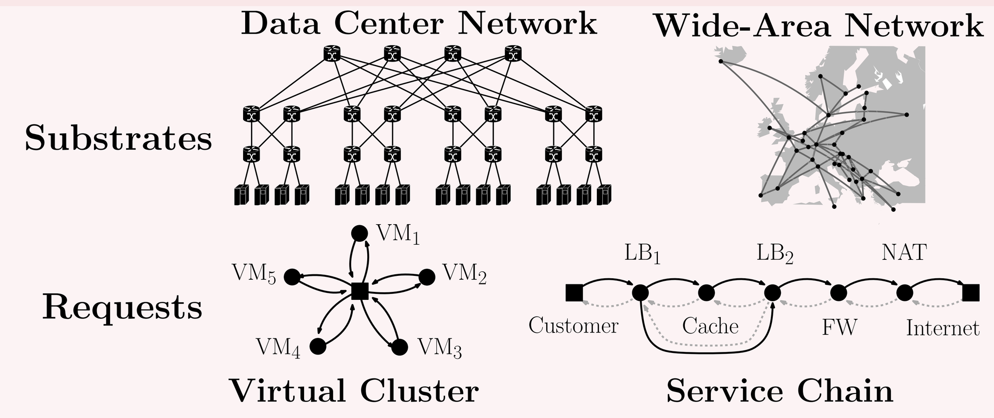
Virtual Network Request $G_r = (V_r, E_r)$

Represents **workload**

- demands $d_r : G_r \to \mathbb{R}_{\geq 0}$
- profit $p_r \in \mathbb{R}_{\geq 0}$
- mapping restrictions
  - $V_S^i \subseteq V_S$ for $i \in V_r$
  - $E_S^{i,j} \subseteq E_S$ for $(i,j) \in E_r$



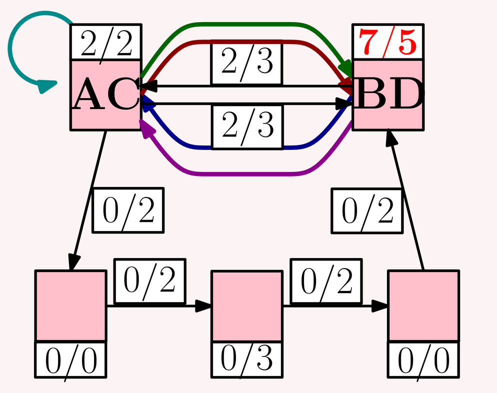
Example Substrates & Requests



Valid Mappings $m_r \in \mathcal{M}_r$ and Feasible Embeddings

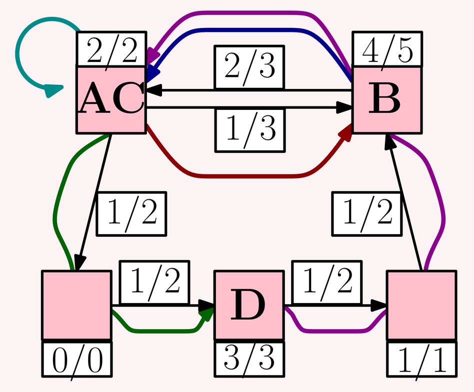
A valid mapping $m_r = (m_V, m_E) \in \mathcal{M}_r$, with $m_V : V_r \to V_S$ and $m_E : E_r \to \mathcal{P}(E_S)$, satisfies . . .

valid connectivity: $m_V(i) \overset{m_E(i,j)}{\rightsquigarrow} m_V(j)$
valid node mapping: $m_V(i) \in V_S^i$
valid edge mapping: $m_E(i,j) \subseteq E_S^{i,j}$



Valid Mapping

A feasible embedding is valid and respects capacity constraints



Feasible Embedding

Problem Statement

Given: substrate G_S and a set of requests $\{G_r\}_{r \in \mathcal{R}}$
Task: embed subset $\mathcal{R}' \subseteq \mathcal{R}$ *feasibly* maximizing $\sum_{r \in \mathcal{R}'} p_r$

Previous Works

- survey from 2013 lists more than 100 papers [1]
- **only heuristics & exact algorithms known**

