It's About Time: On Optimal Virtual Network Embeddings under Temporal Flexibilities

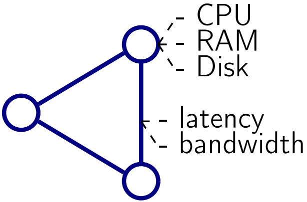
IPDPS 2014

Matthias Rost, Stefan Schmid, Anja Feldmann
Technische Universität Berlin

May 24th, 2014
Phoenix, Arizona

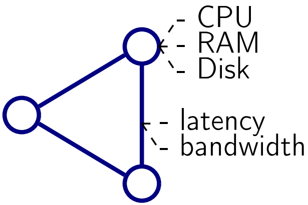
The Virtual Network Embedding Problem (VNEP)

Physical Network

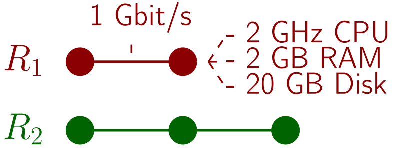


The Virtual Network Embedding Problem (VNEP)

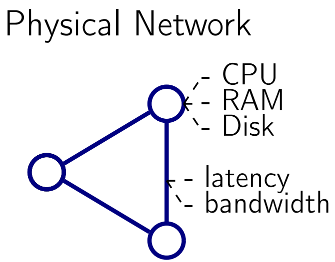
Physical Network

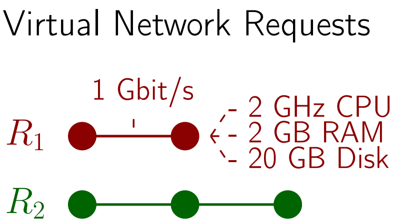


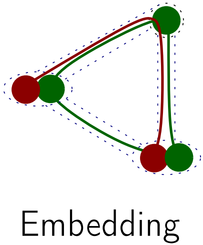
Virtual Network Requests



The Virtual Network Embedding Problem (VNEP)







- map virtual onto substrate nodes
- map virtual links onto substrate paths
- obeying the substrate's capacities

Facets of the VNEP

Setting

(De)centralized

Multi-Provider

Reliability

Reconfigurations

Objectives

Access Control

Load Balancing

Energy Savings

Algorithms

Exact

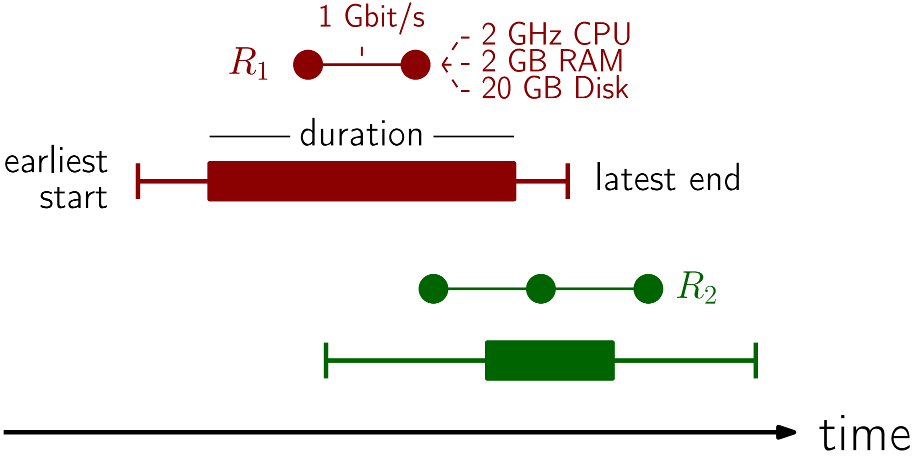
Heuristic

Facets of the VNEP

Setting	Objectives	Algorithms
(De)centralized	Access Control	Exact
Multi-Provider	Load Balancing	Heuristic
Reliability	Energy Savings	
Reconfigurations		

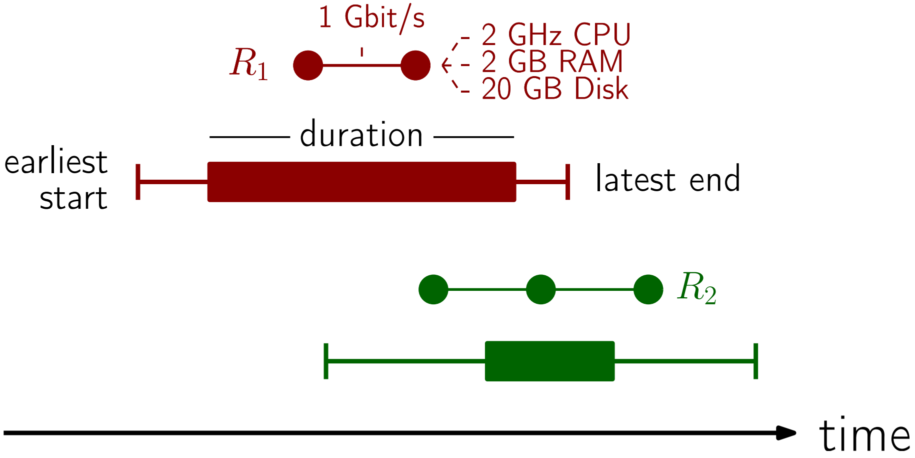
Novelty: Temporality!

