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

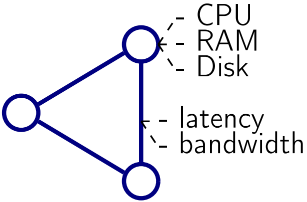
IEEE IPDPS 2014

Matthias Rost, Stefan Schmid, Anja Feldmann
Technische Universität Berlin

May 22th, 2014
Arizona State University

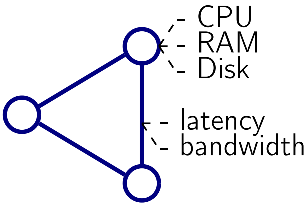
The Virtual Network Embedding Problem (VNEP)

Physical Network

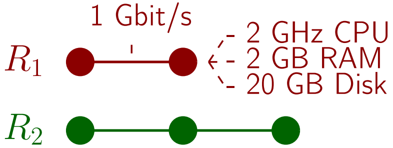


The Virtual Network Embedding Problem (VNEP)

Physical Network

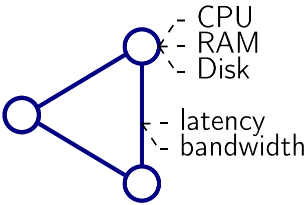


Virtual Network Requests

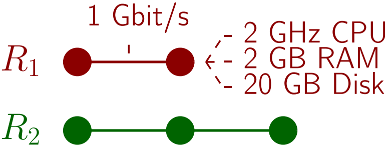


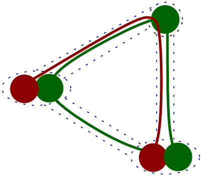
The Virtual Network Embedding Problem (VNEP)

Physical Network



Virtual Network Requests





Embedding

- 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

Related Work

TABLE III
TAXONOMY OF CONCISE VNE APPROACHES

Category	Reference	Optimization	Coordination	Contribution
C/S/C	[26] Inführ and Raidl (2011)	Exact	One Stage	Provides delay, location and routing constraints
	[59] Liu et al. (2011)	Exact	One Stage	Exact VNE based on correspondence matrices
	[60] Trinh et al. (2011)	Exact	One Stage	Exact VNE problem with SLA QoS guarantees
	[61] Pages et al. (2012)	Exact/Metaheuristic	One Stage	Introduces the VNE for optical networks
	[44] Lischka and Karl (2009)	Heuristic	One Stage	Provides one stage VNE. Based on SID
	[62] Di et al. (2010)	Heuristic	One Stage	Improvement of the approach in [44]
	[63] Ghazar and Samaan (2011)	Heuristic	One Stage	Introduces hierarchical management of the SN
	[64], [65] Yun et al. (2011-2012)	Heuristic	One Stage	First VNE approach in wireless multihop networks. Introduces metrics and feasibility measures for wireless VNE
	[56] Chen et al. (2012)	Heuristic	One Stage	Reduces resource fragmentation
	[66] Yu et al. (2012)	Heuristic	One Stage	One step VNE that increases coordination
	[67] Liu et al. (2011)	Heuristic	Two Stages	Improves coordination based on nodes proximity
	[24], [25] Sheng et al. (2011-2012)	Heuristic	Two Stages	Opportunistic resource sharing to deal with load fluctuation
	[68] Li et al. (2012)	Heuristic	Two Stages	Topology awareness to enforce VNE coordination
	[69] Lu and Turner (2006)	Heuristic	Uncoordinated	Embedding in specific backbone-star VN topologies
	[32] Yu et al.* (2008)	Heuristic	Uncoordinated	Utilizes the KSP algorithm [33] for VLiM
	[70] Razzaq and Siraj (2010)	Heuristic	Uncoordinated	Different K values in KSP based VLiM
	[71] Razzaq et al. (2011)	Heuristic	Uncoordinated	Investigates the VNE impact of bottlenecked nodes
	[72] Nogueira et al. (2011)	Heuristic	Uncoordinated	VNE considering SN resources heterogeneity
	[73] Leivadeas et al.* (2011)	Heuristic	Uncoordinated	Introduces VNE for wireless network testbeds
	[22], [57] Botero et al. (2011-2013)	Heuristic	Uncoordinated	Introduces hidden hop constraints
	[74] Zhu and Ammar* (2006)	Heuristic	Uncoordinated	Provides a balanced link and node stress in the SN
	[51] Fajjari et al. (2011)	Metaheuristic	One Stage	Max-Min Ant Colony metaheuristic VNE approach
	[75] Cheng et al. (2012)	Metaheuristic	One Stage	Accelerates convergence of PSO VNE metaheuristic with topology aware node ranking [37]
	[76] Zhang et al. (2012)	Heuristic	Uncoordinated	Maps one virtual node in several substrate nodes
	[77] Di et al. (2012)	Heuristic	One Stage	Coordinated VNE reducing the number of backtracks by carefully chosing the first virtual node to map
	[78] Abedifar and Eshghi (2012)	Heuristic	Uncoordinated	Introduces VNE in the optical domain trying to minimize the number of λs per link
	[79] Aris Leivadeas et al. (2012)	Heuristic	Coordinated	Considers importance of virtual nodes for embedding
	[80] Tae-Ho Lee et al. (2012)	Heuristic	InterInP	clustering of virtual networks in multi-provider environment
C/D/C	[19] Fajjari et al. (2011)	Heuristic	One Stage	Migration of nodes with bottlenecked adjacent links
	[81] Bienkowski et al. (2010)	Heuristic	Two Stages	Migration when service access position changes
	[74] Zhu and Ammar* (2006)	Heuristic	Uncoordinated	Reduce the cost of periodic reconfigurations
	[82] Fan and Ammar (2006)	Heuristic	Uncoordinated	Reduces the cost of VNRs reconfiguration
	[83] Cai et al. (2010)	Heuristic	Uncoordinated	Reconfiguration based on SN evolution
	[84] Shun-li and Xue-song (2011)	Heuristic	Uncoordinated	Identifies mapped virtual nodes and links with not optimal mapping and migrate them to save SN resources
	[85] Sun et al. (2012)	Heuristic	Uncoordinated	Introduces the VNE problem for evolving VNRs
D/S/C	[86], [87] Houidi et al. (2010)	Heuristic	Uncoordinated	First distributed approach to solve VNE. Proposes a VNE protocol to manage the communication among substrate nodes
	[88] Xin et al. (2011)	Heuristic	InterInP	Introduces the InterInP VNE for networked clouds
	[89] Lv et al. (2011)	Heuristic	InterInP	InterInP VNE using hierarchical virtual resource organization
	[42] Houidi et al.* (2011)	Exact/Metaheuristic	InterInP	VNR is split assigning each subVN in different InPs. Provides exact and heuristic splitting approaches
	[90] Leivadeas et al.* (2012)	Heuristic	InterInP	Graph partitioning InterInP VNE using a heuristic integrating a min k-cut algorithm followed by subgraph isomorphism
D/D/C	[91] Marquezan et al. (2010)	Heuristic	Uncoordinated	First distributed dynamic approach. Reorganizes the SN when VNs demands change

TABLE IV
TAXONOMY OF REDUNDANT VNE APPROACHES

Category	Reference	Optimization	Coordination	Contribution
C/S/R	[42] Houidi et al.* (2011)	Exact	One Stage	First approach providing an ILP exact solution
	[92] Zhang et al. (2011)	Exact	One Stage	Optimal resilient solution attaining an enhanced QoS mapping. Provides diversified substrate back-up paths
	[43] Botero et al. (2012)	Exact	One Stage	Introduces the energy aware VNE
	[93] Wang and Wolf (2011)	Exact	One Stage	Redefines the VNR as a traffic matrix
	[94], [95], [96] Shamsi and Brockmeyer (2007-2009)	Heuristic	One Stage	Recover link failures by providing backup paths with intermediate nodes
	[97] Koslovski et al. (2010)	Heuristic	One Stage	Introduces reliability as a service offered by the InP. Reliable VNEs based on subgraph isomorphism detection
	[98] Yu et al. (2010)	Heuristic	One Stage	Introduces failure-dependent protection with a back-up solution for each regional failure
	[99] Lv et al. (2012)	Heuristic	One Stage	Introduces looses to multicast VNE in wireless mesh networks
	[36], [27] Chowdhury et al. (2009-2011)	Heuristic	Two Stages	Coordination in VNE using multi-path for VLiM
	[31] Rahman et al. (2010)	Heuristic	Two Stages	Upon a failure, the economic penalty is minimized by the pre-reservation of a bandwidth quota for back-up in SN links
	[53] Butt et al.* (2010)	Heuristic	Two Stages	VNE awareness of the SN bottlenecked resources
	[29] Yeow et al. (2010)	Heuristic	Two Stages	Introduces sharing among back up resources. Reduces resources allocated for redundancy
	[100] Sun et al. (2011)	Heuristic	Two Stages	Resilient VNE optimizing the embedding cost and reducing computational complexity
	[30] Yu et al. (2011)	Heuristic	Two Stages	Resilient VNE analyzing failures in substrate nodes
	[32] Yu et al.* (2008)	Heuristic	Uncoordinated	Introduces the multi-path approach for VLiM
	[101] Gao et al. (2010)	Heuristic	Uncoordinated	Improvement of the approach in [36]
	[102] Yang et al. (2010)	Heuristic	Uncoordinated	Divides the SN in regions to reduce VNE complexity
	[103] Zho et al. (2010)	Heuristic	Uncoordinated	Maps one virtual node to multiple substrate nodes
	[104] Chen et al. (2010)	Heuristic	Uncoordinated	Reactive resiliency protection approach against failures during the online VNE process. Considers just substrate link failures
	[105] Yu et al. (2011)	Heuristic	Uncoordinated	Proactive VNE approach offering protection against SN link failures for links with high stress
	[23] Sun et al. (2011)	Heuristic	Uncoordinated	Introduces stochastic BW demand to the VNE
	[55] Lu et al. (2011)	Heuristic	Uncoordinated	Introduces load balancing in links
	[28] Guo et al. (2011)	Heuristic	Uncoordinated	Proactive resilient VLiM approach sharing back-up paths
	[37] Cheng et al. (2011)	Metaheuristic	Two Stages	Introduces topology-awareness in VNE
	[106] Sheng et al. (2011)	Metaheuristic	Two Stages	Embedding time depends on VNR lifetime. Uses simulated annealing metaheuristic
	[52] Zhang et al. (2012)	Metaheuristic	Two Stages	Introduces particle swarm optimizaton (PSO) metaheuristic
	[107] Sun et al. (2012)	Metaheuristic	Two Stages	Introduces VNE in multi-datacenter environments
	[108] Lv et al. (2012)	Metaheuristic	Uncoordinated	Introduces VNE in wireless mesh networks
	[90] Leivadeas et al.* (2012)	Heuristic	Two Stages	Uses the approach in [27] to solve the VNE for an arbitrary pool of heterogeneous resources
	[54] Masti and Raghavan (2012)	Heuristic	Two Stages	VNE considering the residual capacity of the substrate links
	[109] Zhang et al. (2012)	Exact/Heuristic	One Stage	Recover link failures providing disjoint SN backup paths
C/D/R	[53] Butt et al.* (2010)	Heuristic	Two Stages	Reactive reconfiguration of virtual links and nodes causing rejection to less critical SN regions
	[32] Yu et al.* (2010)	Heuristic	Uncoordinated	Reconfigure the embedding by changing the splitting ratio in the multipath VLiM solution
	[110] Schaffrath et al. (2010)	Exact	One Stage	ILP-based VNE. Dynamically reconfigures existing mappings
	[111] Chen et al. (2011)	Heuristic	Two Stages	Periodic reconfiguration of SN nodes with high utilization
D/S/R	[38] Chowdhury et al. (2010)	Heuristic	InterInP	First InterInP VNE proposal. Mediates between InP and SP interests. VNR is split across InPs and embedded locally
D/D/R	[112] Houidi et al. (2010)	Heuristic	Two Stages	Fault-tolerant VNE that acts upon node and link failures

