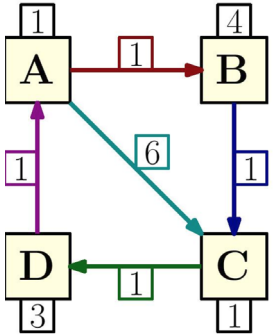
Cost-Efficient Embedding of Virtual Networks With and Without Routing Flexibility

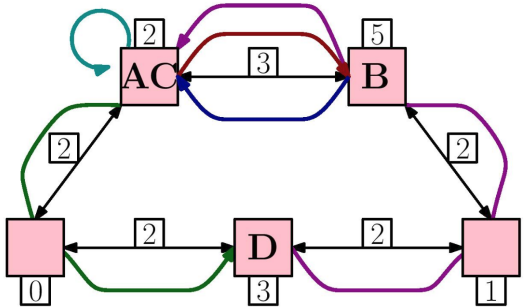
IFIP Networking 2020

Balázs Németh, Budapest Univ. of Technology and Economics, Hungary
Yvonne-Anne Pignolet, DFINITY, Switzerland
Matthias Rost, TU Berlin, Germany
Stefan Schmid, Faculty of Computer Science, University of Vienna, Austria
Balázs Vass, Budapest Univ. of Technology and Economics, Hungary

Introduction: Virtual Network Embeddings



Request and Substrate Graphs



Request: directed weighted graph $G_r=(V_r, E_r, d_r)$

- V_r : request (virtual) nodes
- $E_r \subseteq V_r \times V_r$: request (virtual) edges
  directed data flow between request nodes.
- Element $x \in V_r \cup E_r$ has demand $d_r(x) \geq 0$
  e.g., CPU, memory, bandwidth requirements
- For request graph $r \in R$,
  $n_r=|V_r|$ and $m_r=|E_r|$

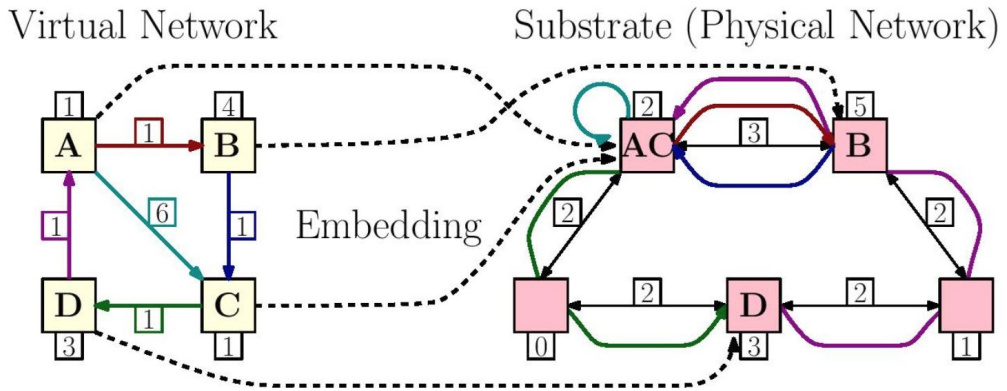
Substrate : weighted graph $G_S = (V_S, E_S, d_S)$

- V_S : physical machines
- E_S : physical communication links
- Element $x \in V_S \cup E_S$ has capacity $d_S(x) \geq 0$
  e.g., CPU, memory, bandwidth, etc.)
- $n_s=|V_s|$ and $m_s=|E_s|$

Embedding of Virtual Networks

Given a set of requests R, we want to:

- Map virtual element to substrate element, s.t. the mapping is *valid* + *feasible*,



- **valid**: respects restrictions (e.g., avoids distant substrate nodes, or substrate links raising security issues)
- **feasible**: valid + respects substrate capacities

Cost of mapping a request r:

- $A(m_r,x)$: resource allocation induced by valid mapping m_r on substrate element $x \in G_S$
- $c_S(x)$: unit cost of substrate resource $x \in G_S$
- Cost of a mapping m_r of r is defined as:

$$c_S(m_r) = \sum_{x \in G_S} c_S(x) \cdot A(m_r, x)$$

Problem Versions

- **Cost Valid Mapping Problem (CVMP):** return valid minimal cost mapping m_r or request r on substrate s, if one exists f
- **Cost Virtual Network Embedding Problem (CVNEP):** return minimal cost feasible embedding M_R for the request set R on substrate s, if one exists.
- **Profit Virtual Network Embedding Problem (PVNEP):** Given a profit $b_r \geq 0$ for each request $r \in R$, return feasible embedding of a subset of requests $R' \subseteq R$ maximizing the profit $\Sigma_{r \in R} b_r$.

- **VNEP without Routing Flexibilities** . Given a routing P, a mapping m_r is valid if each virtual edge takes the path predefined in P. Under this adaptation, CVNEP and PVNEP return a feasible embedding minimizing the cost or maximizing the profit for a request set R.

- **(α, β, γ) -approximation** of the CVNEP (or PVNEP):
  - at most $\alpha > 1$ times the cost (or at least $1/\alpha$ times the profit) of an optimal solution
  - allocations on nodes and edges within factors of $\beta \geq 1$ and $\gamma \geq 1$ of the original capacities respectively, with high probability (whp).

Contributions

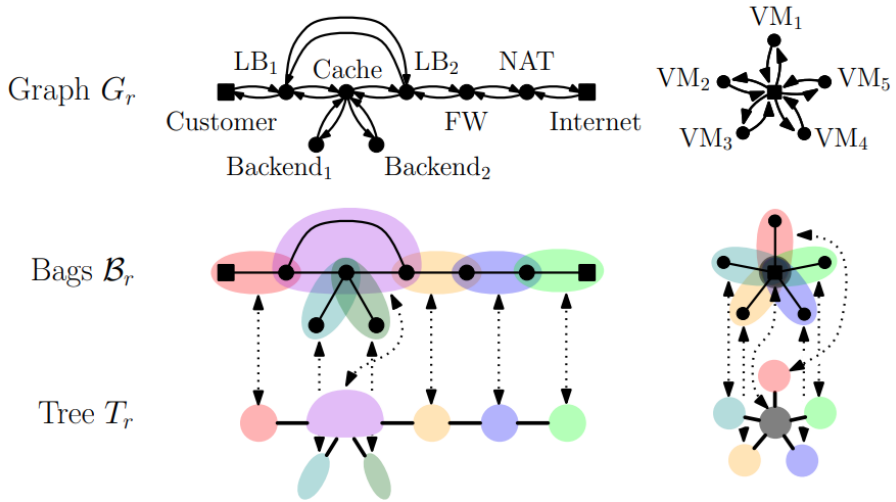
Main Contributions

- polynomial-time approximation for the general Cost VNEP
  - via randomized rounding
  - first polynomial-time approximation algorithm with provable guarantees on **embedding cost**
- defining novel variant of VNEP where routes cannot be optimized,
  (e.g., due to performance considerations or technological constraints)
  - adapting algorithms and approximation guarantees

Preliminaries

Tree Decompositions

- pair $T_r = (T_r, B_r)$,
- undirected tree $T_r=(V_T,E_T)$ and a family of node bags $B_r=\{B_t\}$ $t \in V_T$ with $B_t \subseteq V_r$, s.t.:
  - for node $i \in V_r$, set of tree nodes containing i is connected in T_r,
  - each node and each edge is contained in at least one bag.
- small if no two adjacent bags B_1 and B_2 s.t. $B_1 \subseteq B_2$
- treewidth: max. bag size minus 1
- treewidth *tw(G_r)*: minimum widths among all decompositions
- **Computing a tree decomp. of minimal width is FPT-tractable**
- **...and can be transformed into a small decomp. in linear time**



In practice, request graphs are of low tree -width.