Our Model


1 Gbit/s
R_1
2 GHz CPU
2 GB RAM
20 GB Disk
duration
earliest start
latest end
R_2
time

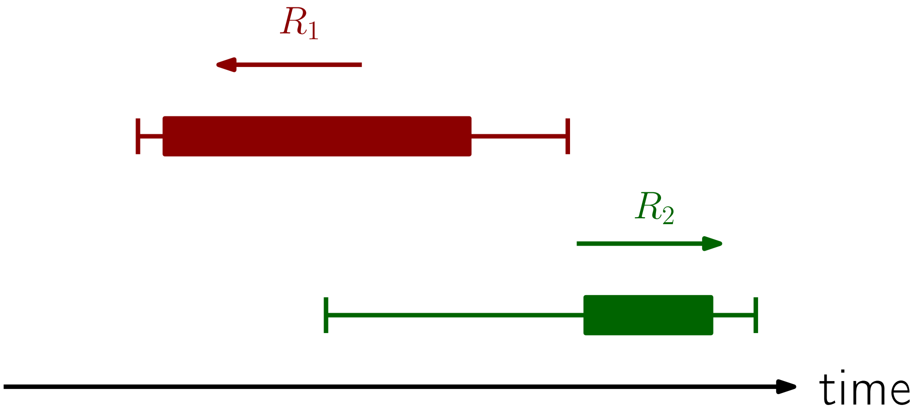
Our Model

Offline scenario

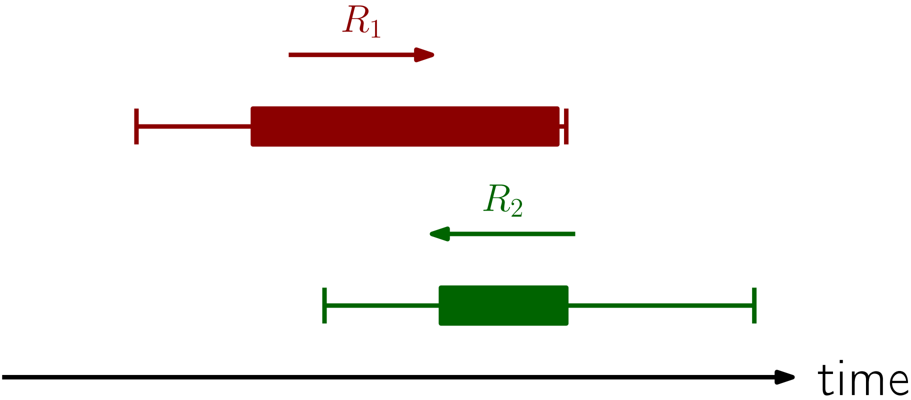


Motivation #1: Business

Provider Incentives: Minimizing Load

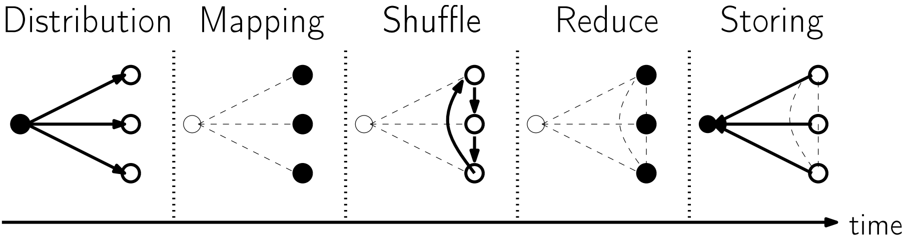


Provider Incentives: Maximizing Utilization by Collocation



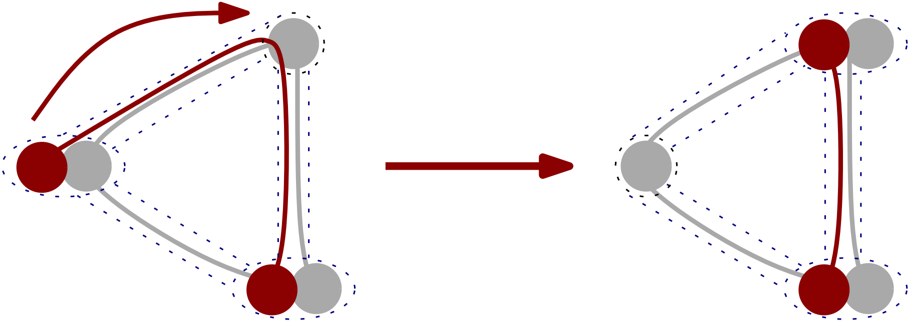
Motivation #2: Modeling Opportunities

Modeling Opportunities: Evolution of VNets

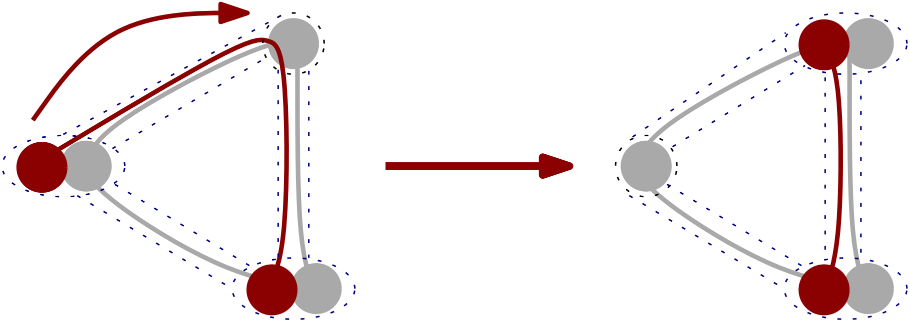


Reservation of maximal allocations over the whole time?

Modeling Opportunities: Migrations

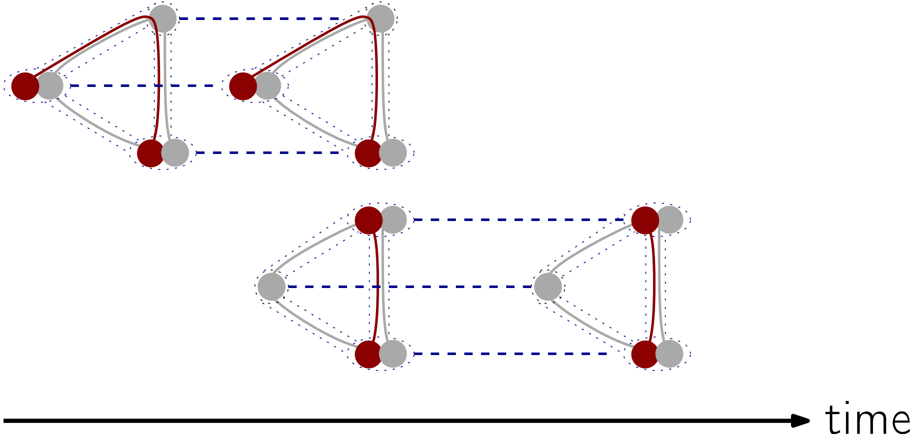


Modeling Opportunities: Migrations

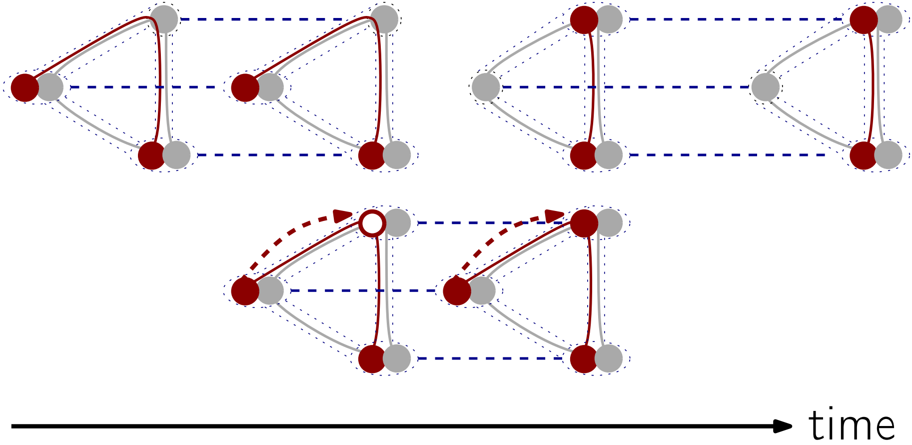


In previous work instantaneous operation!

Modeling Opportunities: Migrations

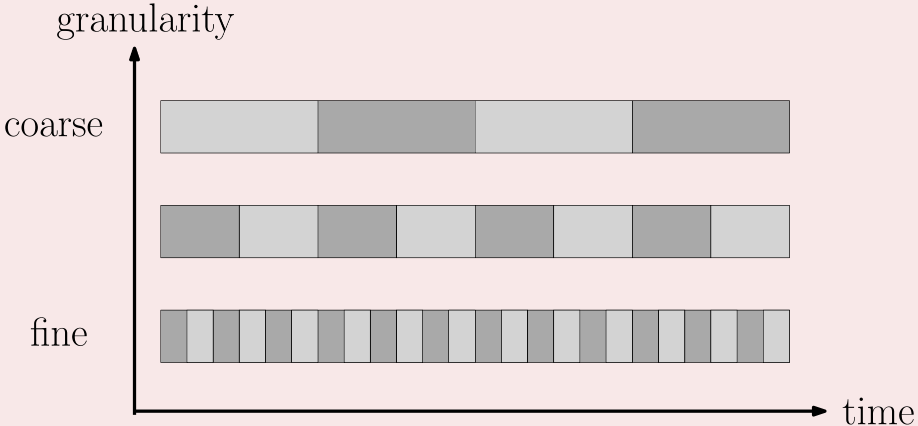


Modeling Opportunities: Fine-grained Migrations



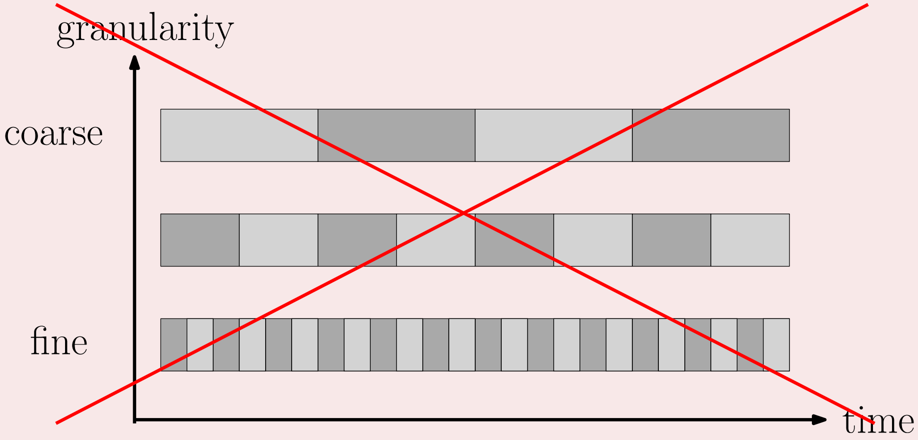
Important Decision: Continuous-Time Model!

Discretization



Important Decision: Continuous-Time Model!

No Discretization!



Problem Statement

Notation

Substrate $\mathcal{S}$

topology $\mathcal{S} = (\mathbf{V_S}, \mathbf{E_S})$

capacities $\mathbf{c_S} : \mathbf{V_S} \cup \mathbf{E_S} \to \mathbb{R}^+$

time horizon $\mathbf{T} > 0$

Requests $\mathcal{R} = \{R_1, \ldots, R_n\}$

topologies $(\mathbf{V}_{R_i}, \mathbf{E}_{R_i})$

resources $\mathbf{c}_{R_i} : \mathbf{V}_{R_i} \cup \mathbf{E}_{R_i} \to \mathbb{R}^+$

temporal spec interval $[\mathbf{t}^s_{R_i}, \mathbf{t}^e_{R_i}]$
duration $\mathbf{d}_{R_i} \leq \mathbf{t}^e_{R_i} - \mathbf{t}^s_{R_i}$

Temporal Virtual Network Embedding Problem (TVNEP)

Access Control Decide which of the requests to embed.