Related Work

TABLE III
TAXONOMY OF CONCISE VNE APPROACHES

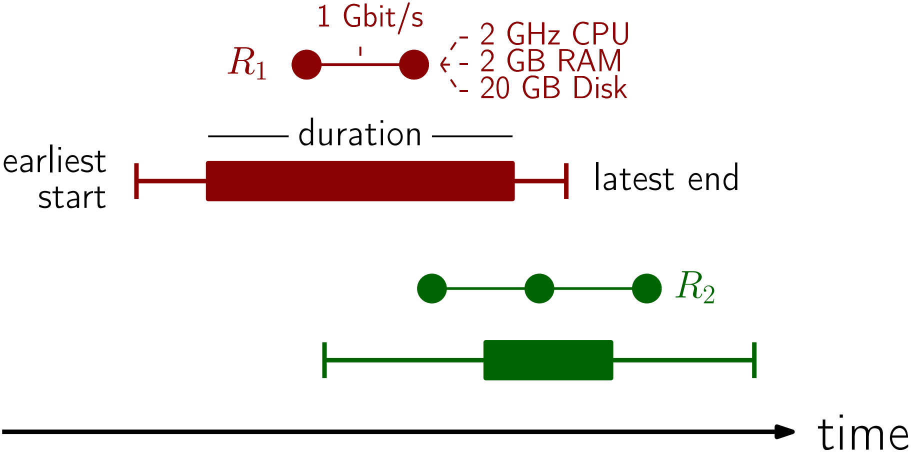
Category	Reference	Optimization	Coordination	Contribution
C/S/C	[26] Infuhr and Raidl (2011)	Exact	One Stage	Provides delay, location and routing constraints
	[59] Liu et al. (2011)	Exact	One Stage	Exact VNE based on correspondence matrices
	[60] Trinh et al. (2011)	Exact	One Stage	Exact VNE problem with SLA QoS guarantees
	[61] Pages et al. (2012)	Exact/Metaheuristic	One Stage	Introduces the VNE for optical networks
	[44] Lischka and Karl (2009)	Heuristic	One Stage	Provides one stage VNE. Based on SID
	[62] Di et al. (2010)	Heuristic	One Stage	Improvement of the approach in [44]
	[63] Ghazar and Samaan (2011)	Heuristic	One Stage	Introduces hierarchical management of the SN
	[64], [65] Yun et al. (2011-2012)	Heuristic	One Stage	First VNE approach in wireless multihop networks. Introduces metrics and feasibility measures for wireless VNE
	[56] Chen et al. (2012)	Heuristic	One Stage	Reduces resource fragmentation
	[66] Yu et al. (2012)	Heuristic	One Stage	One step VNE that increases coordination
	[67] Liu et al. (2011)	Heuristic	Two Stages	Improves coordination based on nodes proximity
	[24], [25] Sheng et al. (2011-2012)	Heuristic	Two Stages	Opportunistic resource sharing to deal with load fluctuation
	[68] Li et al. (2012)	Heuristic	Two Stages	Topology awareness to enforce VNE coordination
	[69] Lu and Turner (2006)	Heuristic	Uncoordinated	Embedding in specific backbone-star VN topologies
	[32] Yu et al.* (2008)	Heuristic	Uncoordinated	Utilizes the KSP algorithm [33] for VLiM
	[70] Razzaq and Siraj (2010)	Heuristic	Uncoordinated	Different K values in KSP based VLiM
	[71] Razzaq et al. (2011)	Heuristic	Uncoordinated	Investigates the VNE impact of bottlenecked nodes
	[72] Nogueira et al. (2011)	Heuristic	Uncoordinated	VNE considering SN resources heterogeneity
	[73] Leivadeas et al.* (2011)	Heuristic	Uncoordinated	Introduces VNE for wireless network testbeds
	[22], [illegible] Botero et al. (2011-2013)	Heuristic	Uncoordinated	Introduces hidden hop co[illegible]
	[7[illegible]] Zhu and [illegible]mmar* (2006)	Heuristic	Uncoordinated	Provides [illegible] balanced link and n[illegible] stress in the SN
	[illegible] Fajjari et [illegible] (20[illegible]1)	Metaheuri[illegible]	O[illegible] Stag[illegible]	M[illegible] Min[illegible] Ant Co[illegible]y me[illegible]heu[illegible]tic V[illegible]E a[illegible]roac[illegible]
	[7[illegible]] Cheng [illegible] (20[illegible]2)	Metaheuris[illegible]	O[illegible] Sta[illegible]	Acc[illegible]lerat[illegible] convergence [illegible] P[illegible] VN[illegible] m[illegible]heu[illegible]tic [illegible]th topology aware node ranking [37]
	[76] Zhang et al. (2012)	Heuristic	Uncoordinated	Maps one virtual node in several substrate nodes
	[77] Di et al. (2012)	Heuristic	One Stage	Coordinated VNE reducing the number of backtracks by carefully chosing the first virtual node to map
	[78] Abedifar and Eshghi (2012)	Heuristic	Uncoordinated	Introduces VNE in the optical domain trying to minimize the number of λs per link
	[79] Aris Leivadeas et al. (2012)	Heuristic	Coordinated	Considers importance of virtual nodes for embedding
	[80] Tae-Ho Lee et al. (2012)	Heuristic	InterInP	clustering of virtual networks in multi-provider environment
C/D/C	[19] Fajjari et al. (2011)	Heuristic	One Stage	Migration of nodes with bottlenecked adjacent links
	[81] Bienkowski et al. (2010)	Heuristic	Two Stages	Migration when service access position changes
	[74] Zhu and Ammar* (2006)	Heuristic	Uncoordinated	Reduce the cost of periodic reconfigurations
	[82] Fan and Ammar (2006)	Heuristic	Uncoordinated	Reduces the cost of VNRs reconfiguration
	[83] Cai et al. (2010)	Heuristic	Uncoordinated	Reconfiguration based on SN evolution
	[84] Shun-li and Xue-song (2011)	Heuristic	Uncoordinated	Identifies mapped virtual nodes and links with not optimal mapping and migrate them to save SN resources
	[85] Sun et al. (2012)	Heuristic	Uncoordinated	Introduces the VNE problem for evolving VNRs
D/S/C	[86], [87] Houidi et al. (2010)	Heuristic	Uncoordinated	First distributed approach to solve VNE. Proposes a VNE protocol to manage the communication among substrate nodes
	[88] Xin et al. (2011)	Heuristic	InterInP	Introduces the InterInP VNE for networked clouds
	[89] Lv et al. (2011)	Heuristic	InterInP	InterInP VNE using hierarchical virtual resource organization
	[42] Houidi et al.* (2011)	Exact/Metaheuristic	InterInP	VNR is split assigning each subVN in different InPs. Provides exact and heuristic splitting approaches
	[90] Leivadeas et al.* (2012)	Heuristic	InterInP	Graph partitioning InterInP VNE using a heuristic integrating a min k-cut algorithm followed by subgraph isomorphism
D/D/C	[91] Marquezan et al. (2010)	Heuristic	Uncoordinated	First distributed dynamic approach. Reorganizes the SN when VNs demands change

TABLE IV
TAXONOMY OF REDUNDANT VNE APPROACHES

Category	Reference	Optimization	Coordination	Contribution
C/S/R	[42] Houidi et al.* (2011)	Exact	One Stage	First approach providing an ILP exact solution
	[92] Zhang et al. (2011)	Exact	One Stage	Optimal resilient solution attaining an enhanced QoS mapping. Provides diversified substrate back-up paths
	[43] Botero et al. (2012)	Exact	One Stage	Introduces the energy aware VNE
	[93] Wang and Wolf (2011)	Exact	One Stage	Redefines the VNR as a traffic matrix
	[94], [95], [96] Shamsi and Brockmeyer (2007-2009)	Heuristic	One Stage	Recover link failures by providing backup paths with intermediate nodes
	[97] Koslowski et al. (2010)	Heuristic	One Stage	Introduces reliabilty as a service offered by the InP. Reliable VNEs based on subgraph isomorphism detection
	[98] Yu et al. (2010)	Heuristic	One Stage	Introduces failure-dependent protection with a back-up solution for each regional failure
	[99] Lv et al. (2012)	Heuristic	One Stage	Introduces looses to multicast VNE in wireless mesh networks
	[36], [27] Chowdhury et al. (2009-2011)	Heuristic	Two Stages	Coordination in VNE using multi-path for VLiM
	[31] Rahman et al. (2010)	Heuristic	Two Stages	Upon a failure, the economic penalty is minimized by the pre-reservation of a bandwidth quota for back-up in SN links
	[53] Butt et al.* (2010)	Heuristic	Two Stages	VNE awareness of the SN bottlenecked resources
	[29] Yeow et al. (2010)	Heuristic	Two Stages	Introduces sharing among back up resources. Reduces resources allocated for redundancy
	[100] Sun et al. (2011)	Heuristic	Two Stages	Resilient VNE optimizing the embedding cost and reducing computational complexity
	[30] Yu et al. (2011)	Heuristic	Two Stages	Resilient VNE analyzing failures in substrate nodes
	[32] Yu et al.* (2008)	Heuristic	Uncoordinated	Introduces the multi-path approach for VLiM
	[101] Gao et al. ([illegible]10)	Heuristic	Uncoordinated	Improvement of the [illegible]pproach in [16]
	[102] Yang et al. [illegible]010)	[illegible]	[illegible]	[illegible] the S[illegible] to re[illegible] V[illegible]E complexity
	[103] Zho et al. ([illegible]10)	[illegible]	U[illegible]coordinated	[illegible] o[illegible] node to [illegible] substrate [illegible]odes
	[104] Chen et al. ([illegible]010)	Heuristic	Uncoordinated	Reactive resiliency protection approach ag[illegible]st failures during the online VNE process. Considers just [illegible]strate link failures
	[105] Yu et al. (2011)	Heuristic	Uncoordinated	Proactive VNE approach offering protection against SN link failures for links with high stress
	[23] Sun et al. (2011)	Heuristic	Uncoordinated	Introduces stochastic BW demand to the VNE
	[55] Lu et al. (2011)	Heuristic	Uncoordinated	Introduces load balancing in links
	[28] Guo et al. (2011)	Heuristic	Uncoordinated	Proactive resilient VLiM approach sharing back-up paths
	[37] Cheng et al. (2011)	Metaheuristic	Two Stages	Introduces topology-awareness in VNE
	[106] Sheng et al. (2011)	Metaheuristic	Two Stages	Embedding time depends on VNR lifetime. Uses simulated annealing metaheuristic
	[52] Zhang et al. (2012)	Metaheuristic	Two Stages	Introduces particle swarm optimizaton (PSO) metaheuristic
	[107] Sun et al. (2012)	Metaheuristic	Two Stages	Introduces VNE in multi-datacenter environments
	[108] Lv et al. (2012)	Metaheuristic	Uncoordinated	Introduces VNE in wireless mesh networks
	[90] Leivadeas et al.* (2012)	Heuristic	Two Stages	Uses the approach in [27] to solve the VNE for an arbitrary pool of heterogeneous resources
	[54] Masti and Raghavan (2012)	Heuristic	Two Stages	VNE considering the residual capacity of the substrate links
	[109] Zhang et al. (2012)	Exact/Heuristic	One Stage	Recover link failures providing disjoint SN backup paths
C/D/R	[53] Butt et al.* (2010)	Heuristic	Two Stages	Reactive reconfiguration of virtual links and nodes causing rejection to less critical SN regions
	[32] Yu et al.* (2010)	Heuristic	Uncoordinated	Reconfigure the embedding by changing the splitting ratio in the multipath VLiM solution
	[110] Schaffrath et al. (2010)	Exact	One Stage	ILP-based VNE. Dynamically reconfigures existing mappings
	[111] Chen et al. (2011)	Heuristic	Two Stages	Periodic reconfiguration of SN nodes with high utilization
D/S/R	[38] Chowdhury et al. (2010)	Heuristic	InterInP	First InterInP VNE proposal. Mediates between InP and SP interests. VNR is split across InPs and embedded locally
D/D/R	[112] Houidi et al. (2010)	Heuristic	Two Stages	Fault-tolerant VNE that acts upon node and link failures