Solving the Cost Valid Mapping Problem

Algorithm 1: DynVMP: Computing Optimal Valid Mappings

Input : substrate G_S, request G_r, tree decomposition $\mathcal{T}_r$
Output: valid mapping of minimal cost or $\perp$ if none exists

- Given a tree decomposition T_r of request graph r, algorithm DynVMP solves the CVMP optimally in $\mathcal{O}(n_r{}^3 \cdot n_S{}^{2 \cdot \mathrm{tw}(\mathcal{T}_r)+3})$

- for details, see [6]:
  M. Rost, E. Döhne, and S. Schmid, "Parametrized complexity of virtual network embeddings: Dynamic & linear programming approximations,"in ACM SIGCOMM Computer Communication Review. ACM, 2019.

Polynomial Time Constant Approximation for Cost VNEP

LP Relaxation of Cost VNEP

LP Formulation 1: Enumerative Cost VNEP – Primal

$$\min \sum_{r\in\mathcal{R}, m_r^k\in\mathcal{M}_r} f_r^k \cdot c_S(m_r^k) \quad (1)$$

$$\sum_{m_r^r\in\mathcal{M}_r} f_r^k = 1 \qquad \forall r \in \mathcal{R} \quad (2)$$

$$\sum_{r\in\mathcal{R}, m_r\in\mathcal{M}_r} f_r^k A(m_r^k, x) \le d_S(x) \qquad \forall x \in G_S \quad (3)$$

$$f_r^k \ge 0 \qquad r \in \mathcal{R}, \forall m_r^k \in \mathcal{M}_r \quad (4)$$

LP Formulation 2: Enumerative Cost VNEP – Dual

$$\max \sum_{r\in\mathcal{R}} \lambda_r + \sum_{x\in G_S} \mu_x \cdot d_S(x) \quad (5)$$

$$\lambda_r \in \mathbb{R} \qquad \forall r \in \mathcal{R} \quad (6)$$

$$\mu_x \le 0 \qquad \forall x \in G_S \quad (7)$$

$$\lambda_r + \sum_{x\in G_S} \mu_x \cdot A(m_r^k, x) \le c_S(m_r^k) \qquad \forall r \in \mathcal{R}, m_r^k \in \mathcal{M}_r \quad (8)$$

(1) minimize sum of costs of fractional embeddings
(2) enforce each request to be fully embedded
(3) respect capacities

Analogously for Profit VNEP

Solve LP Relaxation of CVNEP

Algorithm 3: Solving LP 1 via Column Generation

Output: optimal cost LP solution or $\perp$ if none exists

1 **compute** profit LP solution [6] with $b_r = 1$ for each $r \in \mathscr{R}$
2 **if** LP's solution profit less than $|\mathscr{R}|$ **then**
3 **return** $\perp$ as no solution can exist
4 **initialize** sets of valid mappings $\hat{\mathscr{M}}_r$ using the above LP solution
5 **do**
6 **compute** solution to LP 1 over mapping sets $\{\hat{\mathscr{M}}_r\}_{r \in \mathscr{R}}$
7 **foreach** $r \in \mathscr{R}$ **do**
8 **compute** minimal cost mapping $\hat{m}_r$ using DynVMP under costs $c_\mu(x) := c_S(x) - \mu_x$ for $x \in G_S$
9 **if** $c_\mu(\hat{m}_r) < \lambda_r$ **then add** $\hat{m}_r$ to $\hat{\mathscr{M}}_r$
10 **while** any mapping violating Constraint 8 was added
11 **return** last computed *primal* LP solution

