DATE 24
DESIGN, AUTOMATION & TEST IN EUROPE
25 – 27 March 2024 · Valencia, Spain
The European Event for Electronic System Design & Test

USC Viterbi School of Engineering

Efficient Exploration of Cyber-Physical System Architectures Using Contracts and Subgraph Isomorphism

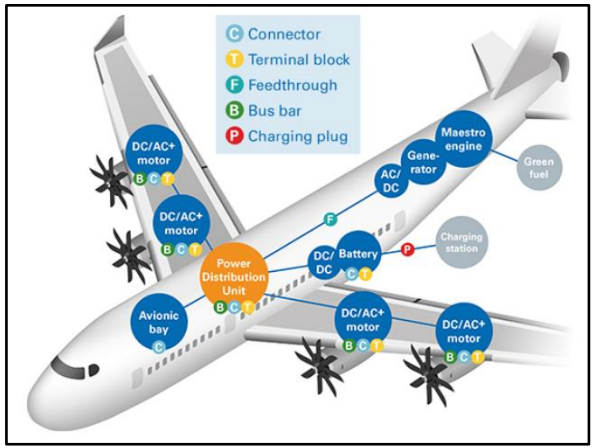
Yifeng Xiao[1], Chanwook Oh[1], Michele Lora[2], and Pierluigi Nuzzo[1]

[1]University of Southern California, Los Angeles, CA, US
[2]University of Verona, Italy

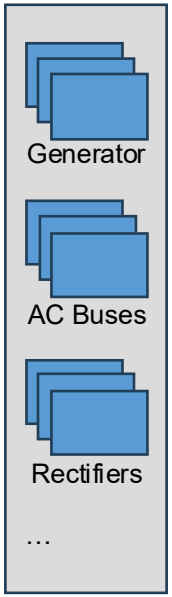
The Challenge: Cyber-Physical System (CPS) Architecture Exploration

Requirements

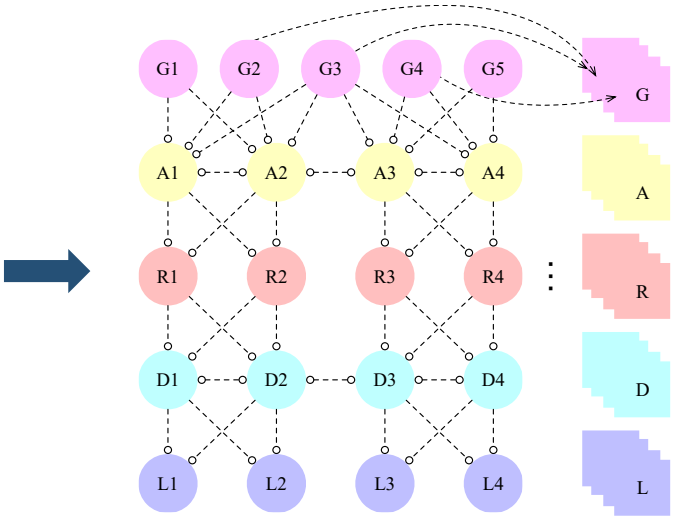
1. Timing
2. Energy Consumption
3. …



Architecture: Set of components and interconnections



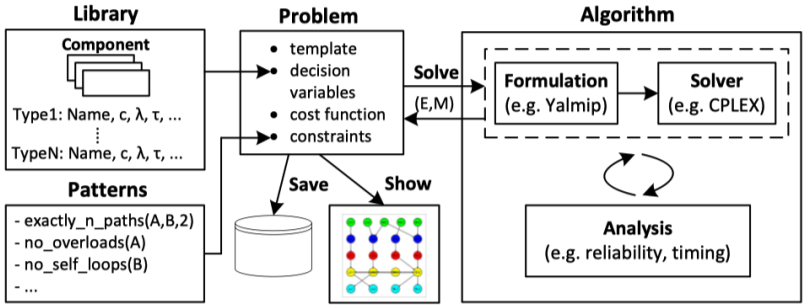
Library



Given a set of **components and connections**, **system requirements**, and an **implementation library,** find an **optimal architecture** that minimizes a cost function while satisfying all the requirements.