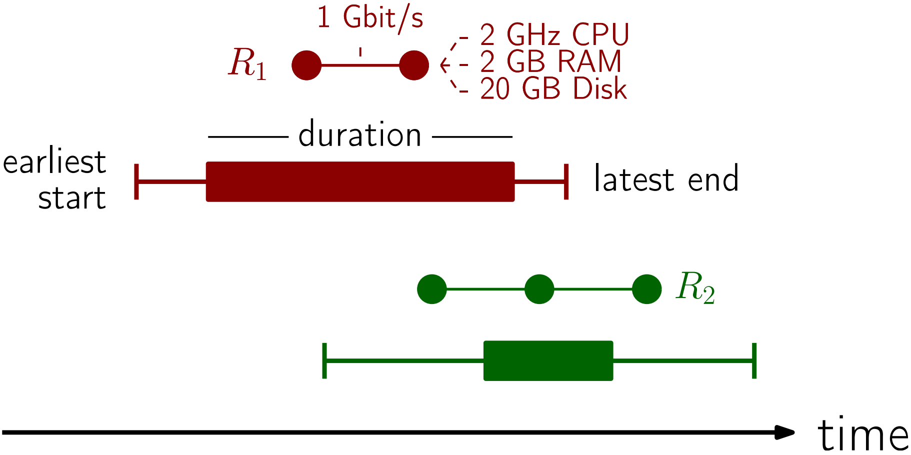
Our Contribution: Temporality

Our Model


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

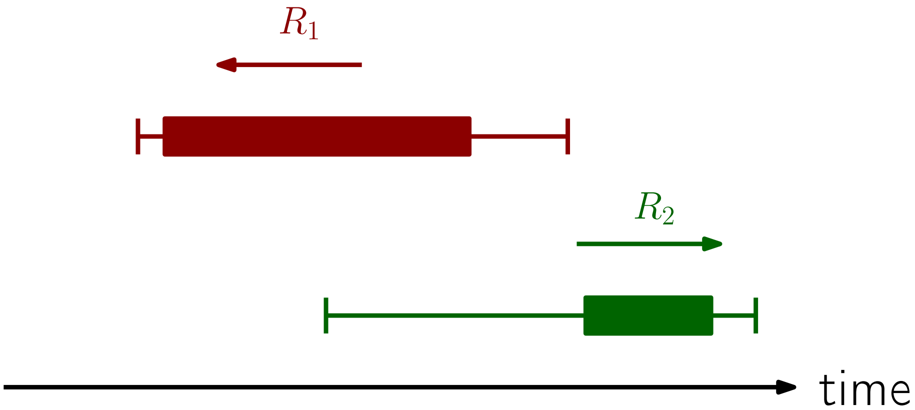
Our Model

Offline scenario

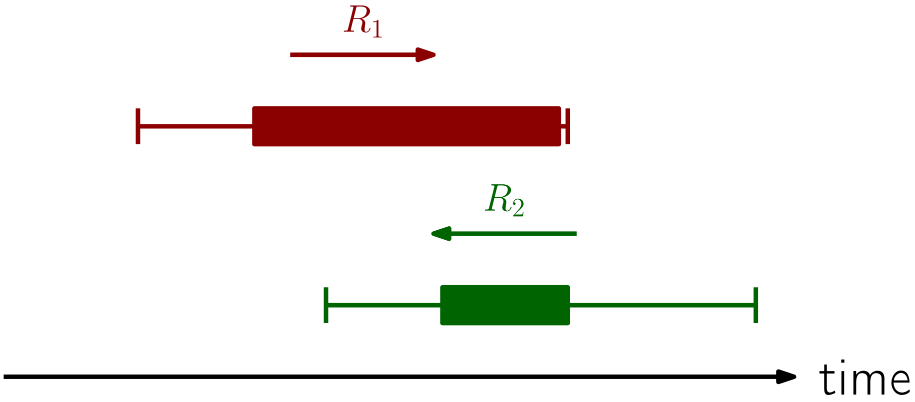


Motivation #1: Business

Provider Incentives: Minimizing Load

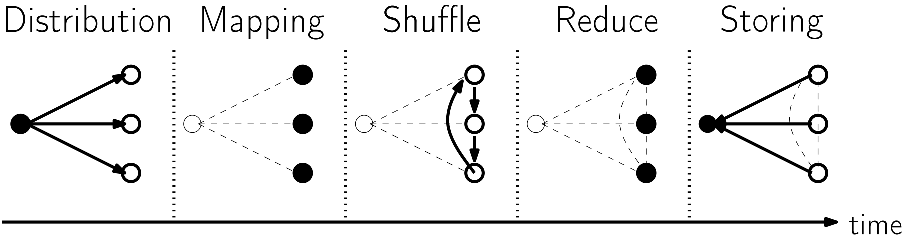


Provider Incentives: Maximizing Utilization by Collocation



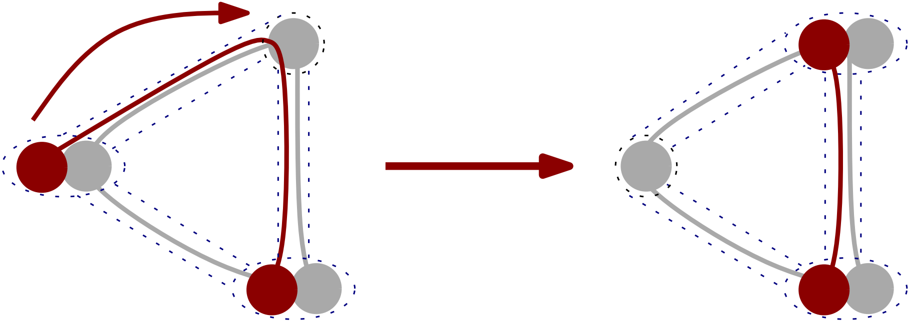
Motivation #2: Modeling Opportunities

Modeling Opportunities: Evolution of VNets

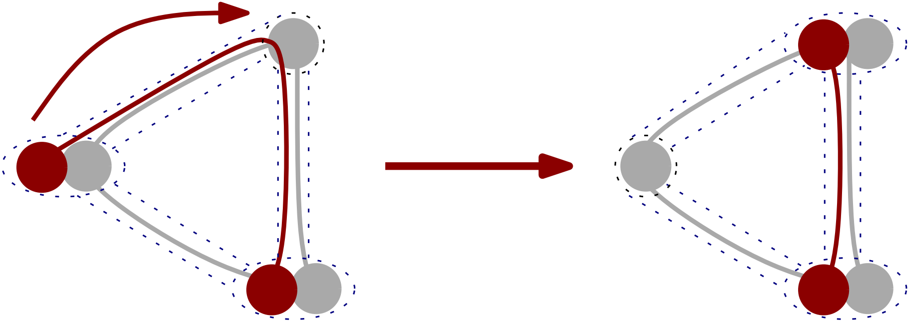


Reservation of maximal allocations over the whole time?

Modeling Opportunities: Migrations

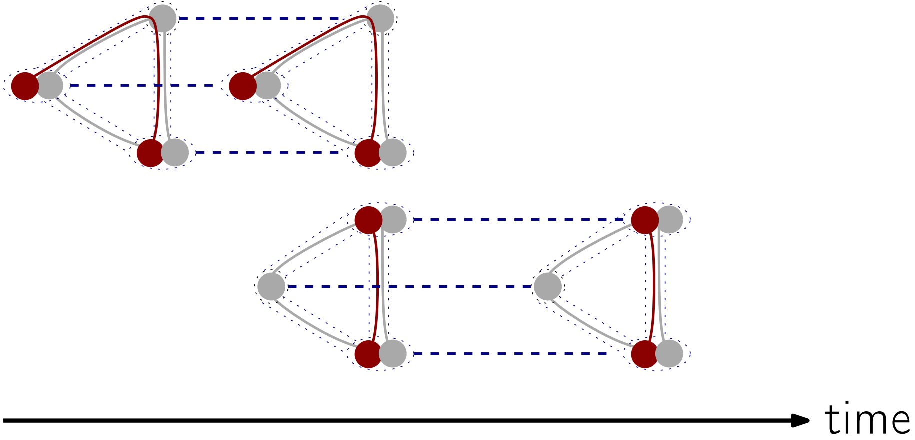


Modeling Opportunities: Migrations

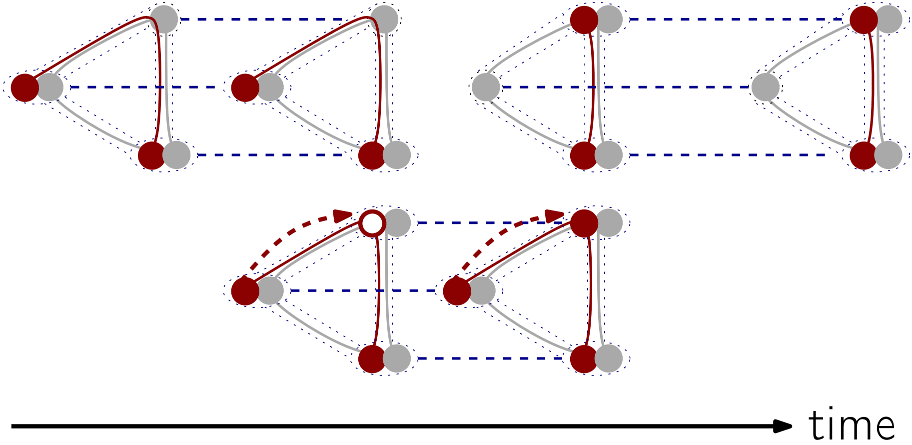


In previous work instantaneous operation!