Resource Mapping Map virtual onto substrate resources, obtaining
$alloc_V : \mathcal{R} \times \mathbf{V_S} \to \mathbb{R}_{\geq 0}$ and
$alloc_E : \mathcal{R} \times \mathbf{E_S} \to \mathbb{R}_{\geq 0}$.

Scheduling Find start $t^+_\mathrm{R} \geq \mathbf{t}^s_\mathrm{R}$ and end time $t^-_\mathrm{R} \leq \mathbf{t}^e_\mathrm{R}$ for $\mathrm{R} \in \mathcal{R}$, such that $t^-_\mathrm{R} + t^+_\mathrm{R} = \mathbf{d}_\mathrm{R}$ holds.

Feasibility For each point in time $t \in [0, \mathbf{T}]$ ensure:

$$\forall N_s \in \mathbf{V_S}.\ \mathbf{c_S}(N_s) \geq \sum_{\substack{\mathrm{R} \in \mathcal{R} \text{ with} \\ t \in (t^+_\mathrm{R}, t^-_\mathrm{R})}} alloc_V(\mathrm{R}, N_s)\ ,$$

$$\forall L_s \in \mathbf{E_S}.\ \mathbf{c_S}(L_s) \geq \sum_{\substack{\mathrm{R} \in \mathcal{R} \text{ with} \\ t \in (t^+_\mathrm{R}, t^-_\mathrm{R})}} alloc_E(\mathrm{R}, L_s)\ .$$

Local Embedding

For sake of simplicity, treat mapping as a black box (see e.g.[1]).

Classic VNEP Task

Access Control	Decide which of the requests to embed: $x_{\mathcal{R}} : \mathcal{R} \rightarrow \mathbb{B}$.
Resource Mapping	Map virtual onto substrate resources, obtaining $alloc_V : \mathcal{R} \times \mathbf{V_S} \rightarrow \mathbb{R}_{\geq 0}$ and $alloc_E : \mathcal{R} \times \mathbf{E_S} \rightarrow \mathbb{R}_{\geq 0}$.

Overview

Overview

Contributions

1. Continuous-time Mixed-Integer Programming formulations for TVNEP
2. cΣ-Model utilizes state-space and symmetry reductions to render solving TVNEP (computationally) feasible
3. Greedy polynomial time heuristic which is based on cΣ-Model
4. Initial computational evaluation

Why Mixed-Integer Programming?

- TVNEP is a novel problem: baseline for further work
- Offline scenario: trade-off runtime with solution quality

Mixed-Integer Programming Models

Standard VNEP

Access Control Decide which of the requests to embed: $x_{\mathcal{R}} : \mathcal{R} \to \mathbb{B}$.

Resource Mapping Map virtual onto substrate resources, obtaining

$alloc_V : \mathcal{R} \times \mathbf{V_S} \to \mathbb{R}_{\geq 0}$ and

$alloc_E : \mathcal{R} \times \mathbf{E_S} \to \mathbb{R}_{\geq 0}$.

Novel: Continuous-Time Scheduling

Scheduling Find start $t^+_{\mathrm{R_i}} \geq \mathbf{t}^s_{\mathrm{R_i}}$ and end time $t^-_{\mathrm{R_i}} \leq \mathbf{t}^e_{\mathrm{R_i}}$, such that $t^-_{\mathrm{R_i}} + t^+_{\mathrm{R_i}} = \mathbf{d}_{\mathrm{R_i}}$ holds.

Feasibility For each point in time $t \in [0, \mathbf{T}]$:

$$\forall N_s \in \mathbf{V_S}.\ \mathbf{c_S}(N_s) \geq \sum_{\substack{\mathrm{R} \in \mathcal{R} \text{ with} \\ t \in (t^+_{\mathrm{R}}, t^-_{\mathrm{R}})}} alloc_V(\mathrm{R}, N_s) \ ,$$

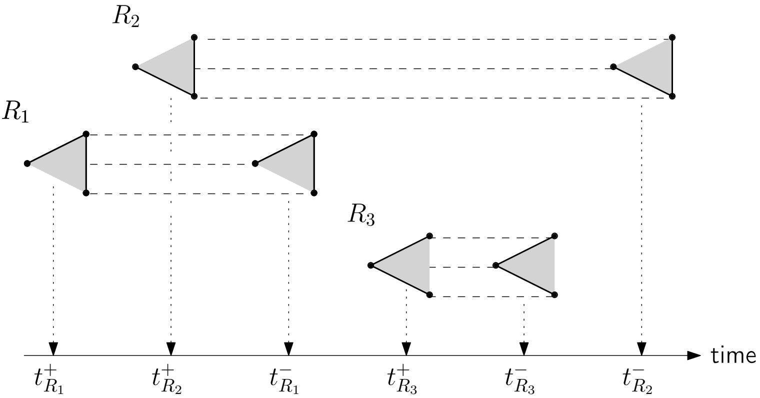
$$\forall L_s \in \mathbf{E_S}.\ \mathbf{c_S}(L_s) \geq \sum_{\substack{\mathrm{R} \in \mathcal{R} \text{ with} \\ t \in (t^+_{\mathrm{R}}, t^-_{\mathrm{R}})}} alloc_E(\mathrm{R}, L_s) \ .$$

Modeling Continuous-Time: Checking Feasibility

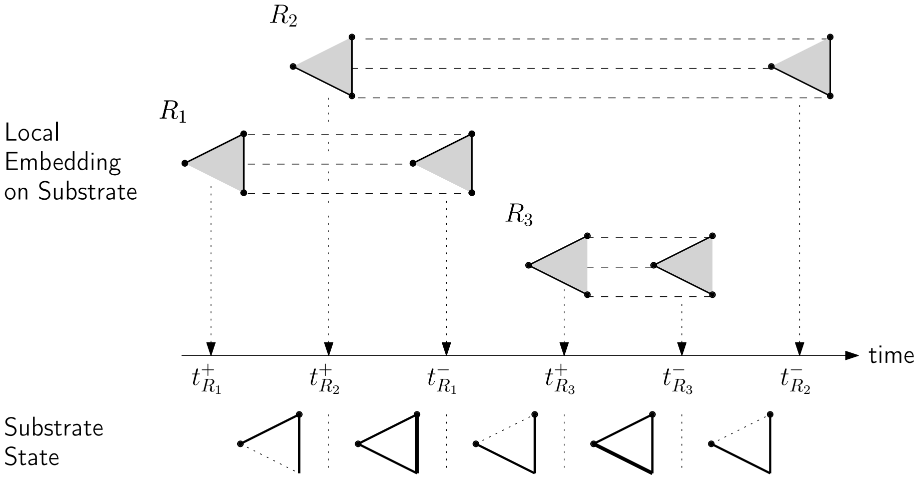
Assume for now:
Local embeddings and start / end times are fixed.

Modeling Continuous-Time: Checking Feasibility

Local
Embedding
on Substrate

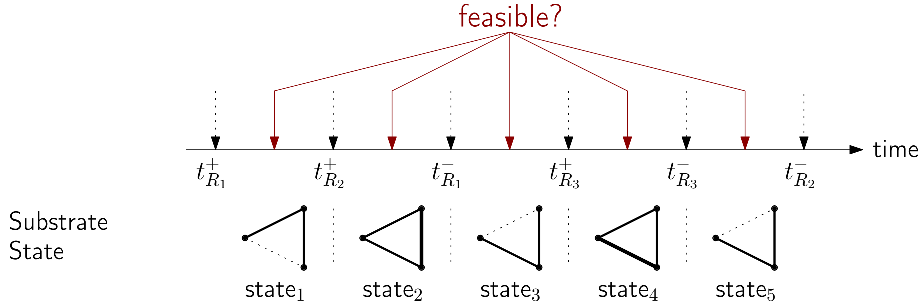


Modeling Continuous-Time: Checking Feasibility



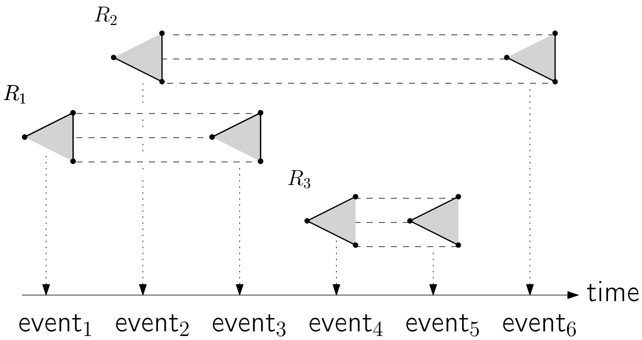
Modeling Continuous-Time: Checking Feasibility

Check the feasibility of the $2 \cdot |\mathcal{R}| - 1$ states.