CPS Design Space Exploration: Existing Approaches

- Satisfiability modulo theories [S. Peter et al., 2015]
- Graph-based methods: e.g., based on ordered binary decision diagram [H. Neema et al., 2014]
- **Mixed integer linear programming (MILP)**
  - Flexible: Can express multiple heterogeneous requirements and cost objectives
  - ***ArchEx*** [D. Kirov et al., 2017]: Proposes efficient encodings and solutions strategies
  - Exponential complexity with the size of the architecture



Our Approach: Support novel **decomposition strategies** and search **space pruning methods** to reduce exploration costs and enhance scalability

Assume-Guarantee (A/G) Contracts Facilitate Compositional Reasoning

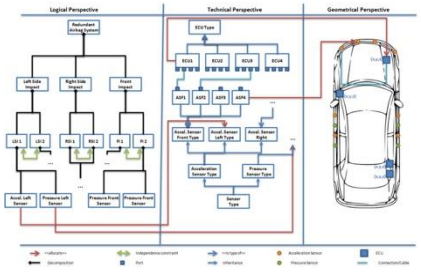
Assume-Guarantee Contracts

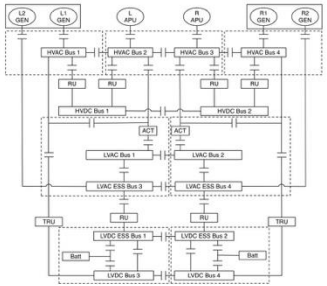
[A. Benveniste, et al., 2012]

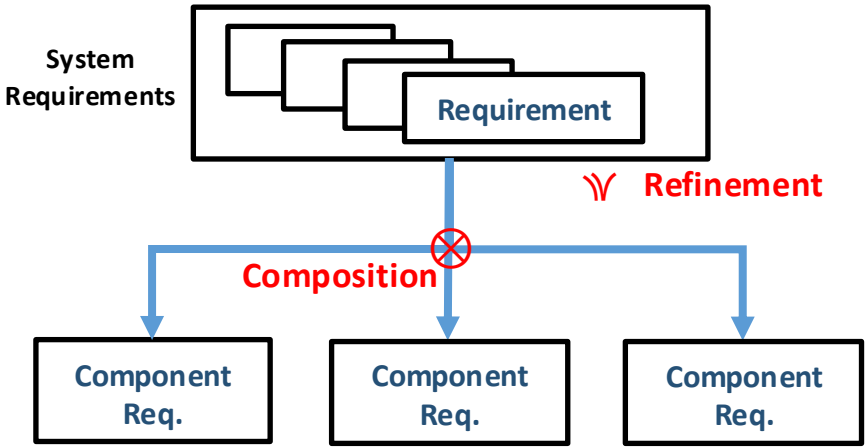
$$\boldsymbol{C} = (\boldsymbol{A}, \boldsymbol{G}) \qquad \boldsymbol{M} \models \boldsymbol{C}$$

Effectively applied to CPS design:

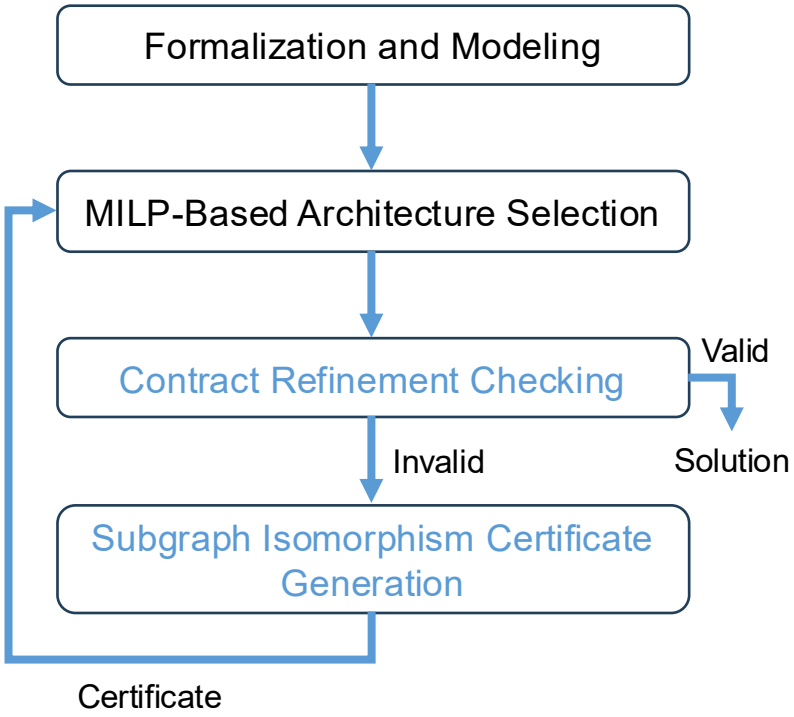
- Virtual integration testing and architecture design of a vehicle airbag system [Damm et al., 2011]
- Correct-by-construction aircraft electric power system design [Nuzzo et al., 2014]