LP Formulation 1: Enumerative Cost VNEP – Primal

$$\min \sum_{r \in \mathscr{R}, m_r^k \in \mathscr{M}_r} f_r^k \cdot c_S(m_r^k) \quad (1)$$

$$\sum_{m_r^k \in \mathscr{M}_r} f_r^k = 1 \qquad \forall r \in \mathscr{R} \quad (2)$$

$$\sum_{r \in \mathscr{R}, m_r \in \mathscr{M}_r} f_r^k A(m_r^k, x) \le d_S(x) \qquad \forall x \in G_S \quad (3)$$

$$f_r^k \ge 0 \qquad r \in \mathscr{R}, \forall m_r^k \in \mathscr{M}_r \quad (4)$$

LP Formulation 2: Enumerative Cost VNEP – Dual

$$\max \sum_{r \in \mathscr{R}} \lambda_r + \sum_{x \in G_S} \mu_x \cdot d_S(x) \quad (5)$$

$$\lambda_r \in \mathbb{R} \qquad \forall r \in \mathscr{R} \quad (6)$$

$$\mu_x \le 0 \qquad \forall x \in G_S \quad (7)$$

$$\lambda_r + \sum_{x \in G_S} \mu_x \cdot A(m_r^k, x) \le c_S(m_r^k) \qquad \forall r \in \mathscr{R}, m_r^k \in \mathscr{M}_r \quad (8)$$

Theorem 1: Alg. 3 solves LP 1 optimally in

$$\mathcal{O}\left(\text{poly}\left(\sum_{r \in \mathscr{R}} n_r^3 \cdot n_S^{2 \cdot tw(\mathcal{T}_r)+3}\right)\right)$$

Approximation of (the Integer) CVNEP

Algorithm 5: Approximation of Cost VNEP

Input : substrate G_S with cost $c_S : G_S \to \mathbb{R}_{\geq 0}$, requests $\mathscr{R}$, approx. factors α, β, γ and N rounding tries

Output: (α, β, γ)-approx. solution whp. or $\perp$ if none exists

compute optimal fractional cost LP solution using Algorithm 3
if *LP solution is* $\perp$ **then**
  return $\perp$ // no feasible embedding exists
// post-process: prune costly mappings
foreach $r \in \mathscr{R}$ **do**
  compute $WC_r \leftarrow \sum_{m_r^k \in \hat{\mathscr{M}}_r} f_r^k \cdot c_S(m_r^k)$
  set $f_r^k \leftarrow 0$ for $m_r^k \in \mathscr{M}_r$ with $c_S(m_r^k) > \alpha \cdot WC_r$
  normalize weights such that $\sum_{m_r^k \in \hat{\mathscr{M}}_r} f_r^k = 1$ holds again
do // perform randomized rounding
  foreach $r \in \mathscr{R}$ **embed** r **using** m_r^k **with probabili**
while *solution not* (α, β, γ)*-approximate and* $\leq N$ *trie*

Theorem 2: Algorithm 5 (α, β, γ)-approximates CVNEP.

Approximation ratios β, γ (for node and edge capacity exceedance, resp.) are:

$$\beta = \alpha/(\alpha-1) + \varepsilon \cdot \sqrt{2 \cdot \Delta(V_S) \cdot \log(n_S)}$$

$$\gamma = \alpha/(\alpha-1) + \varepsilon \cdot \sqrt{2 \cdot \Delta(E_S) \cdot \log(m_S)}$$

with $\Delta(X) = \max_{x \in X} \sum_{r \in \mathscr{R}: d_{\max}(r,x)>0} (A_{\max}(r,x)/d_{\max}(r,x))^2$ *being the maximal sum of squared maximal allocation-to-capacity ratios over the resource set* X *and the maximum demand-to-capacity ratio* $\varepsilon = \max_{r \in \mathscr{R}, x \in G_S} d_{\max}(r,x)/d_S(x)$.