Abstract Event Model

Modeling Continuous-Time: Abstract Event Model



Mapping Variables

$$\mathcal{E} = \{\mathbf{e}_1, \ldots, \mathbf{e}_{2\cdot|\mathcal{R}|}\}$$

$$\forall \mathsf{R} \in \mathcal{R}.\ \chi^+_{\mathsf{R}} \colon \mathcal{E} \to \mathbb{B}$$

$$\forall \mathsf{R} \in \mathcal{R}.\ \chi^-_{\mathsf{R}} \colon \mathcal{E} \to \mathbb{B}$$

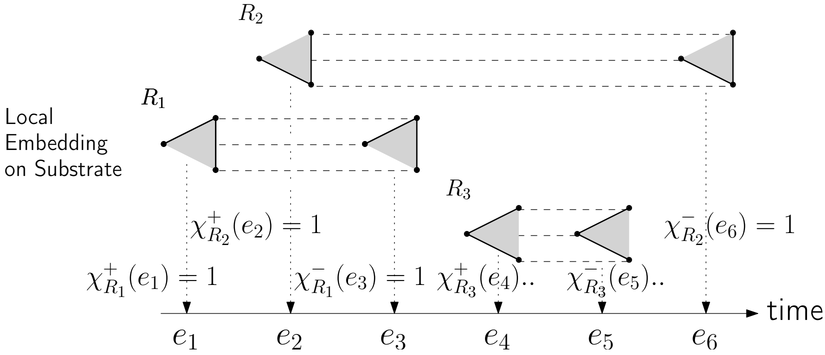
Bijective Mapping

$$\forall \mathsf{R} \in \mathcal{R}.\ \sum_{\mathbf{e}_i \in \mathcal{E}} \chi^+_{\mathsf{R}}(\mathbf{e}_i) = 1 \wedge \sum_{\mathbf{e}_i \in \mathcal{E}} \chi^-_{\mathsf{R}}(\mathbf{e}_i) = 1$$

$$\forall \mathbf{e}_i \in \mathcal{E}.\ \sum_{\mathsf{R} \in \mathcal{R}} \chi^+_{\mathsf{R}}(\mathbf{e}_i) = 1 \wedge \sum_{\mathsf{R} \in \mathcal{R}} \chi^-_{\mathsf{R}}(\mathbf{e}_i) = 1$$

Δ-Model

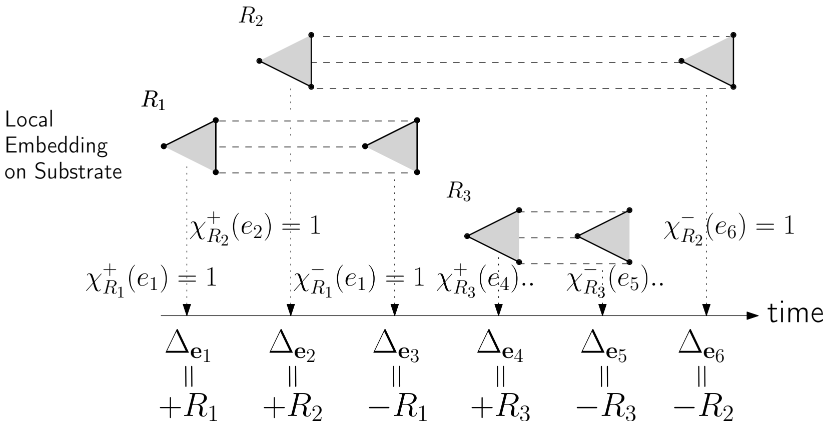
Reconstructing States: Δ-Model



Idea

- Compute state changes via mapping variables $\chi^+_{\mathsf{R}}(e_i), \chi^-_{\mathsf{R}}(e_i)$

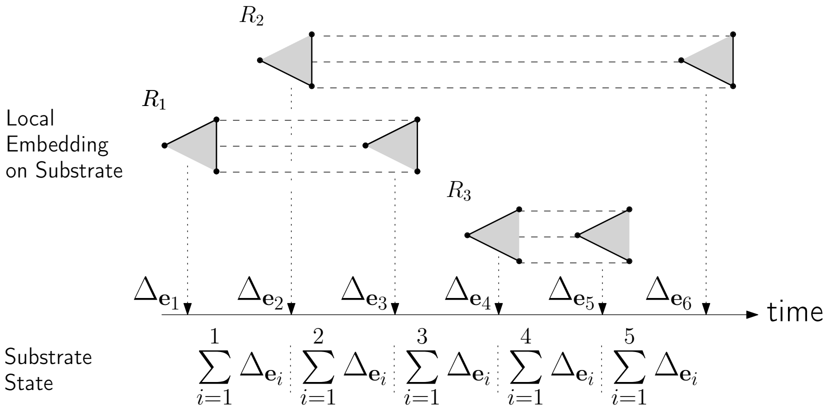
Reconstructing States: Δ-Model



Idea

- Compute state *changes*: $\Delta_{\mathbf{e}_i} : \mathbf{V_S} \cup \mathbf{E_S} \to \mathbb{R}$ via $\chi^+_{\mathsf{R}}(e_i), \chi^-_{\mathsf{R}}(e_i)$

Reconstructing States: Δ-Model



Idea

1. Compute state *changes*: $\Delta_{\mathbf{e}_i} : \mathbf{V_S} \cup \mathbf{E_S} \to \mathbb{R}$ via $\chi^+_{\mathsf{R}}(e_i), \chi^-_{\mathsf{R}}(e_i)$
2. Enforce $\sum_{j=1}^{i} \Delta_{\mathbf{e}_i} \leq \mathbf{c_S}$ for each state

Δ-Model: Computing State Changes

Conditional Assignment

$\forall \mathbf{e}_i \in \mathcal{E}.\ \forall N_s \in \mathbf{V_S}.$

$$\Delta_{\mathbf{e}_i}(N_s) = \begin{cases} +alloc_V(\mathsf{R_j}, N_s) & \text{, if } \chi^{+}_{\mathsf{R_j}}(\mathbf{e}_i) = i \\ -alloc_V(\mathsf{R_j}, N_s) & \text{, if } \chi^{-}_{\mathsf{R_j}}(\mathbf{e}_i) = i \\ \vdots & \\ +alloc_V(\mathsf{R_k}, N_s) & \text{, if } \chi^{+}_{\mathsf{R_k}}(\mathbf{e}_i) = i \\ -alloc_V(\mathsf{R_k}, N_s) & \text{, if } \chi^{-}_{\mathsf{R_k}}(\mathbf{e}_i) = i \end{cases}$$

Δ-Model: Computing State Changes

Conditional Assignment via Big-M Constraints

$\forall \mathrm{R} \in \mathcal{R}.\ \forall \mathbf{e}_i \in \mathcal{E}.\ \forall N_s \in \mathbf{V_S}.$

$$\Delta_{\mathbf{e}_i}(N_s) \leq +\, alloc_V(\mathrm{R}, N_s) + \mathbf{c_S}(N_s)(1 - \chi^{+}_{\mathrm{R}}(\mathbf{e}_i))$$
$$\Delta_{\mathbf{e}_i}(N_s) \geq +\, alloc_V(\mathrm{R}, N_s) - \mathbf{c_S}(N_s)(1 - \chi^{+}_{\mathrm{R}}(\mathbf{e}_i)) \cdot 2$$
$$\Delta_{\mathbf{e}_i}(N_s) \leq -\, alloc_V(\mathrm{R}, N_s) + \mathbf{c_S}(N_s)(1 - \chi^{-}_{\mathrm{R}}(\mathbf{e}_i)) \cdot 2$$
$$\Delta_{\mathbf{e}_i}(N_s) \geq -\, alloc_V(\mathrm{R}, N_s) - \mathbf{c_S}(N_s)(1 - \chi^{-}_{\mathrm{R}}(\mathbf{e}_i))$$