ContrArc: Contract-Based Architecture Exploration



- **Contract-based modeling and decomposition** methods to enhance scalability.
- **Coordination** between MILP solving and **graph analysis** to generate infeasible certificates
- Novel certificate generation method combining **contract refinement checking** with **subgraph isomorphism** to exclude invalid results

Outline

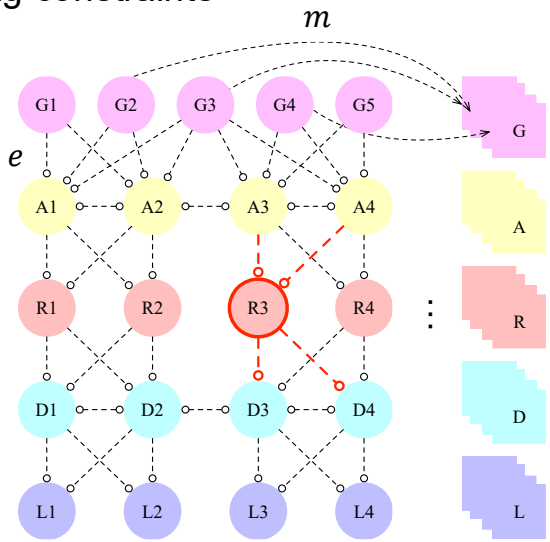
- Contract-Based Formalization and Modeling
- MILP-Based Architecture Selection
- Contract Refinement Checking
- Subgraph Isomorphism Certificate Generation
- Case Studies

Contract-Based Modeling and Formulation

Requirements:

1. **Connection and mapping constraints**
2. Flow constraints
3. Timing constraints

…



- *Assumptions:* if a component has connections, it can be mapped to an implementation in the library.
- *Guarantees:* if a component has input connections, it should have output connections.

$$C_{R3}^{C} = (\phi_{A_{R3}^{C}}, \phi_{G_{R3}^{C}})$$

$$\phi_{A_{R3}^{C}} := \left((e_{A3,R3} + e_{A4,R3} + e_{R3,D3} + e_{R3,D4}) \geq 1 \rightarrow \Sigma_i m_{R3,i} = 1\right) \wedge \left((e_{A3,R3} + e_{A4,R3} + e_{R3,D3} + e_{R3,D4}) = 0 \rightarrow \Sigma_i m_{R3,i} = 0\right)$$

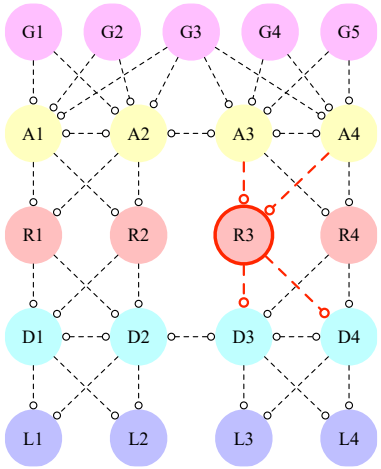
$$\phi_{G_{R3}^{C}} := \left((e_{A3,R3} + e_{A4,R3}) \geq 1 \rightarrow (e_{R3,D3} + e_{R3,D4}) \geq 1\right) \wedge \cdots$$

Formally Capture Requirements as Contracts

Requirements:

1. Connection and mapping constraints
2. **Flow constraints**
3. Timing constraints

…



Component-level:

- *Assumptions:* Input flow remains below the prescribed throughput.
- *Guarantees:* Input flow and output flow must be balanced.

$$C_{R3}^{F} = (\phi_{A_{R3}^{F}}, \phi_{G_{R3}^{F}})$$

$$\phi_{A_{R3}^{F}} := f_{R3}^{P} \geq e_{A3,R3} f_{A3,R3} + e_{A4,R3} f_{A4,R3} \geq f_{R3}^{C}$$

$$\phi_{G_{R3}^{F}} := e_{A3,R3} f_{A3,R3} + e_{A4,R3} f_{A4,R3} \geq e_{R3,D3} f_{R3,D3} + e_{R3,D4} f_{R3,D4} + f_{R3}^{C}$$