Modeling Opportunities: Migrations

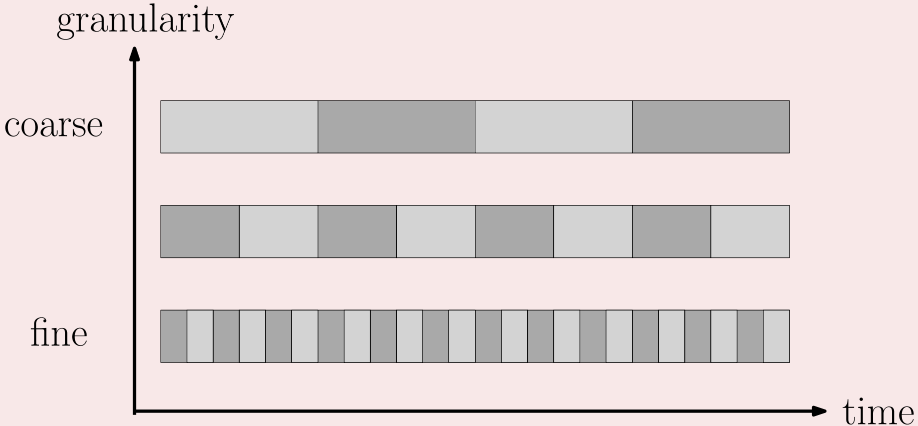


Modeling Opportunities: Fine-grained Migrations



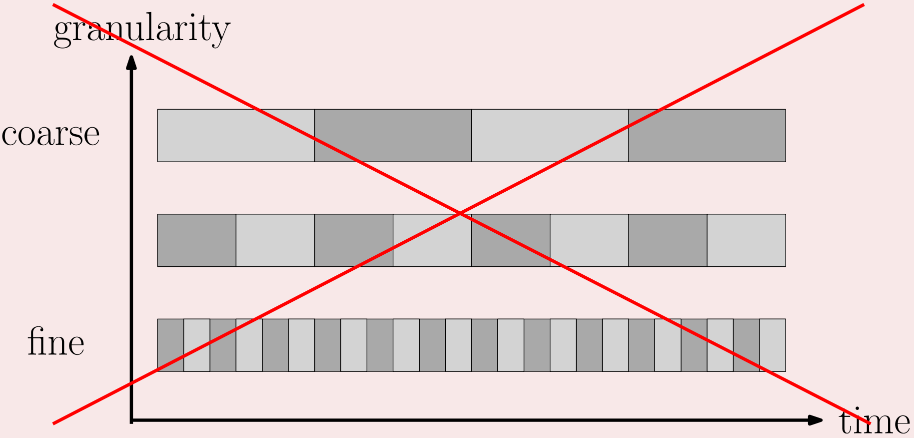
Important Decision: Continuous-Time Model!

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

Mapping process will be explained in a bit.

Classic VNEP Task

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

Overview

Overview

Contributions

1. Continuous-time Mixed-Integer Programming formulations for TVNEP
2. $c\Sigma$-Model utilizes state-space and symmetry reductions to render solving TVNEP (computationally) feasible
3. Greedy polynomial time heuristic which is based on $c\Sigma$-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 $t \in [0, \mathbf{T}]$:

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

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

Standard VNEP Access Control & Resource Mapping

Variables

Access Control $x_{\mathcal{R}} : \mathcal{R} \rightarrow \mathbb{B}$

Node Mapping $\forall R \in \mathcal{R}.\ x_V : \mathbf{V}_R \times \mathbf{V_S} \rightarrow \mathbb{B}$

Link Mapping $\forall R \in \mathcal{R}.\ x_E : \mathbf{E}_R \times \mathbf{E_S} \rightarrow [0, 1]$

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).

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

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

Standard VNEP Access Control & Resource Mapping

Variables

Access Control		$x_{\mathcal{R}} : \mathcal{R} \to \mathbb{B}$
Node Mapping	$\forall \mathrm{R} \in \mathcal{R}.$	$x_V : \mathbf{V}_\mathrm{R} \times \mathbf{V_S} \to \mathbb{B}$
Link Mapping	$\forall \mathrm{R} \in \mathcal{R}.$	$x_E : \mathbf{E}_\mathrm{R} \times \mathbf{E_S} \to [0,1]$

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

$$x_{\mathcal{R}}(\mathrm{R}) = \sum_{N_s \in \mathbf{V_S}} x_V(N_v, N_s)$$

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

$$\sum_{L_s \in \delta^+(N_s)} x_E(L_v, L_s) - \sum_{L_s \in \delta^-(N_s)} x_E(L_v, L_s) = x_V(N_v^-, N_s) - x_V(N_v^+, N_s)$$

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

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

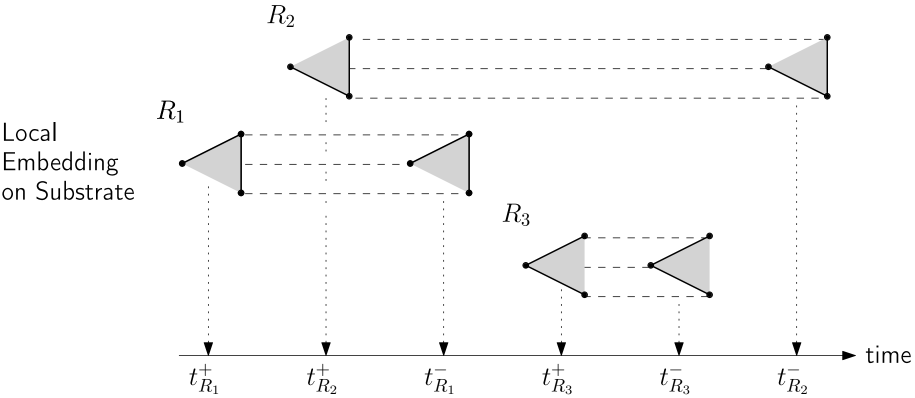
Macro $alloc_V(R, N_s)$: $\forall R \in \mathcal{R}. \forall L_s \in \mathbf{E_S}$

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

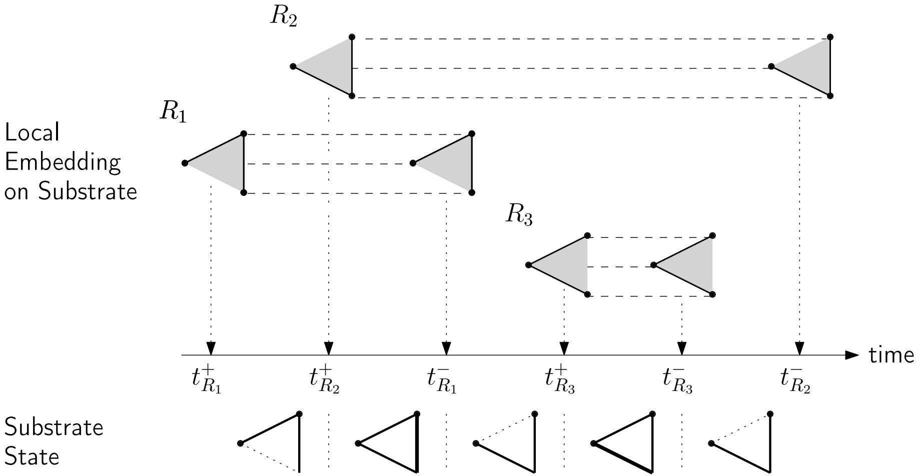
Modeling Continuous-Time: Checking Feasibility

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

Modeling Continuous-Time: Checking Feasibility

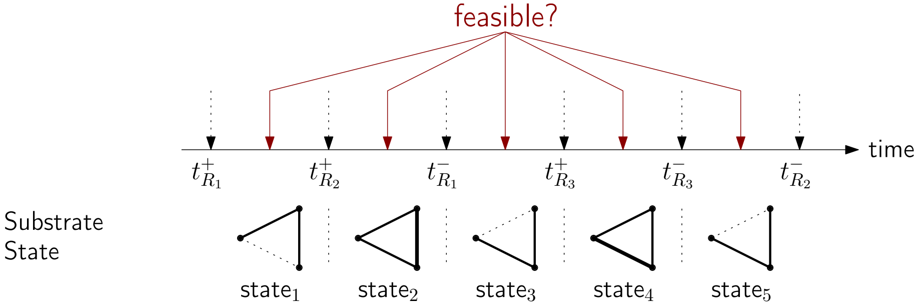


Modeling Continuous-Time: Checking Feasibility



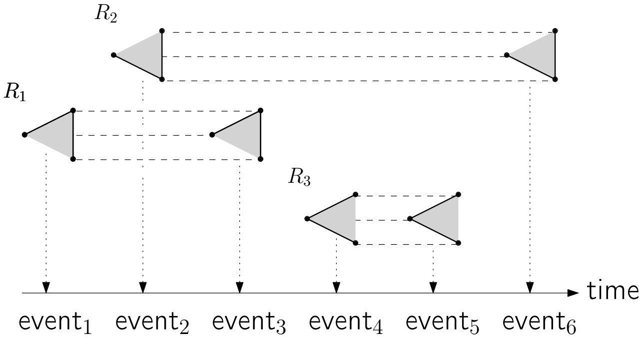
Modeling Continuous-Time: Checking Feasibility