Δ-Model: Computing State Changes

Big-M Assignment of Start

$\forall \mathrm{R} \in \mathcal{R}.\ \forall \mathbf{e}_i \in \mathcal{E}.\ \forall N_s \in \mathbf{V_S}.$

$$\Delta_{\mathbf{e}_i}(N_s) \leq + \mathit{alloc}_V(R, N_s) + \mathbf{c_S}(N_s)(1 - \chi^{+}_{\mathrm{R}_1}(\mathbf{e}_i))$$
$$\Delta_{\mathbf{e}_i}(N_s) \geq + \mathit{alloc}_V(R, N_s) - \mathbf{c_S}(N_s)(1 - \chi^{+}_{\mathrm{R}_1}(\mathbf{e}_i)) \cdot 2$$

$\chi^{+}_{\mathrm{R}}(\mathbf{e}_i) = 0$

$$\Delta_{\mathbf{e}_i}(N_s) \leq + \mathit{alloc}_V(\mathrm{R}, N_s) + \mathbf{c_S}(N_s)$$
$$\Delta_{\mathbf{e}_i}(N_s) \geq + \mathit{alloc}_V(\mathrm{R}, N_s) - 2 \cdot \mathbf{c_S}(N_s)$$

$\Rightarrow$

unbounded

$$\Delta_{\mathbf{e}_i}(N_s) \leq \mathbf{c_S}(N_s)$$
$$\Delta_{\mathbf{e}_i}(N_s) \geq - \mathbf{c_S}(N_s)$$

Δ-Model: Computing State Changes

Big-M Assignment of Start

$\forall \mathrm{R} \in \mathcal{R}.\ \forall \mathbf{e}_i \in \mathcal{E}.\ \forall N_s \in \mathbf{V_S}.$

$$\Delta_{\mathbf{e}_i}(N_s) \leq +\, alloc_V(R, N_s) + \mathbf{c_S}(N_s)(1 - \chi^+_{\mathrm{R}_1}(\mathbf{e}_i))$$
$$\Delta_{\mathbf{e}_i}(N_s) \geq +\, alloc_V(R, N_s) - \mathbf{c_S}(N_s)(1 - \chi^+_{\mathrm{R}_1}(\mathbf{e}_i)) \cdot 2$$

$\chi^+_{\mathrm{R}}(\mathbf{e}_i) = 1$

$$\Delta_{\mathbf{e}_i}(N_s) \leq +\, alloc_V(\mathrm{R}, N_s)$$
$$\Delta_{\mathbf{e}_i}(N_s) \geq +\, alloc_V(\mathrm{R}, N_s)$$

$\Rightarrow$

equal

$$\Delta_{\mathbf{e}_i}(N_s) = alloc_V(\mathrm{R}, N_s)$$

Δ-Model Issue: LP Smearings!

LP Relaxation Example

$\chi^{+}_{\mathrm{R_j}}(\mathbf{e}_j) = 0.5$ for $j \in \{1, 2\}$:

$$-\mathbf{c_S}(N_s) + alloc_V(\mathrm{R_j}, N_s) \leq \Delta_{\mathbf{e}_j}(N_s) \leq alloc_V(\mathrm{R_j}, N_s) + 0.5 \cdot \mathbf{c_S}(N_s)$$

Δ-Model Issue: LP Smearings!

LP Relaxation Example

$\chi^{+}_{\mathrm{R_j}}(\mathbf{e}_j) = 0.5$ for $j \in \{1, 2\}$:

$$-\mathbf{c_S}(N_s) + alloc_V(\mathrm{R_j}, N_s) \leq \Delta_{\mathbf{e}_j}(N_s) \leq alloc_V(\mathrm{R_j}, N_s) + 0.5 \cdot \mathbf{c_S}(N_s)$$

Implications

1. $\Delta_{\mathbf{e}_j}(s) \leq 0$ is always feasible when $\chi^{+}_{\mathrm{R_j}}(\mathbf{e}_j) = 0.5$ for $j \in \{1, 2\}$
2. $\Delta_{\mathbf{e}_j}(s) = -\mathbf{c_S}(N_s)$ possible if $alloc_V(\mathrm{R_j}, N_s) = 0$

Δ-Model Issue: LP Smearings!

LP Relaxation Example

$\chi^{+}_{\mathrm{R_j}}(\mathbf{e}_j) = 0.5$ for $j \in \{1,2\}$:

$$-\mathbf{c_S}(N_s) + alloc_V(\mathrm{R_j}, N_s) \leq \Delta_{\mathbf{e}_j}(N_s) \leq alloc_V(\mathrm{R_j}, N_s) + 0.5 \cdot \mathbf{c_S}(N_s)$$

Implications

1. $\Delta_{\mathbf{e}_j}(s) \leq 0$ is always feasible when $\chi^{+}_{\mathrm{R_j}}(\mathbf{e}_j) = 0.5$ for $j \in \{1,2\}$
2. $\Delta_{\mathbf{e}_j}(s) = -\mathbf{c_S}(N_s)$ possible if $alloc_V(\mathrm{R_j}, N_s) = 0$

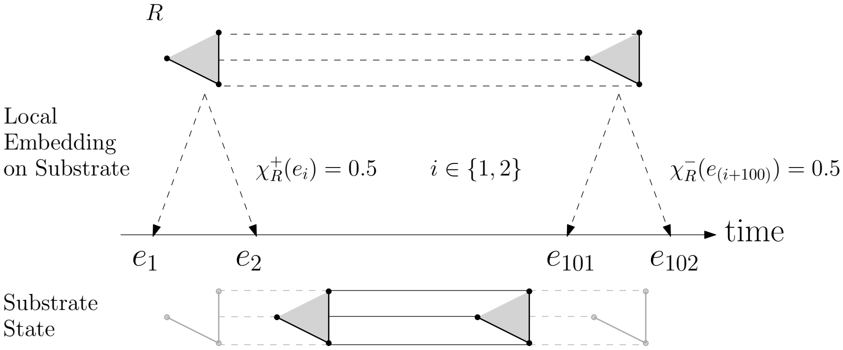
Necessitates more explicit representation of states.

Σ-Model

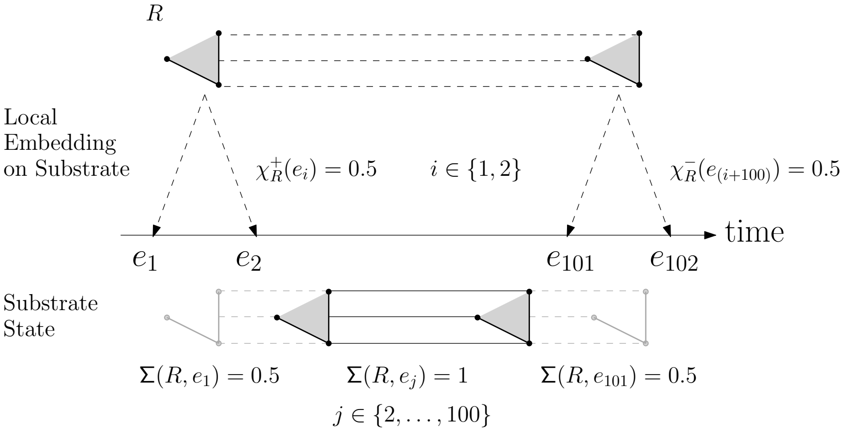
Σ-Model: Intuition

Requirement

Resource allocations must materialize in the substrate's state.



Σ-Model: Intuition



$\forall \mathsf{R} \in \mathcal{R}. \forall \mathbf{e}_i \in \mathcal{E}.$

$$\Sigma(R, \mathbf{e}_i) = \sum_{j=1,\ldots,i} \chi_{\mathsf{R}}^+(\mathbf{e}_j, \mathsf{R}) - \sum_{j=1,\ldots,i} \chi_{\mathsf{R}}^-(\mathbf{e}_j, \mathsf{R})$$

Σ-Model

Request allocations are computed for each state

- States $\mathcal{S} = \{\mathbf{s}_1, \ldots, \mathbf{s}_{2\cdot|\mathcal{R}|-1}\}$
- $\forall \mathrm{R} \in \mathcal{R}. \forall \mathbf{s}_i \in \mathcal{S}. \forall N_s \in \mathbf{V_S}.$

$$\mathrm{a}_{\mathrm{R}}(\mathbf{s}_i, N_s) \geq alloc_V(\mathrm{R}, N_s) - \mathbf{c_V}(N_s) \cdot (1 - \Sigma(R, \mathbf{e}_i))$$

- $\forall \mathbf{s}_i \in \mathcal{S}. \forall r \in \mathbf{V_S}.$

$$\mathbf{c_S}(N_s) \geq \sum_{\mathrm{R} \in \mathcal{R}} \mathrm{a}_{\mathrm{R}}(\mathbf{s}_i, N_s)$$

$\forall \mathrm{R} \in \mathcal{R}. \forall \mathbf{e}_i \in \mathcal{E}.$

$$\Sigma(R, \mathbf{e}_i) = \sum_{j=1,\ldots,i} \chi^+_{\mathrm{R}}(\mathbf{e}_j, \mathrm{R}) - \sum_{j=1,\ldots,i} \chi^-_{\mathrm{R}}(\mathbf{e}_j, \mathrm{R})$$

cΣ-Model

cΣ-Model: Overview

Computational Trade-Off

- The Σ-Model is provably stronger than Δ-Model.
- However, the Σ-Model uses (approximately) $2 \cdot |\mathcal{R}|$ more variables!