Randomized Rounding Framework

Enumerative Packing Formulation

$$\max \sum_{r \in \mathcal{R}} \sum_{m_r^k \in \mathcal{M}_r} f_r^k \cdot p_r$$
$$\sum_{m_r^k \in \mathcal{M}_r} f_r^k \leq 1 \qquad \forall r \in \mathcal{R}$$
$$\sum_{r \in \mathcal{R}} \sum_{m_r^k \in \mathcal{M}_r} f_r^k \cdot A(m_r^k, x) \leq d_S(x) \qquad \forall x \in R_S$$
$$f_r^k \in [0,1] \qquad \forall r \in \mathcal{R}, m_r^k \in \mathcal{M}_r$$

result are 'convex' combinations of mappings
$\mathcal{D}_r = \{(f_r^k, m_r^k) | f_r^k \geq 0, m_r^k \in \mathcal{M}_r\}$ for $r \in \mathcal{R}$

Classic Randomized Rounding à la Raghavan & Thomspon [2]

Algorithm: VNEP Approximation
```
// perform preprocessing
```
compute *optimal* LP solution
compute $\{\mathcal{D}_r\}_{r \in \mathcal{R}}$ from LP solution
do
  solution ← ROUNDING($\{\mathcal{D}_r\}_{r \in \mathcal{R}}$)
while (*solution not* (α, β, γ)*-approximate and rounding tries not exceeded*)

Algorithm 2: ROUNDING
Input : Decomp. LP Solution $\{\mathcal{D}_r\}_{r \in \mathcal{R}}$
foreach $r \in \mathcal{R}$ **do**
  choose m_r^k **with probability** f_r^k
end
return *solution*

Derived Heuristics

Vanilla Rounding rounds n many solutions and return the solution maximizing the profit ($\mathrm{RR}_{\mathrm{MaxProfit}}$) or the one minimizing the resource augmentations ($\mathrm{RR}_{\mathrm{MinLoad}}$).

Heuristic Rounding discards rounded mappings whose inclusion would exceed capacities → no resource augmenations.

Our Randomized Rounding Algorithm returns (α, β, γ)-approximate solutions for the VNEP *with high probability*. [3], [4]

$$\underbrace{\alpha = 1/3}_{\text{profit}} \qquad \underbrace{\beta = 1 + \varepsilon \cdot \left(2 \cdot \underset{\leq |\mathcal{R}| \cdot \max_{r \in \mathcal{R}} |V_r|}{\underline{\Delta(V_S)}} \cdot \log(|V_S|)\right)^{1/2}}_{\text{node resource augmentations}} \qquad \underbrace{\gamma = 1 + \varepsilon \cdot \left(2 \cdot \underset{\leq |\mathcal{R}| \cdot \max_{r \in \mathcal{R}} |E_r|}{\underline{\Delta(E_S)}} \cdot \log(|E_S|)\right)^{1/2}}_{\text{edge resource augmentations}}$$

$$\underbrace{\varepsilon = \max_{r \in \mathcal{R}, x \in R_S} d_{max}(r,x)/d_S(x) \leq 1}_{\text{max demand-to-capacity ratio}} \qquad \Delta(X) = \max_{x \in X} \sum_{r \in \mathcal{R}} \left(\underbrace{A_{max}(r,x)}_{\substack{\text{max alloc. on resource } x \\ \text{by any valid mapping of } r}} \Big/ \underbrace{d_{max}(r,x)}_{\substack{\text{max single demand for} \\ \text{resource } x \text{ by request } r}} \right)^2$$

Main Contribution: Decomposable Linear Programming Formulations

Classic Multi-Commodity Flow LP

$\mathbf{y^u_{r,i}} \in [0,1]$: maps node $i \in V_r$ on V_S
$\mathbf{z^{u,v}_{r,i,j}} \in [0,1]$: maps edge $(i,j) \in E_r$ on $(u,v) \in E_S$

$$\sum_{(u,v) \in \delta^+(u)} \mathbf{z^{u,v}_{r,i,j}} - \sum_{(v,u) \in \delta^-(u)} \mathbf{z^{v,u}_{r,i,j}} = \mathbf{y^u_{r,i}} - \mathbf{y^u_{r,j}}$$