Runtime in $\mathcal{O}\left(\text{poly}\left(\sum_{r \in \mathscr{R}} n_r^3 \cdot n_S^{2 \cdot tw(\mathcal{T}_r)+3}\right)\right)$

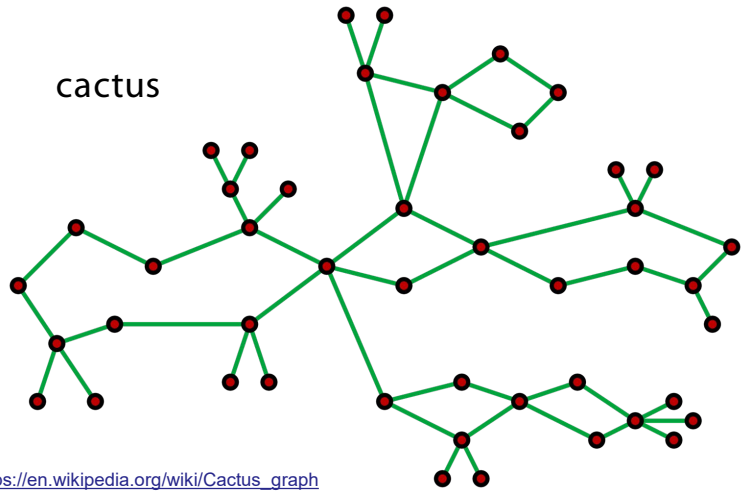
differs from profit version by pruning costly mappings

Evaluation

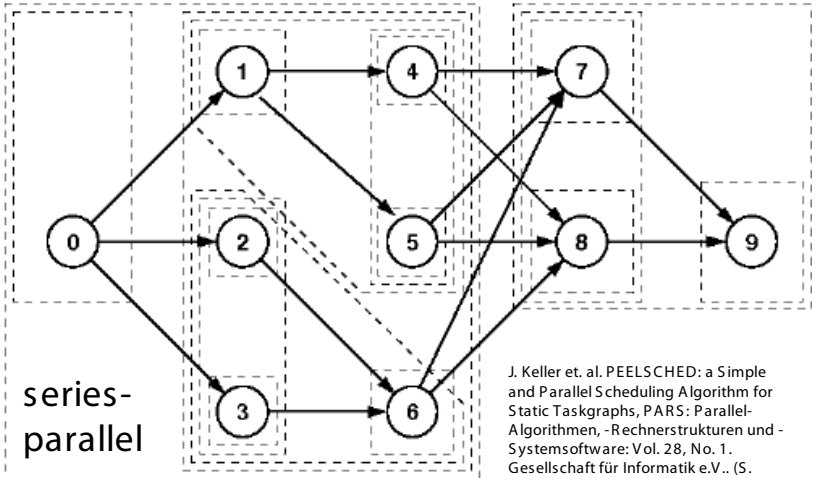
Synthetic Experiments

Substrate: cactus graphs
(model industrial communication networks)

Request: series-parallel graphs
(model control loops, map-reduce apps, etc.)