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



Abstract Event Model

Modeling Continuous-Time: Abstract Event Model



Mapping Variables

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

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

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

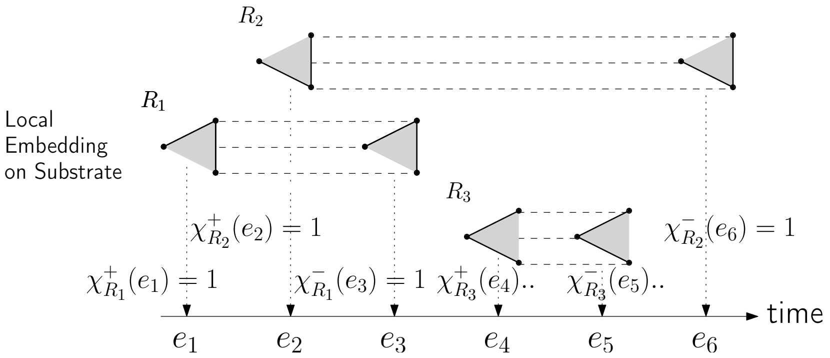
Bijective Mapping

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

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

Δ-Model

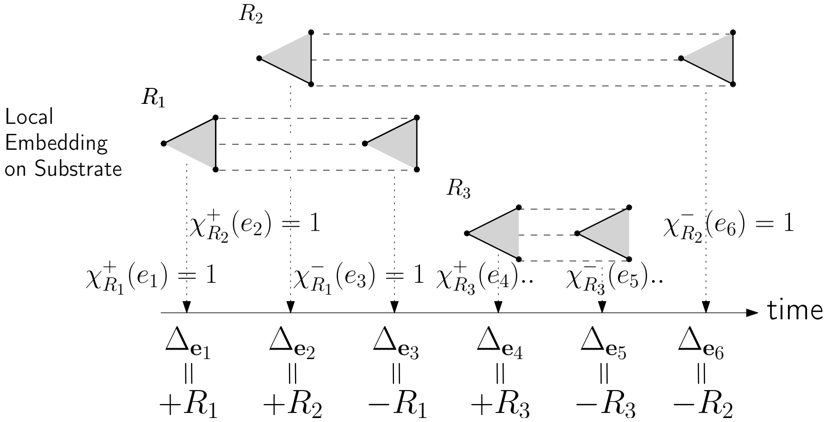
Reconstructing States: Δ-Model



Idea

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

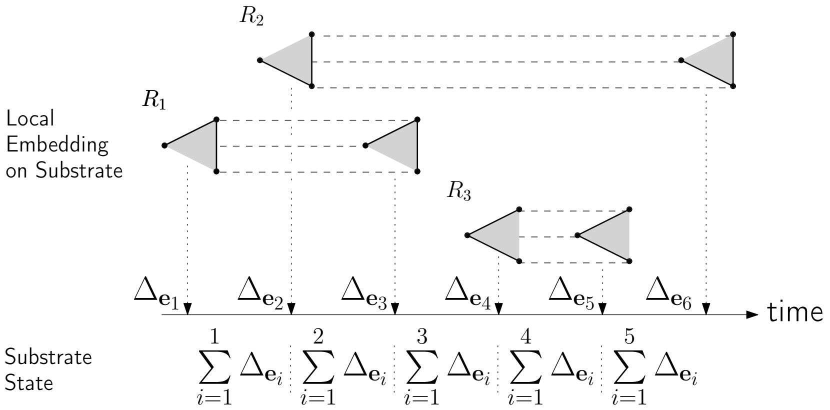
Reconstructing States: Δ-Model



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*: $\Delta_{\mathbf{e}_i} : \mathbf{V_S} \cup \mathbf{E_S} \rightarrow \mathbb{R}$ 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

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

$$\Delta_{\mathbf{e}_i}(N_s) = \begin{cases} +alloc_V(\mathsf{R}_1, N_s) & \text{, if } \chi^+_{\mathsf{R}_1}(\mathbf{e}_i) = 1 \\ -alloc_V(\mathsf{R}_1, N_s) & \text{, if } \chi^-_{\mathsf{R}_1}(\mathbf{e}_i) = 1 \\ \vdots & \\ +alloc_V(\mathsf{R}_\mathsf{k}, N_s) & \text{, if } \chi^+_{\mathsf{R}_\mathsf{k}}(\mathbf{e}_i) = 1 \\ -alloc_V(\mathsf{R}_\mathsf{k}, N_s) & \text{, if } \chi^-_{\mathsf{R}_\mathsf{k}}(\mathbf{e}_i) = 1 \end{cases}$$

Δ-Model: Computing State Changes

Conditional Assignment via Big-M Constraints

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

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

Δ-Model: Computing State Changes

Big-M Assignment of Start

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

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

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

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

$\Rightarrow$

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

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

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

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

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

$\Rightarrow$

equal

$$\Delta_{\mathbf{e}_i}(N_s) = \mathit{alloc}_V(\mathrm{R}, N_s)$$

Short Excursion: B&B

Branch-and-Bound

- branch-and-bound algorithms are in most cases used to solve MIPs
- *branching* generates subproblems (in a tree)
- subproblems can be cut off by *bounding* via computing LP relaxations
  - subproblem might be infeasible
  - subproblem might have worse objective value than best known solution

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

This is really bad!

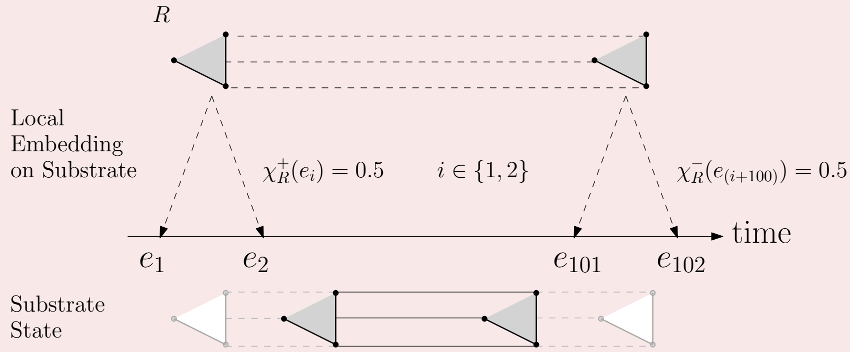
1. states do not 'materialize' well in LP relaxations: allocations will *never* be accounted for in the substrate's state
2. bounding is unable to reduce search space

Σ-Model

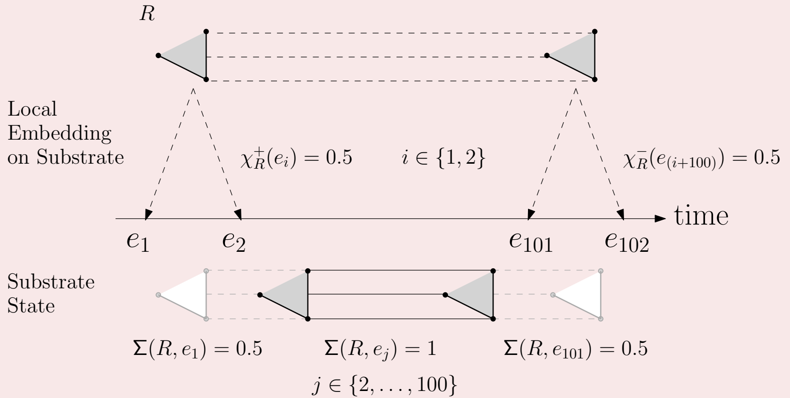
Σ-Model: Intuition

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: State Computation

Request allocations are computed for each state

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

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

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

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

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

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

Σ-Model: State Computation

Request allocations are computed for each state

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

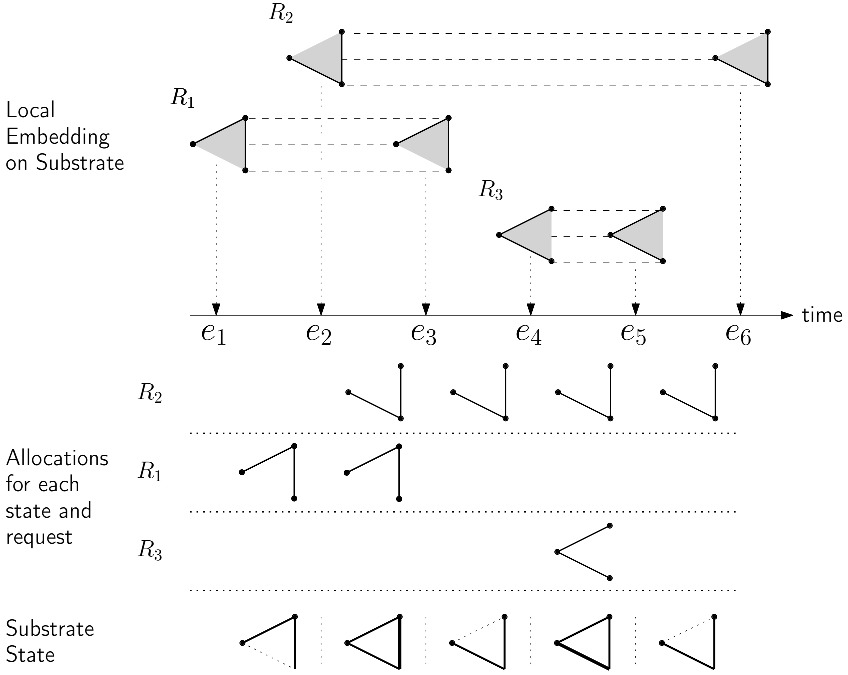
$$alloc_V(\mathrm{R}, \mathbf{s}_i, N_s) \geq alloc_V(\mathrm{R}, N_s) - \mathbf{c_S}(N_s) \cdot (1 - \Sigma(R, \mathbf{e}_i))$$

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

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

LP-Smearings

State allocations fully ‘materialize’ if $\Sigma(R, \mathbf{e}_i) = 1$.



R_2
R_1
Local Embedding on Substrate
R_3
time
e_1 e_2 e_3 e_4 e_5 e_6
R_2
Allocations for each state and request
R_1
R_3
Substrate State

cΣ-Model

cΣ-Model: Overview

Computational Trade-Off

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

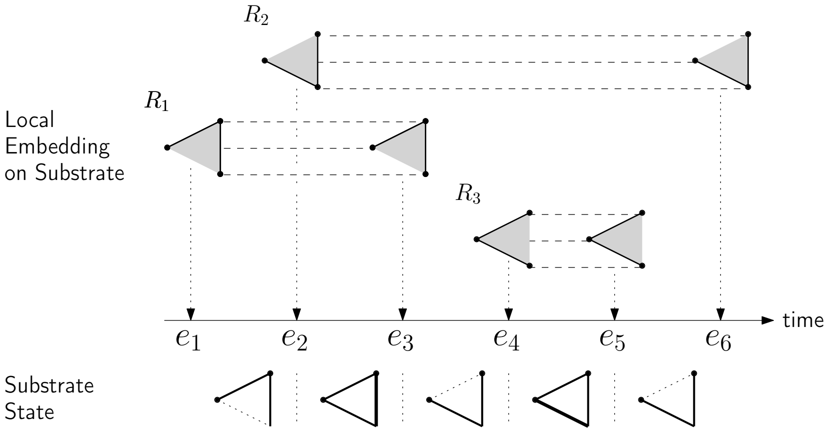
Σ-Model can be strengthened: cΣ-Model

Compactification Consider only *partial* event order. Yields *state-space* and *symmetry reductions*.