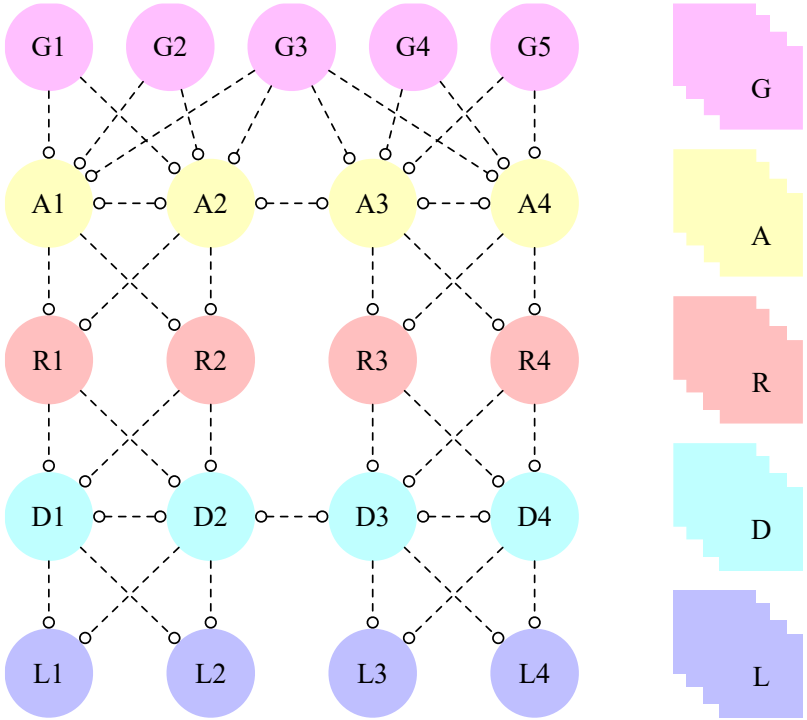
System-level:

- *Assumptions:* The generated flow is bounded by the capacity of the source nodes.
- *Guarantees:* The total flow consumption is bounded.

Select a Minimum-Cost Architecture Satisfying the Component-Level Contracts

$$\min_{e,m} \sum_{i=1}^{N} \alpha_i \beta_i c_i$$

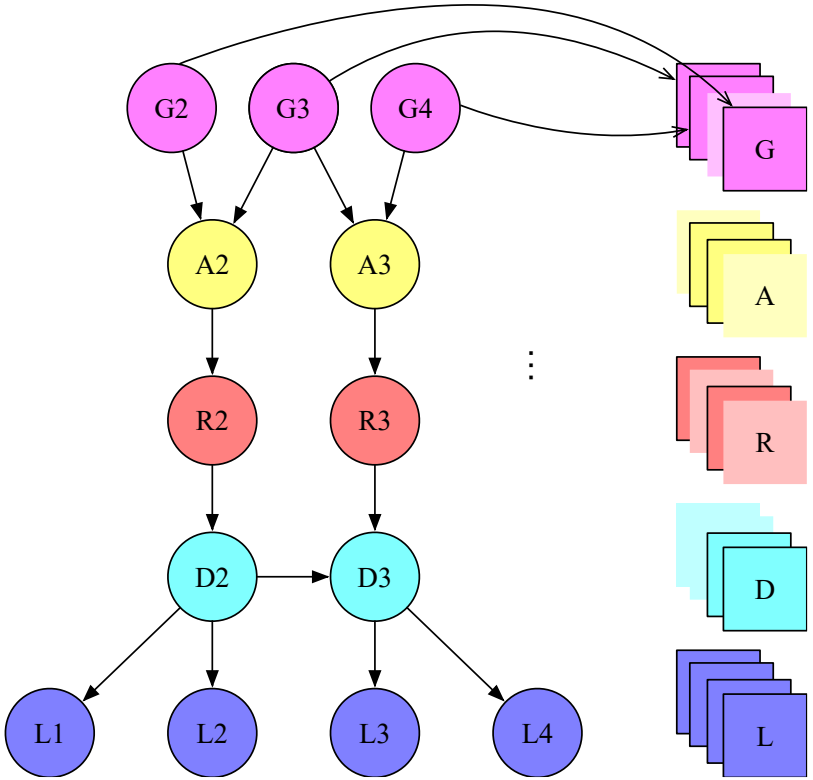
$$\text{s.t. } \forall d, \bigwedge_{i=1}^{N} \phi_{A_i^d} \wedge \phi_{G_i^d}$$