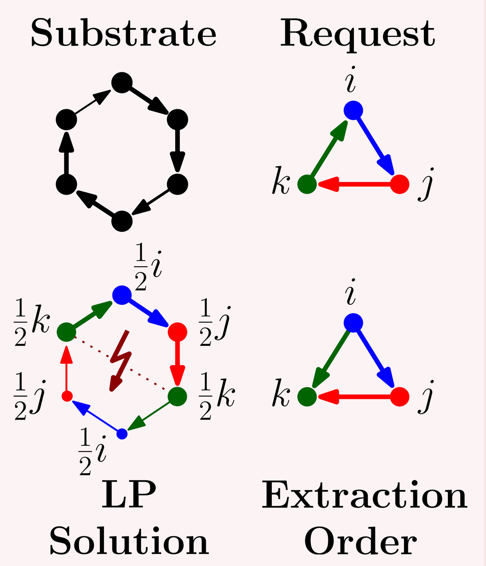
Local Connectivity Property

Given a (fractional) mapping of $i \in V_r$ to $u \in V_S$, a 'valid' mapping can be recovered for edges incident to i and their respective endpoints.

Consider $(i,j) \in E_r$ and $y^u_{r,i} > 0$, then $\exists P : u \rightsquigarrow v$ with $y^v_{r,j} > 0$ and $z^e_{r,i,j} > 0$ for $e \in P$.

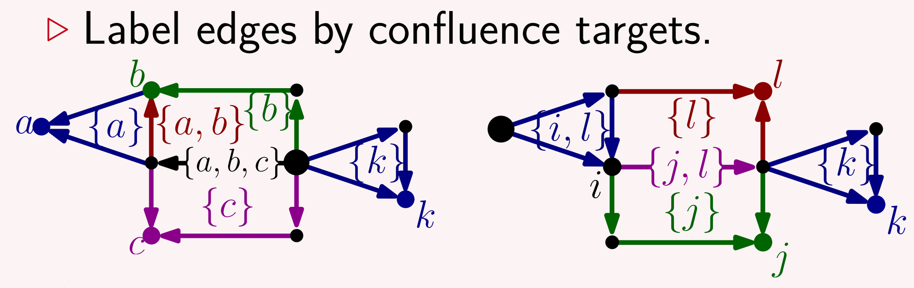
Consider $(k,i) \in E_r$ and $y^u_{r,i} > 0$, then $\exists P : w \rightsquigarrow u$ with $y^w_{r,k} > 0$ and $z^e_{r,k,i} > 0$ for $e \in P$.

Thm: MCF LP solutions cannot be decomposed into valid mappings.

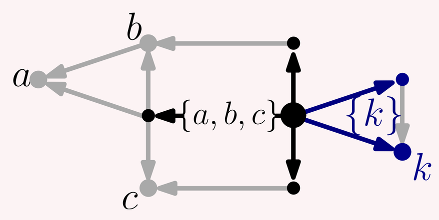


Idea of Novel LP Formulation [4]

- $\{\langle(i,j),(j,k)\rangle, \langle(i,k)\rangle\}$ is **confluence**.
- **Confl. targets need to be decided a priori.**
  - Label edges by confluence targets.



- Outgoing edges are partitioned into bags:



- Edges with overlapping labels are placed into same bag.

- **Extraction Width (of specific order):** maximal edge bag size plus one.

Extraction Width & Formulation Size [4]

- For spec. extraction order $\mathrm{G_r^{\mathcal{X}}}$, the size of our novel LP formulation is $\mathcal{O}(|\mathrm{G_S}|^{\mathrm{ew}_{\mathcal{X}}(\mathrm{G_r^{\mathcal{X}}})} \cdot |\mathrm{G_r}|)$.
- Thm: Finding order of min width is $\mathcal{NP}$-hard.

(FPT-)Approximating the VNEP [3], [4]

- LP Solutions can be decomposed into set $\mathcal{D}_r$ of weighted mappings (cf. Packing LP).
- FPT-approximations for the VNEP; poly.-time approximations for cactus request graphs.

Hardness of Approximating the VNEP [5]

We cannot do better than FPT: computing valid mappings is $\mathcal{NP}$-complete for planar graphs.