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

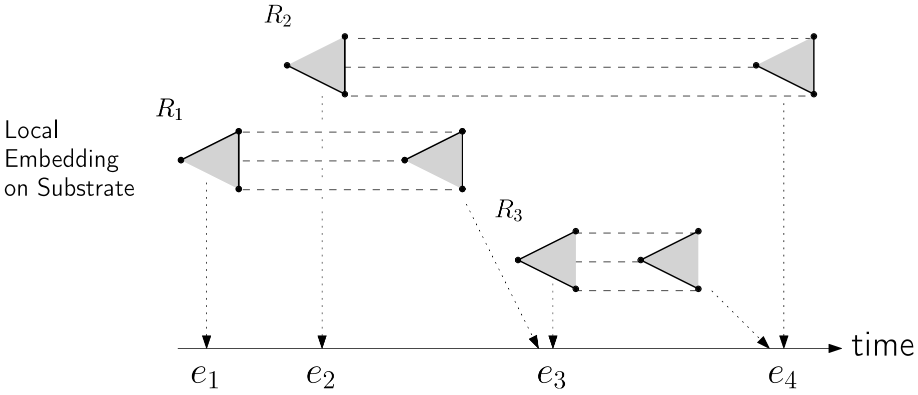
cΣ Optimization: State Compactification

cΣ-Model: State Compactification



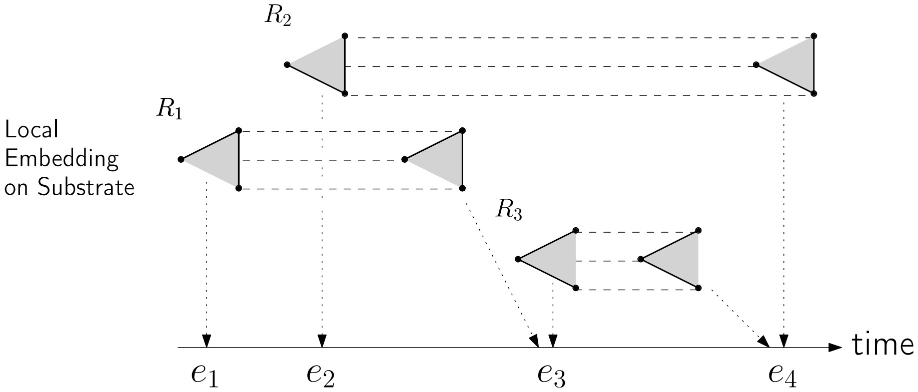
We only need to check feasibility after a request's start!

cΣ-Model: State Compactification



- consider only $|\mathcal{R}| + 1$ event points
- injective mapping of request starts onto first $|\mathcal{R}|$ event points
- mapping of request R's end onto event $\mathbf{e}_j$:
  R ends after $\mathbf{e}_{j-1}$ and before $\mathbf{e}_j$

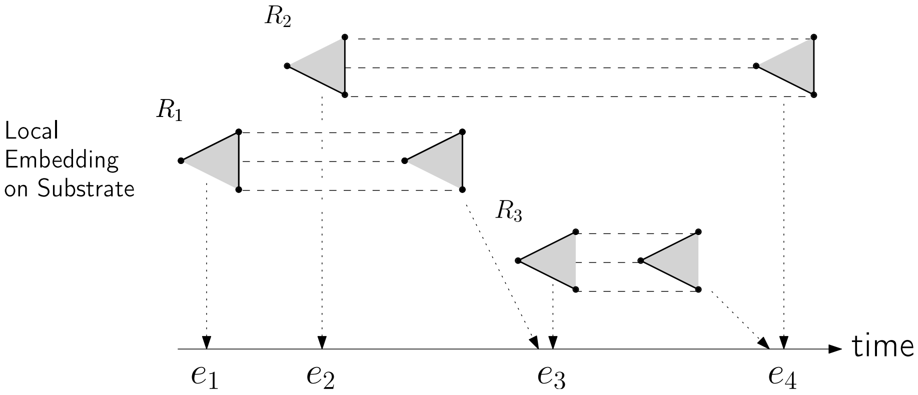
cΣ-Model: State Compactification



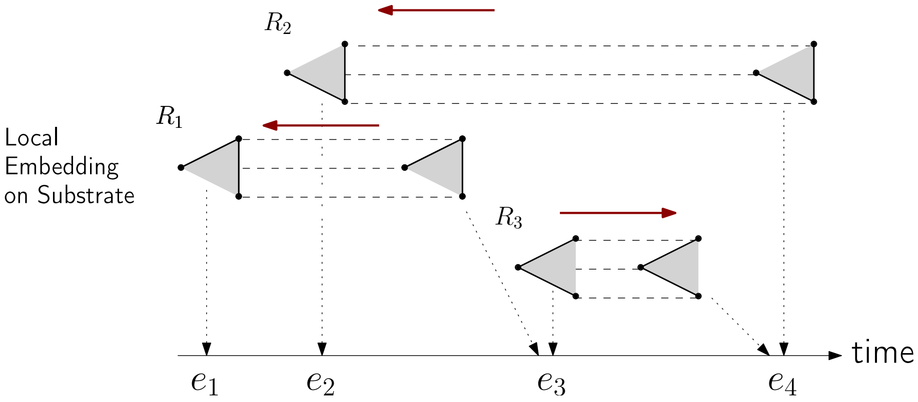
State-space reduction

Number of states is halved ⇒ number of variables is halved.

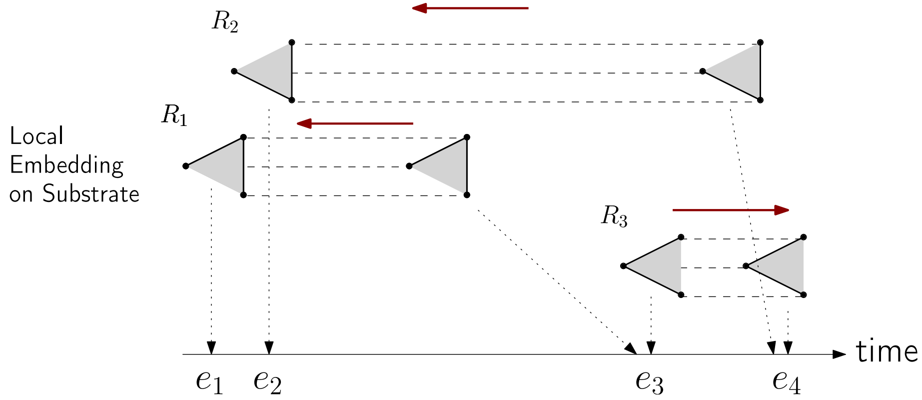
cΣ-Model: State Compactification is Symmetry Reduction



cΣ-Model: State Compactification is Symmetry Reduction

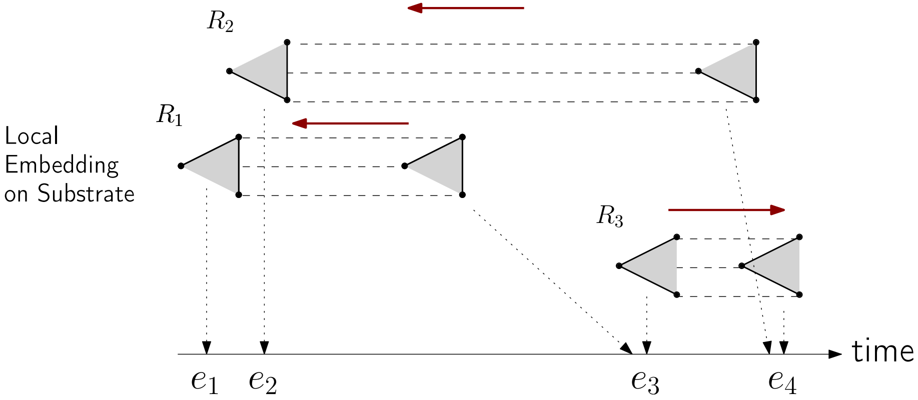


cΣ-Model: State Compactification is Symmetry Reduction



Same order as before!

cΣ-Model: State Compactification is Symmetry Reduction



Theorem

Compactification is symmetry reduction.

Intermezzo: Incorporating Time

cΣ-Model: Incorporating Time

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

$$\mathrm{t}_{\mathbf{e}_\mathrm{i}} \leq \mathrm{t}_{\mathbf{e}_{\mathrm{i}+1}}$$

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

$$\mathbf{d}_\mathrm{R} = t^-_\mathrm{R} - t^+_\mathrm{R}$$

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

$$t^+_\mathrm{R} \leq \mathrm{t}_{\mathbf{e}_\mathrm{i}} + (1 - \chi^+_\mathrm{R}(\mathbf{e}_i, \mathrm{R})) \cdot \mathbf{T}$$
$$t^+_\mathrm{R} \geq \mathrm{t}_{\mathbf{e}_\mathrm{i}} - (1 - \chi^+_\mathrm{R}(\mathbf{e}_i, \mathrm{R})) \cdot \mathbf{T}$$

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

$$t^-_\mathrm{R} \leq \mathrm{t}_{\mathbf{e}_\mathrm{i}} + (1 - \chi^-_\mathrm{R}(\mathbf{e}_i, \mathrm{R})) \cdot \mathbf{T}$$
$$t^-_\mathrm{R} \geq \mathrm{t}_{\mathbf{e}_{\mathrm{i}-1}} - (1 - \chi^-_\mathrm{R}(\mathbf{e}_i, \mathrm{R})) \cdot \mathbf{T}$$

cΣ-Model: Incorporating Time

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

$$\mathrm{t}_{\mathbf{e}_\mathrm{i}} \leq \mathrm{t}_{\mathbf{e}_{\mathrm{i}+1}}$$

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

$$\mathbf{d}_\mathsf{R} = t^-_\mathsf{R} - t^+_\mathsf{R}$$

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

$$t^+_\mathsf{R} \leq \mathrm{t}_{\mathbf{e}_\mathrm{i}} + \Big(1 - \sum_{j=1,\ldots,i} \chi^+_\mathsf{R}(\mathbf{e}_j, \mathsf{R})\Big) \cdot \mathbf{T}$$

$$t^+_\mathsf{R} \geq \mathrm{t}_{\mathbf{e}_\mathrm{i}} - \Big(1 - \sum_{j=i,\ldots,|\mathcal{E}|} \chi^+_\mathsf{R}(\mathbf{e}_j, \mathsf{R})\Big) \cdot \mathbf{T}$$

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

$$t^-_\mathsf{R} \leq \mathrm{t}_{\mathbf{e}_\mathrm{i}} + \Big(1 - \sum_{j=2,\ldots,i} \chi^-_\mathsf{R}(\mathbf{e}_j, \mathsf{R})\Big) \cdot \mathbf{T}$$

$$t^-_\mathsf{R} \geq \mathrm{t}_{\mathbf{e}_{\mathrm{i}-1}} - \Big(1 - \sum_{j=i,\ldots,|\mathcal{E}|} \chi^-_\mathsf{R}(\mathbf{e}_j, \mathsf{R})\Big) \cdot \mathbf{T}$$

cΣ-Model: Incorporating Time

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

$$\mathrm{t}_{\mathbf{e}_\mathrm{i}} \leq \mathrm{t}_{\mathbf{e}_{\mathrm{i}+1}}$$

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

$$\mathbf{d}_\mathsf{R} = t^-_\mathsf{R} - t^+_\mathsf{R}$$

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

$$t^+_\mathsf{R} \leq \mathrm{t}_{\mathbf{e}_\mathrm{i}} + (1 - \sum_{j=1,\ldots,i} \chi^+_\mathsf{R}(\mathbf{e}_j, \mathsf{R})) \cdot \mathbf{T}$$

$$t^+_\mathsf{R} \geq \mathrm{t}_{\mathbf{e}_\mathrm{i}} - (1 - \sum_{j=i,\ldots,|\mathcal{E}|} \chi^+_\mathsf{R}(\mathbf{e}_j, \mathsf{R})) \cdot \mathbf{T}$$