Select a Minimum-Cost Architecture Satisfying the Component-Level Contracts

$$\min_{e,m} \sum_{i=1}^{N} \alpha_i \beta_i c_i$$

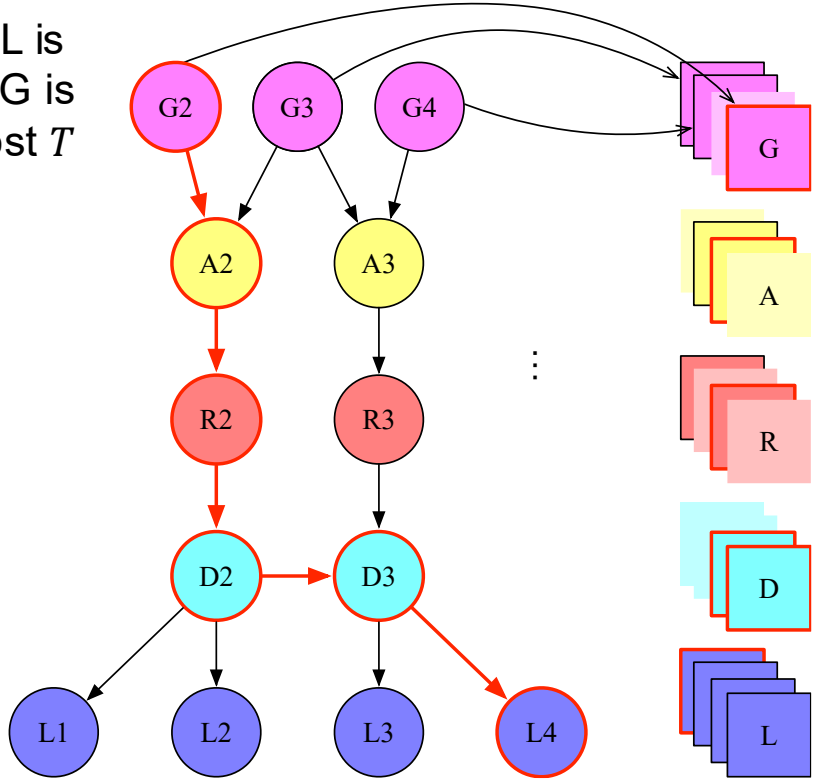
$$\text{s.t. } \forall d, \bigwedge_{i=1}^{N} \phi_{A_i^d} \wedge \phi_{G_i^d}$$



Selected Architecture

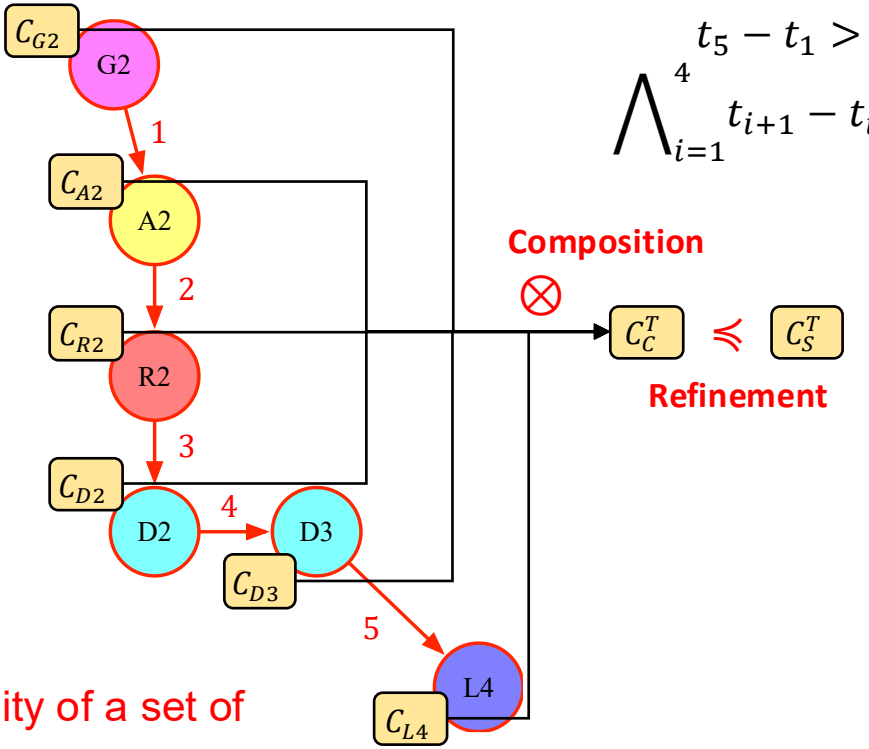
Check Whether the Selected Architecture Satisfies the System-Level Contracts Via Refinement Checking