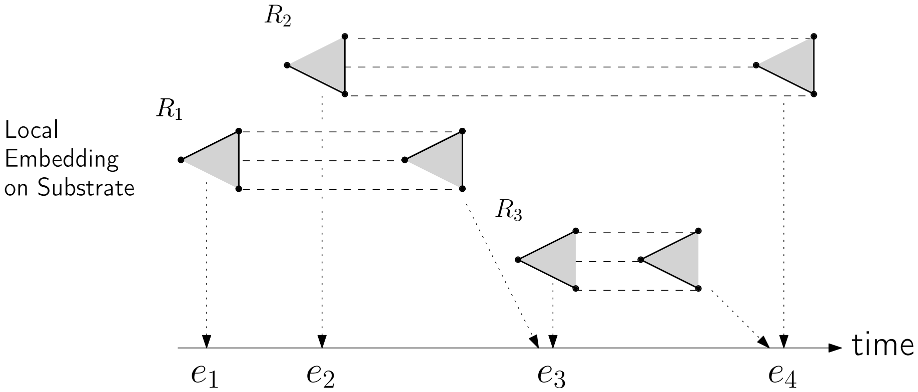
Σ-Model can be strengthened: cΣ-Model

State compactification: Consider only *partial* event order. Yields *state-space* and *symmetry reductions*.

User cuts: Use temporal information to reduce *state-space* and strengthen formulation.

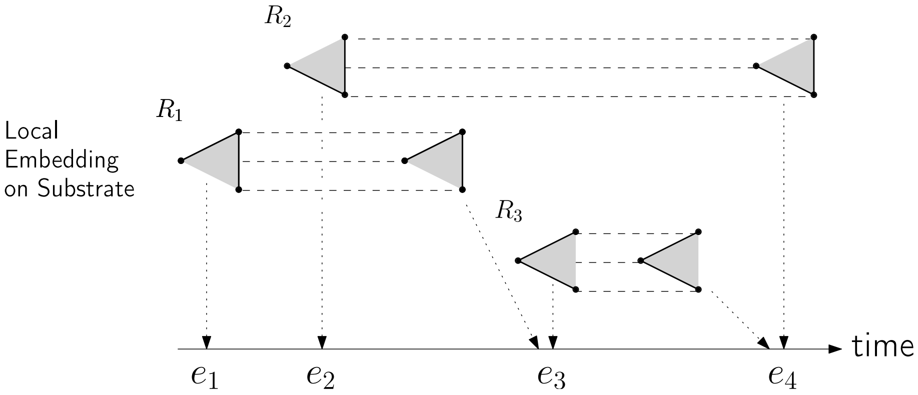
cΣ Optimization: State Compactification

cΣ-Model: State Compactification

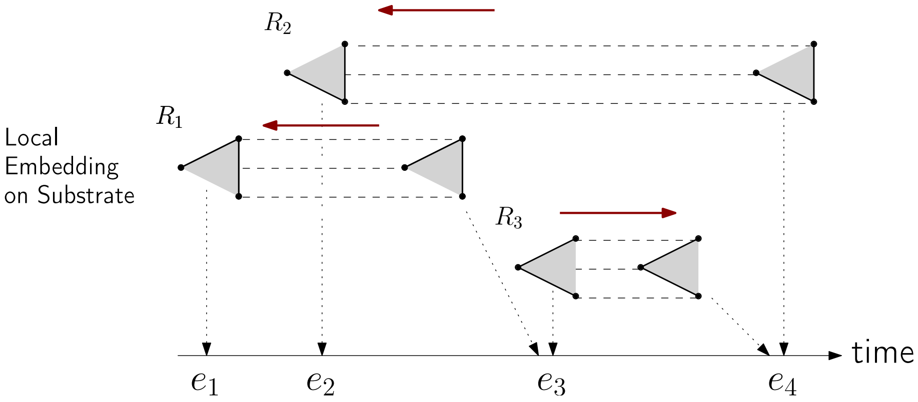


No need to check feasibility at *end* of a request.
Halves the number of variables (approximately):
State-space reduction.

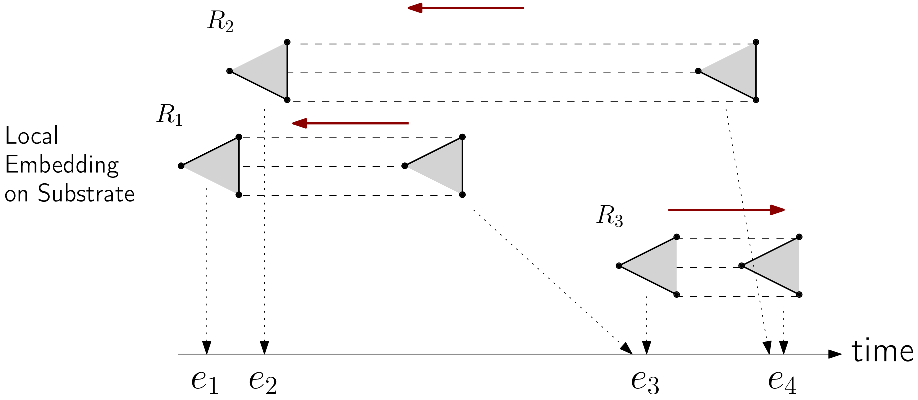
cΣ-Model: State Compactification is Symmetry Reduction



cΣ-Model: State Compactification is Symmetry Reduction

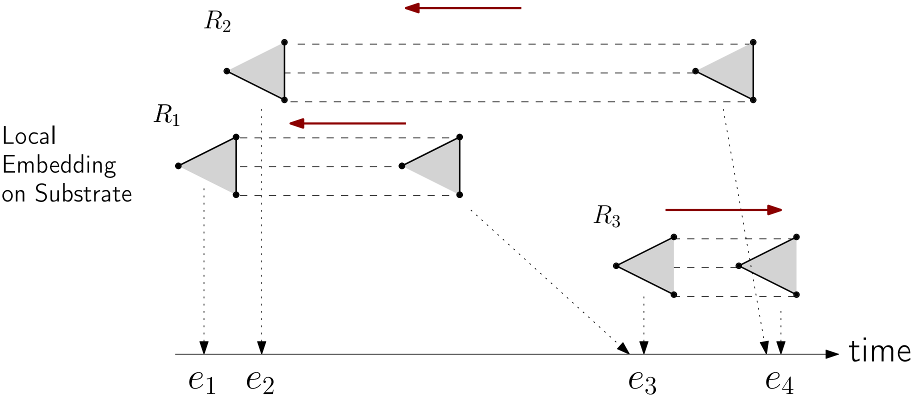


cΣ-Model: State Compactification is Symmetry Reduction



Same order as before!

cΣ-Model: State Compactification is Symmetry Reduction



Theorem

Compactification is symmetry reduction.