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

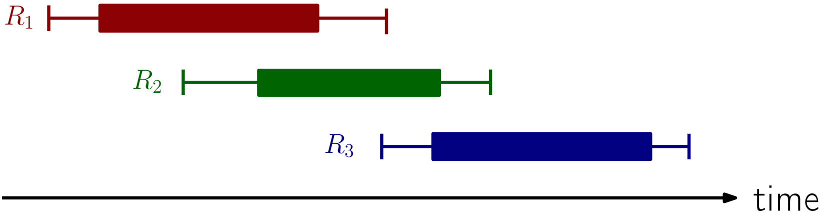
$$t^-_\mathsf{R} \leq \mathrm{t}_{\mathbf{e}_\mathrm{i}} + (1 - \sum_{j=2,\ldots,i} \chi^-_\mathsf{R}(\mathbf{e}_j, \mathsf{R})) \cdot \mathbf{T}$$

$$t^-_\mathsf{R} \geq \mathrm{t}_{\mathbf{e}_{\mathrm{i}-1}} - (1 - \sum_{j=i,\ldots,|\mathcal{E}|} \chi^-_\mathsf{R}(\mathbf{e}_j, \mathsf{R})) \cdot \mathbf{T}$$

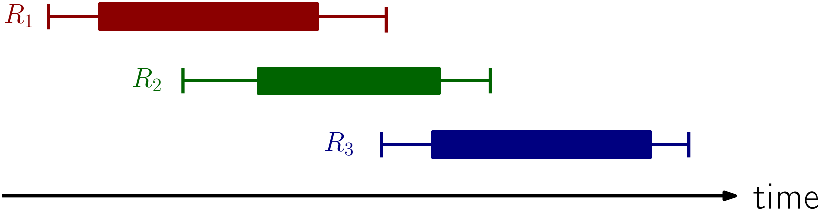
B
I
G
|
M

Optimizations: Temporal Dependency Graph User Cuts

Temporal Dependency Graph



Temporal Dependency Graph



Latest possible point in time for R_1 to start is less than the earliest point in time at which R_2 can start.

$\Rightarrow$ We know that R_1 must start before R_2.

Temporal Dependency Graph

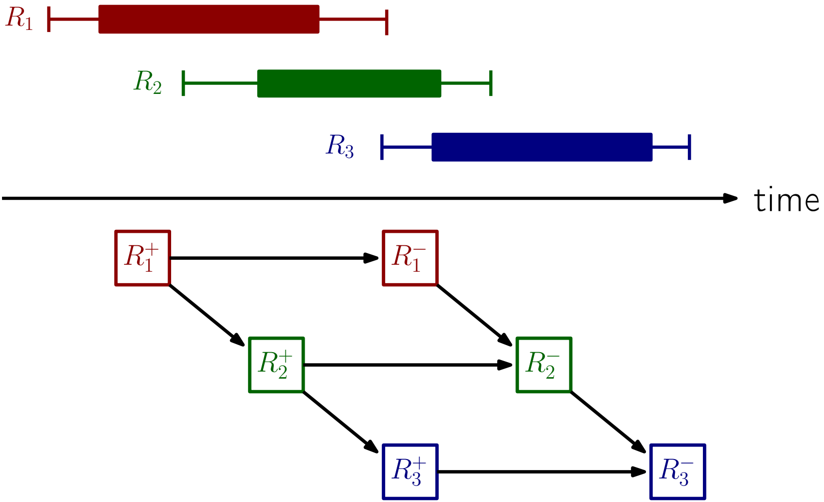


Figure: Temporal Dependency Graph

Temporal Dependency Graph (Formal)

Definition

- $G_{dep}(\mathcal{R}) = (V_{dep}, E_{dep})$
- $V_{dep} = \mathcal{R} \times \{start, end\}$
- $E_{dep} = \{(v, w) \in V_{dep}^2 | latest(v) < earliest(w)\}$

$$earliest((\mathrm{R}, t) \in V_{dep}) = \begin{cases} t_R^s & \text{, if } t = start \\ t_R^s + \mathbf{d}_\mathrm{R} & \text{, if } t = end \end{cases}$$

$$latest((\mathrm{R}, t) \in V_{dep}) = \begin{cases} t_R^e - \mathbf{d}_\mathrm{R} & \text{, if } t = start \\ t_R^e & \text{, if } t = end \end{cases}$$

Weighted Temporal Dependency Graph

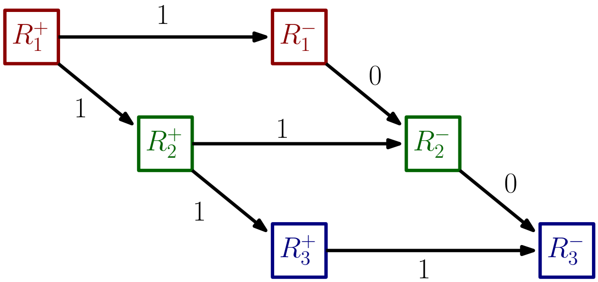


Figure: Temporal Dependency Graph with weights

By computing maximal distances (in polynomial time) 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

First Set of User Cuts (Valid Inequalities)

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

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

Macro χ_{Event}

$$\chi_{Event}(\mathbf{e}_i \in \mathcal{E}, (\mathrm{R}, t) \in V_{dep}) = \begin{cases} \chi^{+}_{\mathrm{R}}(\mathbf{e}_i) & \text{if } t = start \\ \chi^{-}_{\mathrm{R}}(\mathbf{e}_i) & \text{if } t = end \end{cases}$$

State-space reduction!

1. Effectively eliminates all mapping variables outside the interval $\{|dist^{+}_{\max}(v)| + 1, \ldots, |\mathcal{R}| + 1 - |dist^{-}_{\max}(v)|\}$

First Set of User Cuts (Valid Inequalities)

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

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

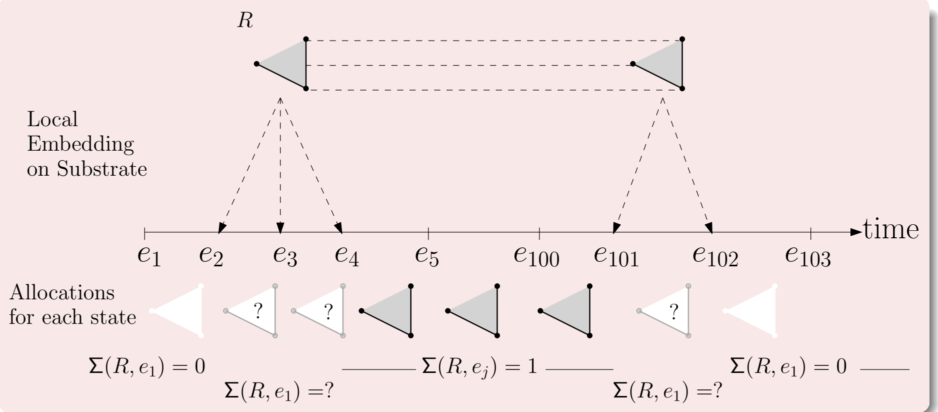
Macro χ_{Event}

$$\chi_{Event}(\mathbf{e}_i \in \mathcal{E}, (\mathrm{R}, t) \in V_{dep}) = \begin{cases} \chi^{+}_{\mathrm{R}}(\mathbf{e}_i) & \text{if } t = start \\ \chi^{-}_{\mathrm{R}}(\mathbf{e}_i) & \text{if } t = end \end{cases}$$

State-space reduction!

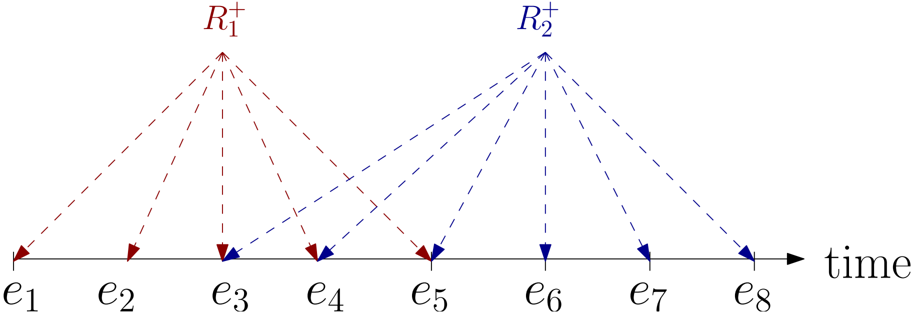
1. Effectively eliminates all mapping variables outside the interval $\{|dist^{+}_{\max}(v)| + 1, \ldots, |\mathcal{R}| + 1 - |dist^{-}_{\max}(v)|\}$ ...
2. and also state variables!

Elimination of State Variables



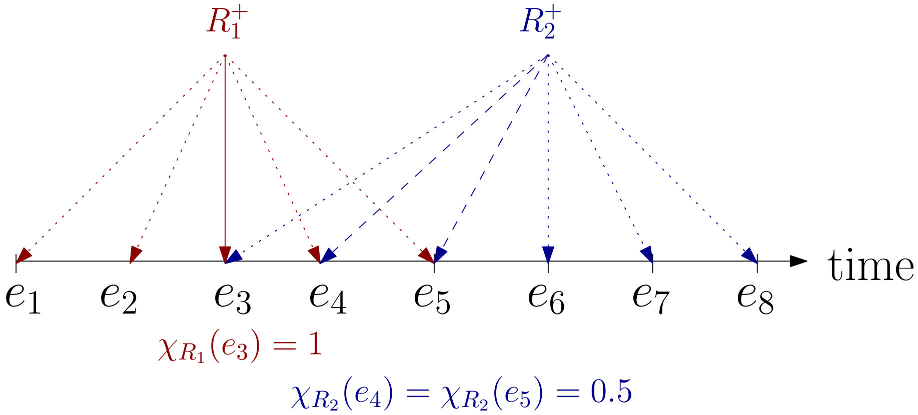
Second Set of User Cuts (Valid Inequalities)

$$dist^{-}_{\max}(R_1^+, R_2^+) = 2$$



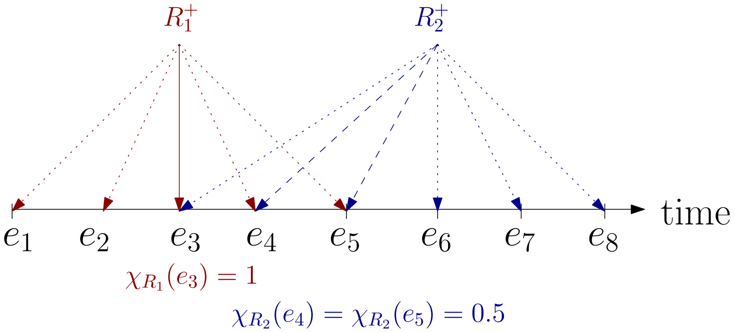
Second Set of User Cuts (Valid Inequalities)

$$dist^{-}_{\max}(R_1^+, R_2^+) = 2$$



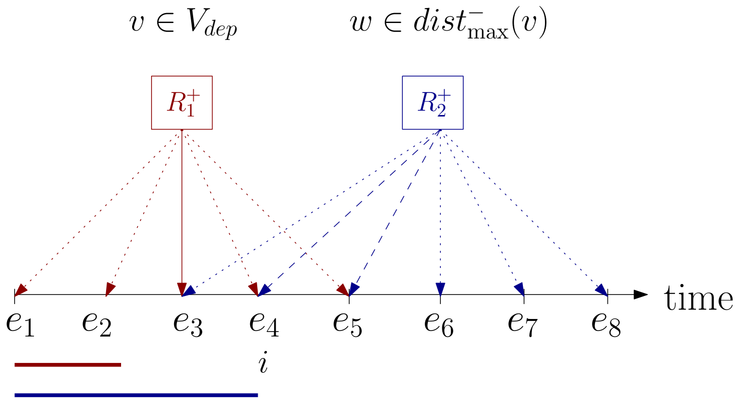
Second Set of User Cuts (Valid Inequalities)

$$dist^{-}_{\max}(R^{+}_1, R^{+}_2) = 2$$



Mapping $\chi^{+}_{R_2}(e_4) > 0$ should be forbidden!