The delay by which a load L is powered after a generator G is switched on must be at most T



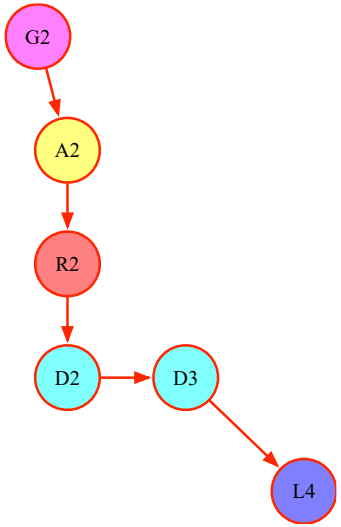
Check Whether the Selected Architecture Satisfies the System-Level Contracts Via Refinement Checking

The delay by which a load L is powered after a generator G is switched on must be at most T

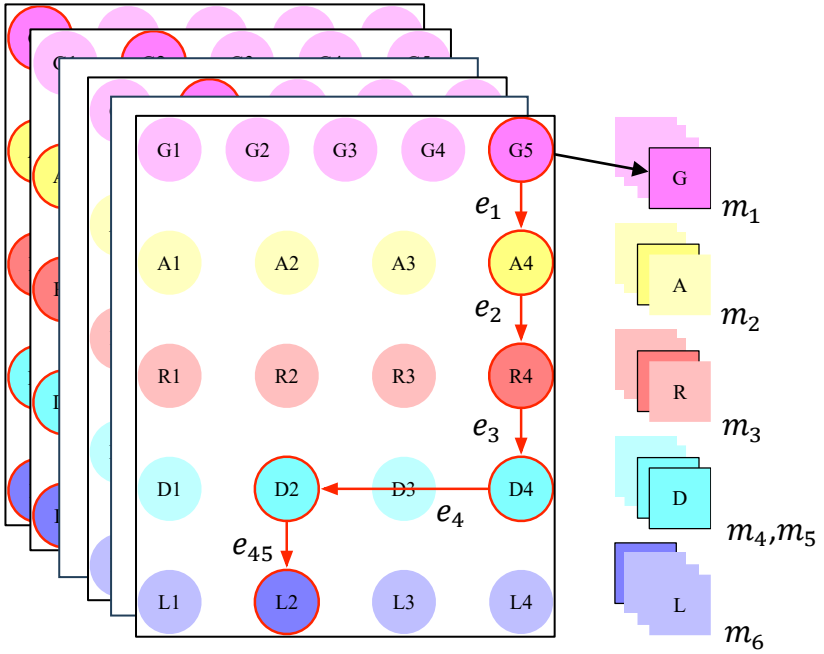


Reduced to checking the infeasibility of a set of mixed integer linear constraints