https://en.wikipedia.org/wiki/Cactus_graph

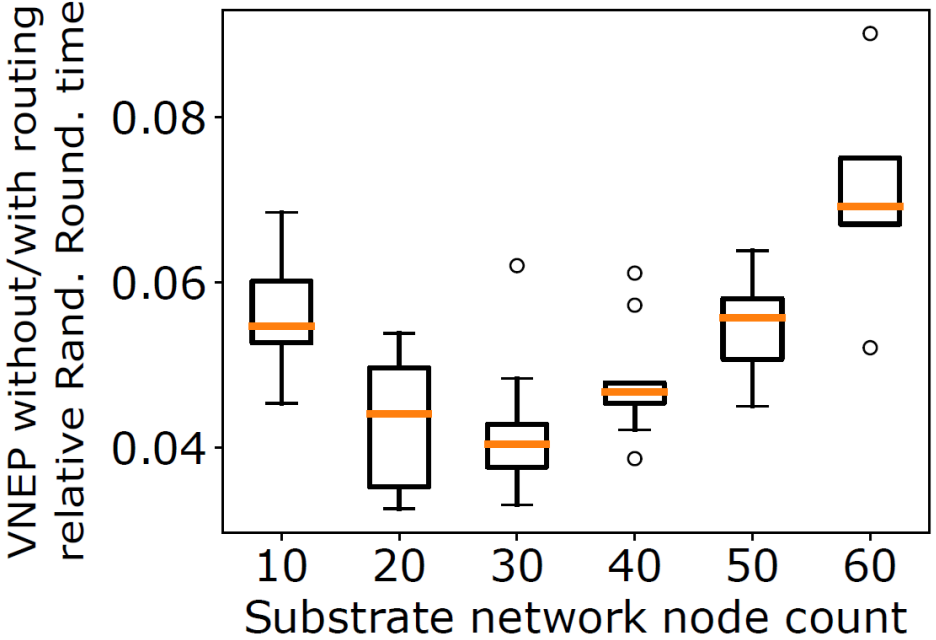


J. Keller et. al. PEELSCHED: a Simple and Parallel Scheduling Algorithm for Static Taskgraphs, PARS: Parallel-Algorithmen, -Rechnerstrukturen und -Systemsoftware: Vol. 28, No. 1. Gesellschaft für Informatik e.V.. (S. 100-109). DOI: 10.1007/BF03341989

- Both graphs are randomly generated with 2x more nodes in request graphs than in substrate

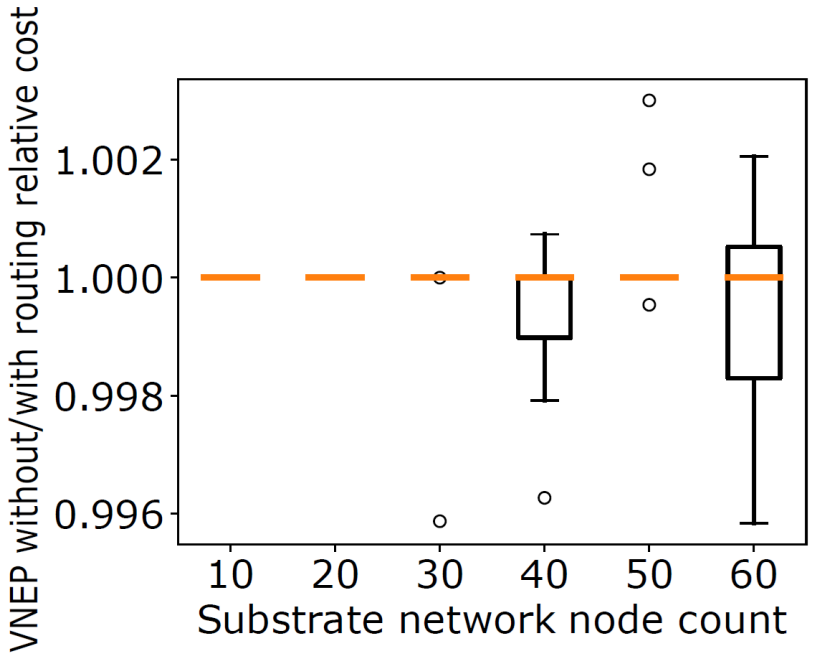
cactus https://www.graphclasses.org/classes/gc_108.html
series-parallel: https://www.graphclasses.org/classes/gc_275.html

Synthetic Experiment Results (1/2)



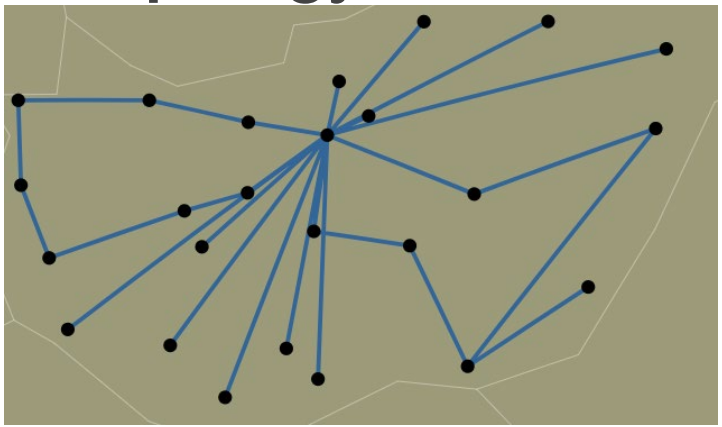
- Relative rounding time of running **without** routing constraints to **with** routing constraints.
- Only 4-8% of time required for rounding solutions **without**
- Reason: **with** routing constraints more valid embeddings are generated (Algoritm 3 runs longer)