Second Set of User Cuts (Valid Inequalities)



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

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

Temporal Dependency Graph User Cuts

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

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

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

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

Strengthen formulation!

Overview cΣ-Model

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

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

Access Control & Resource Mapping

Mapping onto Event Points

Variables

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

Mapping each start / end: $\forall \mathsf{R} \in \mathcal{R}$.

$$\sum_{\mathbf{e}_i \in \{\mathbf{e}_1, \ldots, \mathbf{e}_{|\mathcal{R}|}\}} \chi^{+}_{\mathsf{R}}(\mathbf{e}_i) = 1 \qquad \sum_{\mathbf{e}_i \in \{\mathbf{e}_2, \ldots, \mathbf{e}_{|\mathcal{R}|+1}\}} \chi^{-}_{\mathsf{R}}(\mathbf{e}_i) = 1$$

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

$$\sum_{\mathsf{R} \in \mathcal{R}} \left(\chi^{+}_{\mathsf{R}}(\mathbf{e}_i)\right) = 1$$

Guaranteeing State Feasibility

Variables

$alloc_V : \mathcal{R} \times \mathcal{S} \times \mathbf{V_S} \rightarrow \mathbb{R}_{\geq\nvdash}$ $\quad$ $alloc_E : \mathcal{R} \times \mathcal{S} \times \mathbf{E_S} \rightarrow \mathbb{R}_{\geq\nvdash}$

Computing allocations at states: $\forall \mathrm{R} \in \mathcal{R}. \forall \mathbf{s}_i \in \mathcal{S}. \forall N_s \in \mathbf{V_S} / \forall L_s \in \mathbf{E_S}.$

- $alloc_V(\mathrm{R}, \mathbf{s}_i, N_s) \geq alloc_V(\mathrm{R}, N_s) - \mathbf{c_S}(N_s) \cdot (1 - \Sigma(R, \mathbf{e}_i))$
- $alloc_E(\mathrm{R}, \mathbf{s}_i, L_s) \geq alloc_E(\mathrm{R}, L_s) - \mathbf{c_S}(L_s) \cdot (1 - \Sigma(R, \mathbf{e}_i))$

Ensuring feasibility: $\forall \mathbf{s}_i \in \mathcal{S}. \forall N_s \in \mathbf{V_S} / L_s \in \mathbf{E_S}.$

- $\mathbf{c_S}(N_s) \geq \sum_{\mathrm{R} \in \mathcal{R}} alloc_V(\mathrm{R}, \mathbf{s}_i, N_s)$
- $\mathbf{c_S}(L_s) \geq \sum_{\mathrm{R} \in \mathcal{R}} alloc_E(\mathrm{R}, \mathbf{s}_i, L_s)$

Access Control & Resource Mapping

Mapping onto Event Points

Guaranteeing State Feasibility

Guaranteeing Temporal Feasibility

Variables

$\forall \mathsf{R} \in \mathcal{R}. t_{\mathsf{R}}^{+}, t_{\mathsf{R}}^{-} \in \mathbb{R}_{\geq 0} \qquad \forall \mathbf{e}_i \in \mathcal{E}. t_{\mathbf{e}_i} \in \mathbb{R}_{\geq 0}$

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

$\mathbf{d}_{\mathsf{R}} = t_{\mathsf{R}}^{-} - t_{\mathsf{R}}^{+}$

Setting start times: $\forall \mathsf{R} \in \mathcal{R}. \forall \mathbf{e}_i \in \{\mathbf{e}_1, \ldots, \mathbf{e}_{|\mathcal{R}|}\}$.

$$t_{\mathsf{R}}^{+} \leq \mathrm{t}_{\mathbf{e}_i} + (1 - \sum_{j=1,\ldots,i} \chi_{\mathsf{R}}^{+}(\mathbf{e}_j, \mathsf{R})) \cdot \mathbf{T} \qquad t_{\mathsf{R}}^{+} \geq \mathrm{t}_{\mathbf{e}_i} - (1 - \sum_{j=i,\ldots,|\mathcal{E}|} \chi_{\mathsf{R}}^{+}(\mathbf{e}_j, \mathsf{R})) \cdot \mathbf{T}$$

Setting end times: $\forall \mathsf{R} \in \mathcal{R}. \forall \mathbf{e}_i \in \{\mathbf{e}_2, \ldots, \mathbf{e}_{|\mathcal{R}|+1}\}$.

$$t_{\mathsf{R}}^{-} \leq \mathrm{t}_{\mathbf{e}_i} + (1 - \sum_{j=2,\ldots,i} \chi_{\mathsf{R}}^{-}(\mathbf{e}_j, \mathsf{R})) \cdot \mathbf{T} \qquad t_{\mathsf{R}}^{-} \geq \mathrm{t}_{\mathbf{e}_{i-1}} - (1 - \sum_{j=i,\ldots,|\mathcal{E}|} \chi_{\mathsf{R}}^{-}(\mathbf{e}_j, \mathsf{R})) \cdot \mathbf{T}$$

Access Control & Resource Mapping

Mapping onto Event Points

Guaranteeing State Feasibility

Guaranteeing Temporal Feasibility

Temporal Dependency Graph User Cuts

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

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

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

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

Access Control & Resource Mapping

Mapping onto Event Points

Guaranteeing State Feasibility

Guaranteeing Temporal Feasibility

Temporal Dependency Graph User Cuts

Some further optimizations

- Big-M constants are chosen as *tight* as possible
- virtual links can be aggregated if their virtual source or their virtual destination is the same

Greedy Heuristic $c\Sigma_A^G$

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

There are maximally $|\mathcal{R}|$ many possible orderings to consider.

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.

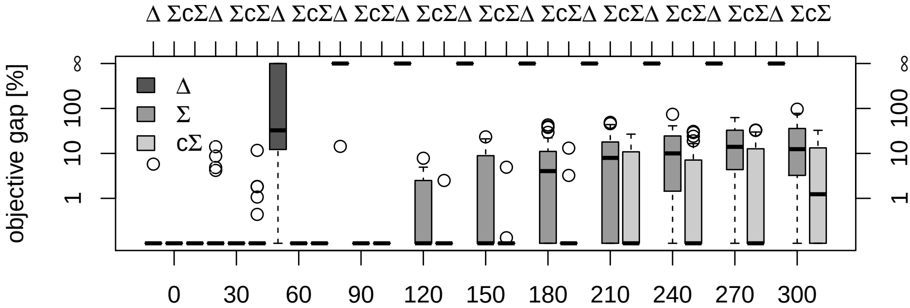
Computational Setup

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

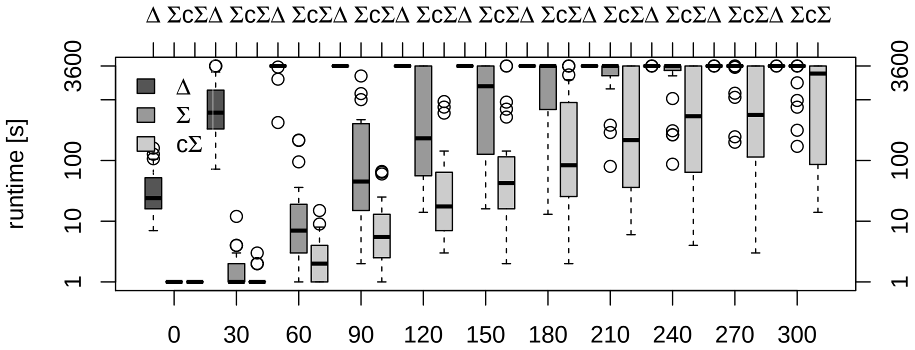
Task: Maximize revenue $\propto$ load $\cdot$ duration

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

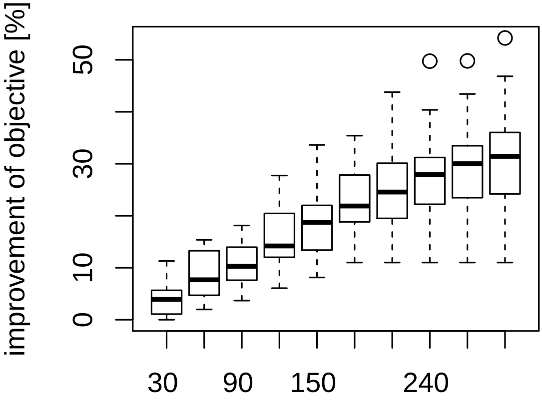
Objective Gap: MIP Formulations



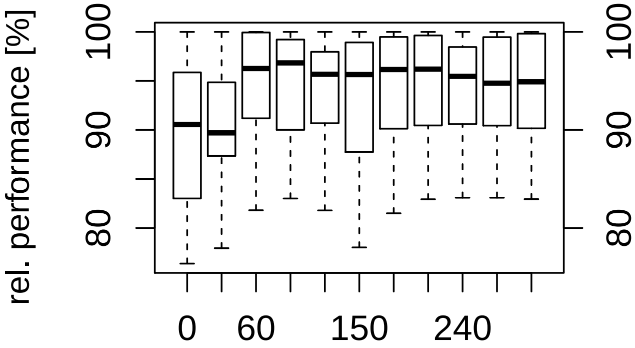
Runtime: MIP Formulations



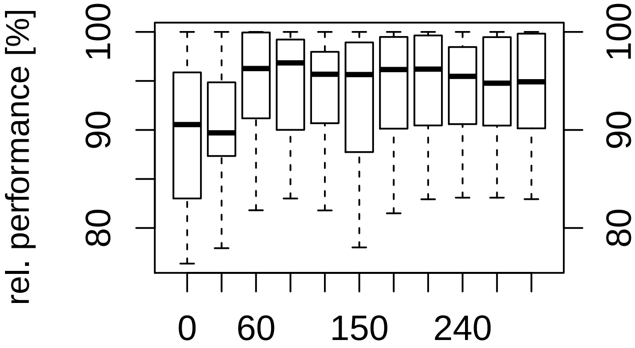
Benefit of Flexibility



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.

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