Computational Evaluation [3]

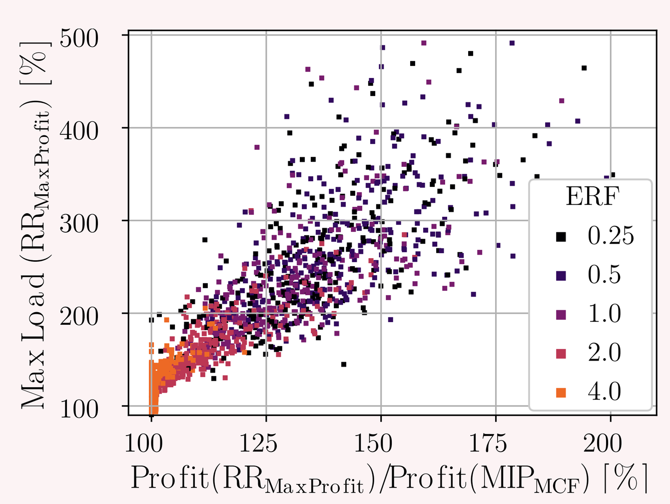
Requests
- Cactus Graphs

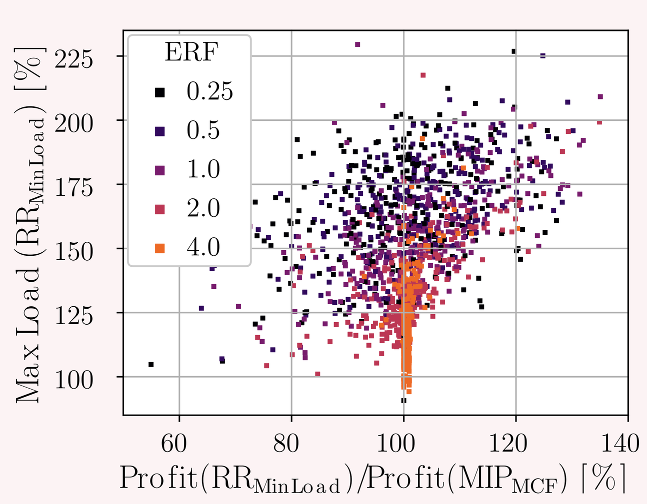
Substrate
- GEANT Network

Parameter Space
- Node Resources
- Edge Resources
- Number Requests

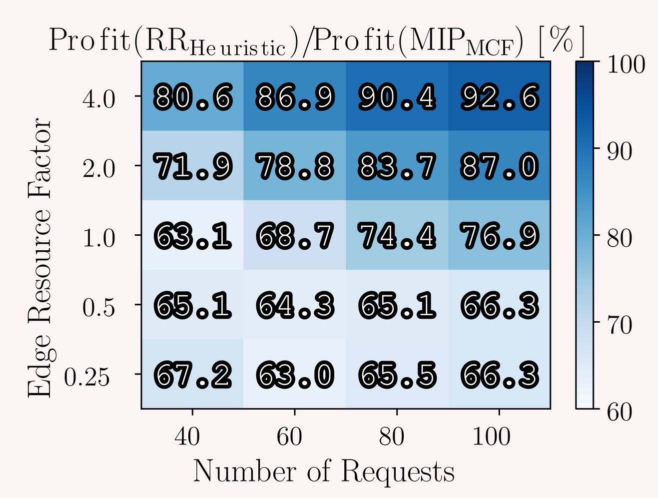
Baseline
- Integer Program

Vanilla Rounding: Max Profit & Min Augmentations





Heuristic Rounding



References

[1] A. Fischer, J. F. Botero, M. T. Beck, H. De Meer, and X. Hesselbach, "Virtual network embedding: a survey", *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, 2013.

[2] P. Raghavan and C. D. Thompson, "Provably Good Routing in Graphs: Regular Arrays", in *Proc. 17th ACM STOC*, 1985.

[3] M. Rost and S. Schmid, "Virtual Network Embedding Approximations: Leveraging Randomized Rounding", in *Proc. IFIP Networking*, 2018.

[4] ——,"(FPT-)Approximation Algorithms for the Virtual Network Embedding Problem", Tech. Rep., Mar. 2018, (under submission). [Online]. Available: http://arxiv.org/abs/1803.04452.

[5] ——,"Charting the Complexity Landscape of Virtual Network Embeddings", in *Proc. IFIP Networking*, 2018.

Contact: Matthias Rost - mrost@inet.tu-berlin.de