Synthetic Experiment Results (2/2)



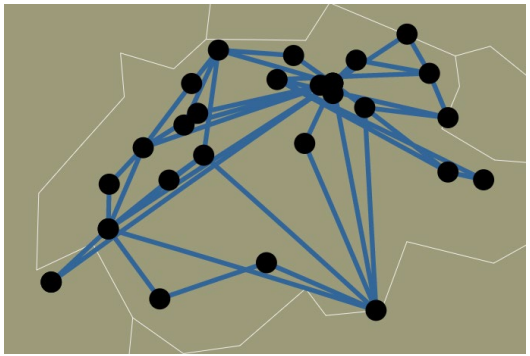
- Relative integer cost between the two variants
- Cost are not really influenced by the problem variant
- Shortest path routing is a good edge embedding

Topology Zoo Substrates



GtsHungary (30 nodes)

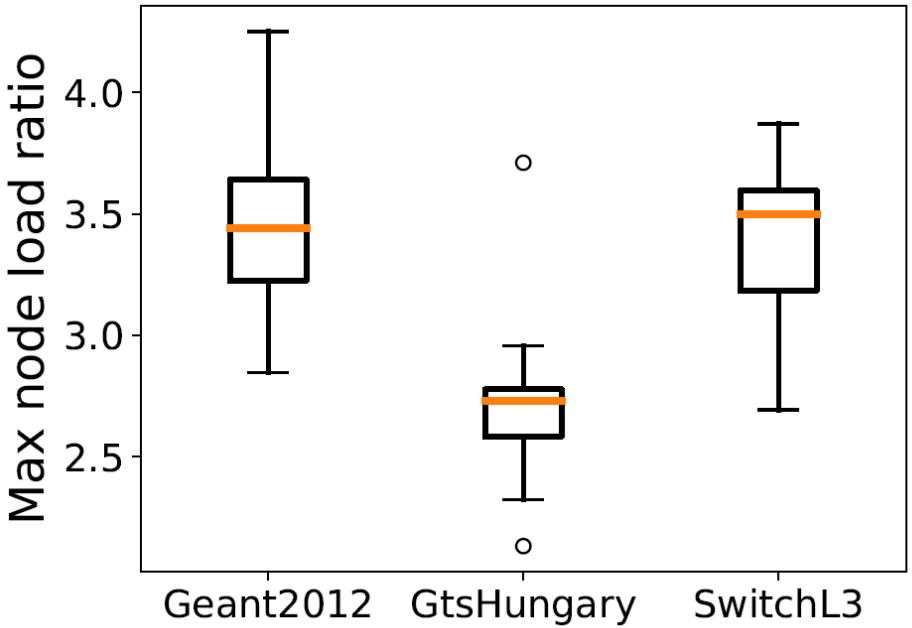
series-parallel request graphs as for purely synthetic evaluation



SwitchL3 (42 nodes)

Geant2012 (40 nodes)

Topology Zoo Results



- Max node load : node capacity violation (violation bound is 5.0)
- Networks with mores nodes have higher max violation on average
- Reason: DynVMP tends to collocate nodes, rounding can only work from those generated mappings

Conclusion

- Virtual Network Embedding Problem (VNEP) is fundamental and intensively studied
- First constant approximation for cost VNEP
- Consider VNEP with fixed routing paths
- Simulation results on synthetic and real-world topologies show applicability