The Last Bit: Incorporating Time

cΣ-Model: Incorporating Time

$\forall \mathbf{e}_i \in \{\mathbf{e}_1, \ldots, \mathbf{e}_{|\mathcal{R}|}\}$.

$$t_{\mathbf{e}_i} \leq t_{\mathbf{e}_{i+1}}$$

$\forall R \in \mathcal{R}$.

$$\mathbf{d}_R = t^-{}_R - t^+{}_R$$

$\forall R \in \mathcal{R}. \forall \mathbf{e}_i \in \{\mathbf{e}_1, \ldots, \mathbf{e}_{|\mathcal{R}|}\}$.

$$t_R^+ \leq t_{\mathbf{e}_i} + \Big(1 - \sum_{j=1,\ldots,i} \chi_R^+(\mathbf{e}_j, R)\Big) \cdot \mathbf{T}$$

$$t_R^+ \geq t_{\mathbf{e}_i} - \Big(1 - \sum_{j=i,\ldots,|\mathcal{E}|} \chi_R^+(\mathbf{e}_j, R)\Big) \cdot \mathbf{T}$$

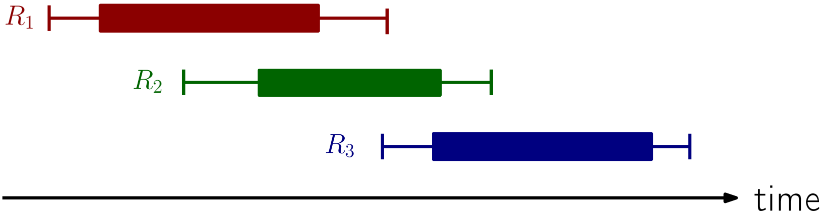
$\forall R \in \mathcal{R}. \forall \mathbf{e}_i \in \{\mathbf{e}_2, \ldots, \mathbf{e}_{|\mathcal{R}|+1}\}$.

$$t_R^- \leq t_{\mathbf{e}_i} + \Big(1 - \sum_{j=2,\ldots,i} \chi_R^-(\mathbf{e}_j, R)\Big) \cdot \mathbf{T}$$

$$t_R^- \geq t_{\mathbf{e}_{i-1}} - \Big(1 - \sum_{j=i,\ldots,|\mathcal{E}|} \chi_R^-(\mathbf{e}_j, R)\Big) \cdot \mathbf{T}$$

Optimizations: Temporal Dependency Graph Cuts

Temporal Dependency Graph Cuts



Temporal Dependency Graph Cuts

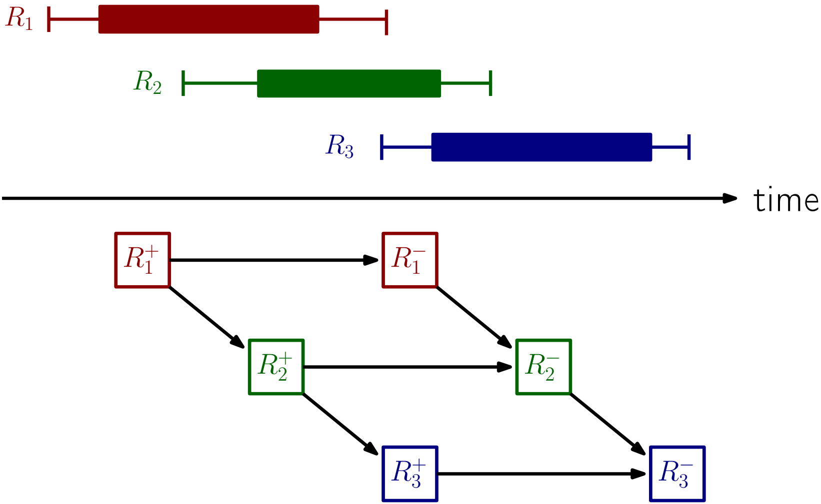


Figure: Temporal Dependency Graph

Temporal Dependency Graph Cuts

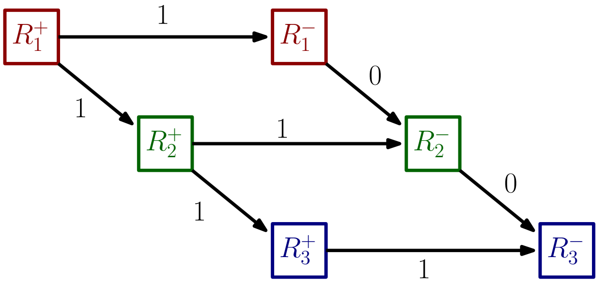


Figure: Temporal Dependency Graph

By computing maximal distances we obtain:

- Start of R_1: e_1
- Start of R_2: e_2
- Start of R_3: e_3
- End of R_1: e_2, e_3, e_4
- End of R_2: e_3, e_4
- End of R_1: e_4

Temporal Dependency Graph Cuts

State-space Reduction via

$\forall v \in V_{dep}.$

$$\sum_{i=|dist^{+}_{\max}(v)|+1}^{|\mathcal{R}|+1-|dist^{-}_{\max}(v)|} \chi_{Event}(\mathbf{e}_i, v) = 1$$

User Cuts:

If R was embedded at e_i, then successor w must have been embedded ..

$\forall v \in V_{dep}.\forall w \in dist^{+}_{\max}(v).\forall \mathbf{e}_i \in \mathcal{E}, dist_{\max}(v,w) + 1 \leq i \leq |\mathcal{R}|.$

$$\sum_{j=1}^{i} \chi_{Event}(\mathbf{e}_j, w) \leq \sum_{\substack{\mathbf{e}_j \in \mathcal{E} \\ \text{with } j \leq i - dist^{-}_{\max}(v,w)}} \chi_{Event}(\mathbf{e}_j, v)$$

Greedy Heuristic $c\Sigma_A^G$

Greedy Heuristic $c\Sigma_A^G$

Setting

Node placements are fixed.

Outline

1. Order requests according to their earliest start time.
2. Iteratively try to embed requests as soon as possible using $c\Sigma$-Model
    1. If the request was embedded: fix start and end time.

Greedy Heuristic $c\Sigma_A^G$

Setting

Node placements are fixed.

Outline

1. Order requests according to their earliest start time.
2. Iteratively try to embed requests as soon as possible using $c\Sigma$-Model
    1. If the request was embedded: fix start and end time.

Theorem

$c\Sigma_A^G$ is polynomial-time algorithm.

Important

All link allocations are re-computed in each iteration.

Computational Evaluation

Scenario: One day workload

- 20 requests (star-graphs) are to be embedded on 4×5 grid
- Expected inter-arrival time of one hour [Poisson]
- Expected duration of 3.5 hours [Weibull: heavy-tailed]
- Node-mappings are fixed to concentrate on temporal aspects
- Link-mappings are not fixed
- Increasing temporal flexibility: $0, 30, 60, \ldots, 300$ minutes.

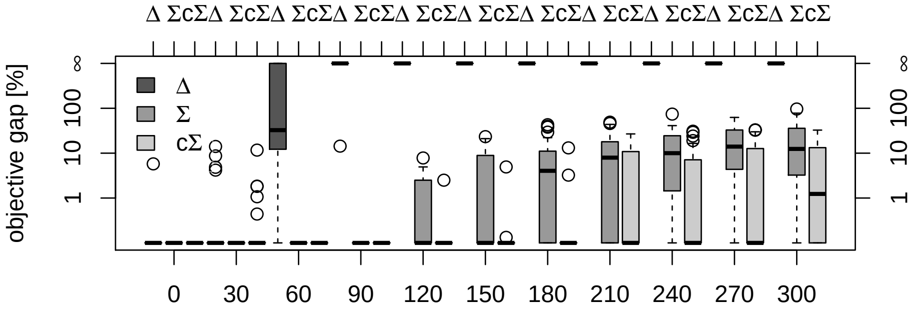
Computational Setup

- 24 independently generated scenarios
- Limited runtime of one hour for MIPs [Gurobi]

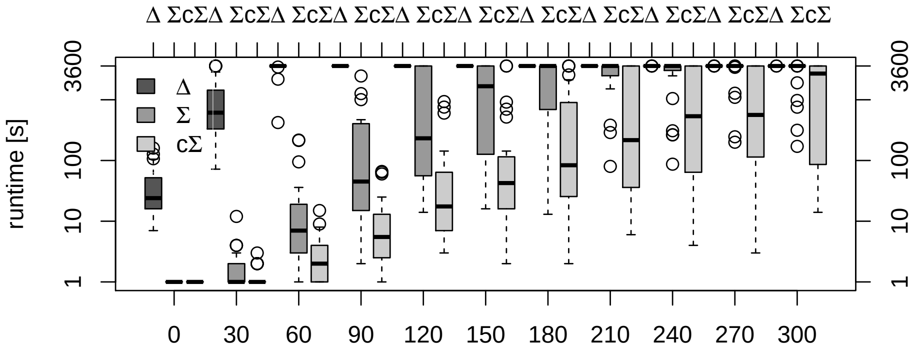
Task: Maximize revenue $\propto$ load $\cdot$ duration

1. Decide which requests to embed (access control).
2. Find time-invariant embedding (routing of data).
3. Decide when to embed the requests.

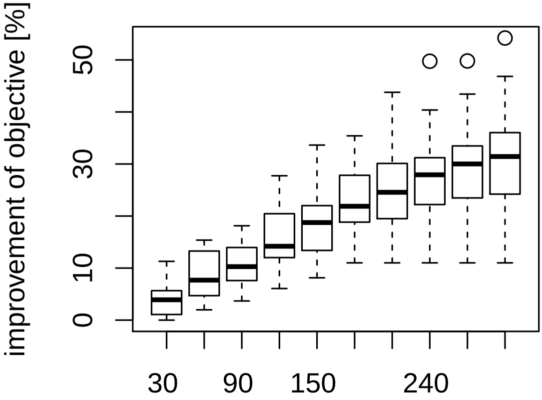
Objective Gap: MIP Formulations



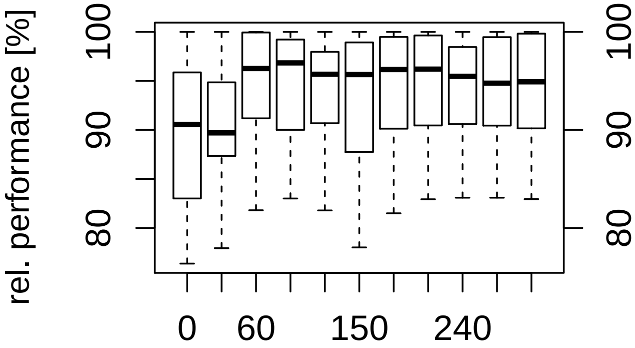
Runtime: MIP Formulations



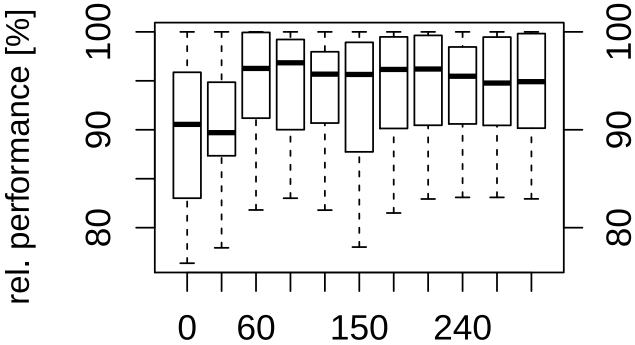
Benefit of Flexibility



Performance of $c\Sigma_A^G$



Performance of $c\Sigma_A^G$



Fast: runtime of few seconds.

Conclusion

Related Work

Chemical plants [3]	Utilize similar event abstraction, but no resource sharing.
Business Perspective [4]	Marketplace based on temporal flexibilities.
MapReduce [5]	Consider temporally predictable jobs (MapReduce-like) and allow for temporally interleaved resource sharing.
VNet Survey [2]	There is no comparable work on TVNEP.
Google B4 [6]	Software-defined network (wide-area) connecting data centers. Only some dozen locations.

Future Work / Discussion

Modeling

- Consider flexible duration of requests.
- Consider delay-tolerant VNets.
- Consider more complex scenarios, e.g. migrations.

Algorithmic

- Incorporate other heuristical embedding approaches.
- Develop local-search algorithms for the TVNEP.

The End

1. Abstract event point model
2. Δ-, Σ- and $c\Sigma$-Model
   - state-space reductions
   - symmetry reduction
3. Greedy heuristic $c\Sigma^G_A$ based on $c\Sigma$
4. Initial computational evaluation
   - $\Delta \ll \Sigma < c\Sigma$
   - $c\Sigma$: near optimal solutions within one hour
   - $c\Sigma^G_A$ only approx. 5-10% off optimum

References I

[1] N. Chowdhury, M. Rahman, and R. Boutaba.
Virtual network embedding with coordinated node and link mapping.
In *Proc. of the IEEE INFOCOM '09*, 2009.

[2] A. Fischer, J. Botero, M. Beck, H. De Meer, and X. Hesselbach.
Virtual network embedding: A survey.
2013.

[3] C. A. Floudas and X. Lin.
Continuous-time versus discrete-time approaches for scheduling of chemical processes: a review.
Computers & Chemical Engineering, 28(11), 2004.

[4] T. A. Henzinger, A. V. Singh, V. Singh, T. Wies, and D. Zufferey.
A marketplace for cloud resources.
In *Proc. of the ACM EMSOFT '10*, 2010.

[5] L. Mai, E. Kalyvianaki, and P. Costa.
Exploiting time-malleability in cloud-based batch processing systems.
In *Proceeding of the ACM SIGOPS LADIS Workshop '13*, 2013.

References II

[6] S. Jain et al.
B4: experience with a globally-deployed software defined wan.
In *Proc. of the ACM SIGCOMM '13*, 2013.

[7] A. Singla, A. Singh, K. Ramachandran, L. Xu, and Y. Zhang.
Proteus: A topology malleable data center network.
In *Proc. of the ACM SIGCOMM HotNets Workshop '10*, 2010.

Standard VNEP Access Control & Resource Mapping

Variables

Access Control		$\mathrm{x}_{\mathcal{R}} : \mathcal{R} \to \mathbb{B}$
Node Mapping	$\forall \mathrm{R} \in \mathcal{R}.$	$\mathrm{x}_V : \mathbf{V}_\mathrm{R} \times \mathbf{V}_\mathbf{S} \to \mathbb{B}$
Link Mapping	$\forall \mathrm{R} \in \mathcal{R}.$	$\mathrm{x}_E : \mathbf{E}_\mathrm{R} \times \mathbf{E}_\mathbf{S} \to [0, 1]$

Node mapping

Map each virtual onto a substrate node, if the request is embedded.

Link mapping

Map each virtual link onto multiple paths in the substrate (splittable flows).

Macro $alloc_V(R, N_s)$: $\forall R \in \mathcal{R}. \forall N_s \in \mathbf{V_S}$

$$alloc_V(R, N_s) = \sum_{N_v \in \mathbf{V}_\mathrm{R}} \mathsf{c}_\mathrm{R}(N_v) \cdot \mathrm{x}_V(N_v, N_s)$$

Macro $alloc_V(R, N_s)$: $\forall R \in \mathcal{R}. \forall L_s \in \mathbf{E_S}$

$$alloc_E(R, L_s) = \sum_{L_v \in \mathbf{E}_\mathrm{R}} \mathsf{c}_\mathrm{R}(L_v) \cdot \mathrm{x}_E(L_v, L_s)$$

Standard VNEP Access Control & Resource Mapping

Variables

Access Control		$\mathrm{x}_{\mathcal{R}} : \mathcal{R} \to \mathbb{B}$
Node Mapping	$\forall \mathrm{R} \in \mathcal{R}.$	$\mathrm{x}_V : \mathbf{V}_\mathrm{R} \times \mathbf{V}_\mathbf{S} \to \mathbb{B}$
Link Mapping	$\forall \mathrm{R} \in \mathcal{R}.$	$\mathrm{x}_E : \mathbf{E}_\mathrm{R} \times \mathbf{E}_\mathbf{S} \to [0,1]$

Node mapping: $\forall \mathrm{R} \in \mathcal{R}.\ \forall N_v \in \mathbf{V}_\mathrm{R}.$

$$\mathrm{x}_{\mathcal{R}}(\mathrm{R}) = \sum_{N_s \in \mathbf{V}_\mathbf{S}} \mathrm{x}_V(N_v, N_s)$$

Link mapping: $\forall \mathrm{R} \in \mathcal{R}.\forall L_v = (N_v^+, N_v^-) \in \mathbf{E}_\mathrm{R}.\forall N_s \in \mathbf{V}_\mathbf{S}$

$$\sum_{L_s \in \delta^+(N_s)} \mathrm{x}_E(L_v, L_s) - \sum_{L_s \in \delta^-(N_s)} \mathrm{x}_E(L_v, L_s) = \mathrm{x}_V(N_v^-, N_s) - \mathrm{x}_V(N_v^+, N_s)$$

Macro $alloc_V(R, N_s)$: $\forall R \in \mathcal{R}.\forall N_s \in \mathbf{V}_\mathbf{S}$

$alloc_V(R, N_s) = \sum_{N_v \in \mathbf{V}_\mathrm{R}} \mathbf{c}_\mathrm{R}(N_v) \cdot \mathrm{x}_V(N_v, N_s)$

Macro $alloc_V(R, N_s)$: $\forall R \in \mathcal{R}.\forall L_s \in \mathbf{E}_\mathbf{S}$

$alloc_E(R, L_s) = \sum_{L_v \in \mathbf{E}_\mathrm{R}} \mathbf{c}_\mathrm{R}(L_v) \cdot \mathrm{x}_E(L_v, L_s)$

Applications

Data center

- e.g. MapReduce cycles through different phases, traffic only during 30-60% of execution [7]
- price incentives for customers and providers to allow for / harness temporal flexibility [5]

Wide area networks

- Google uses SDN in the WAN to connect data centers [6]
- scheduling of bandwidth-intensive synchronizations
  - is necessary to achieve good utilization and resource isolation
  - is enabled by central SDN control