Generate Infeasibility Certificates Via Subgraph Isomorphism Analysis



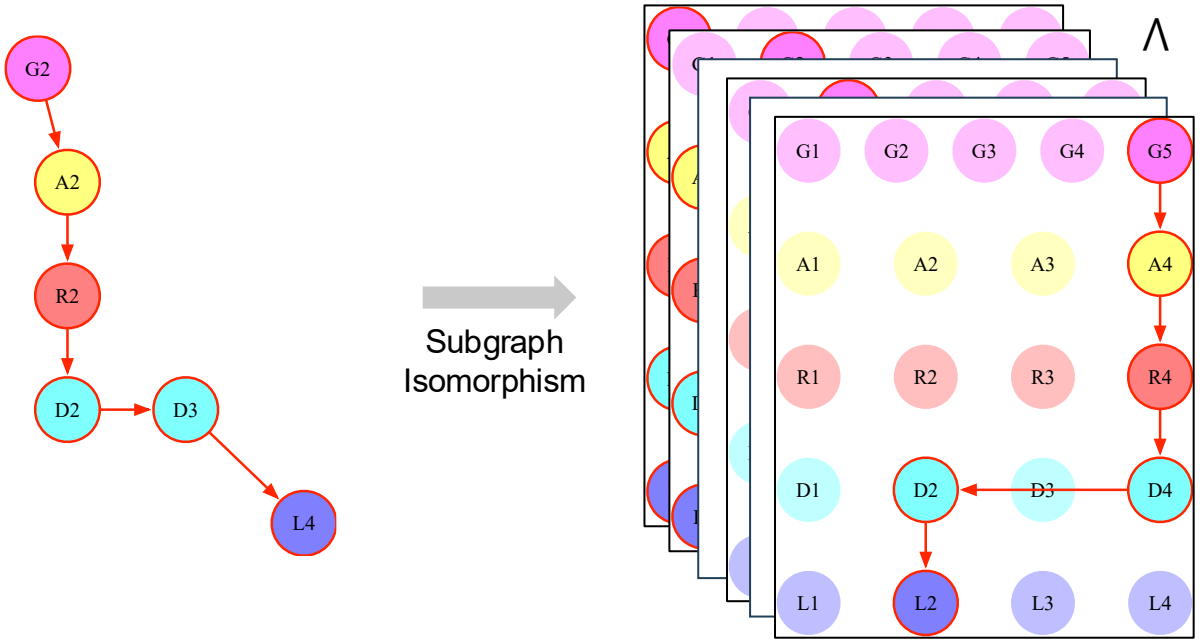
Subgraph Isomorphism



If a path is invalid, then the corresponding assignment to the edge and mapping variables should no longer be selected

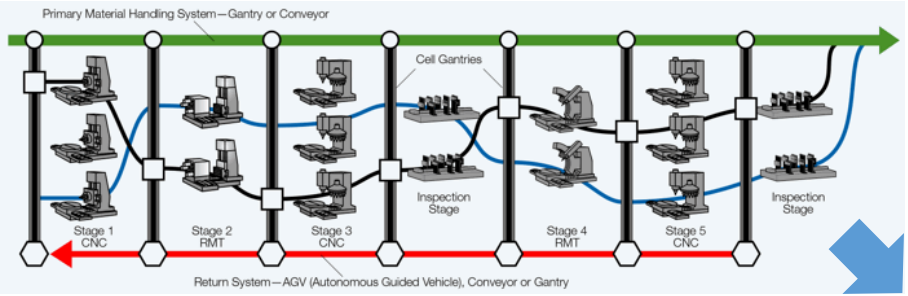
$$\sum_i e_i + \sum_i m_i < 5 + 6$$

Generate Infeasibility Certificates Via Subgraph Isomorphism Analysis

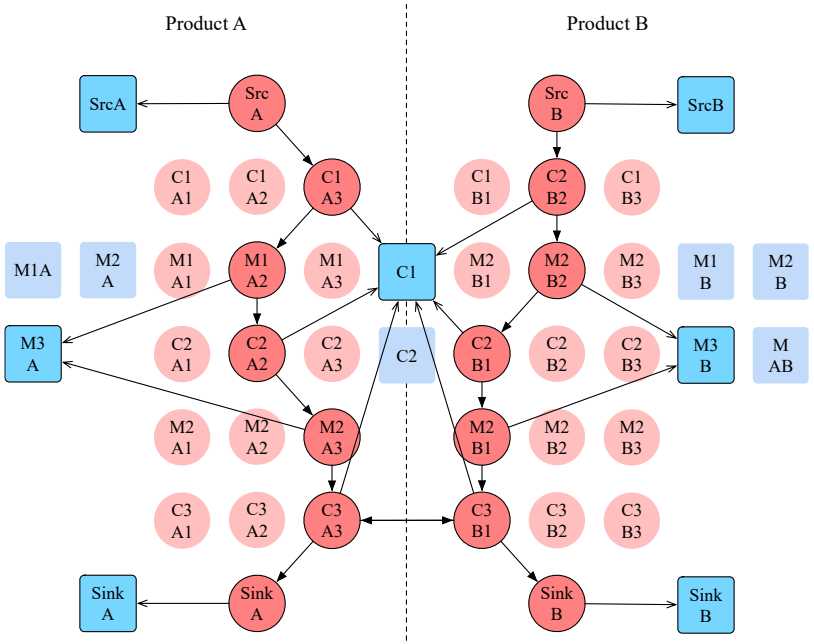


The constraints will be added to the **MILP-based problem** for architecture selection in the next iteration

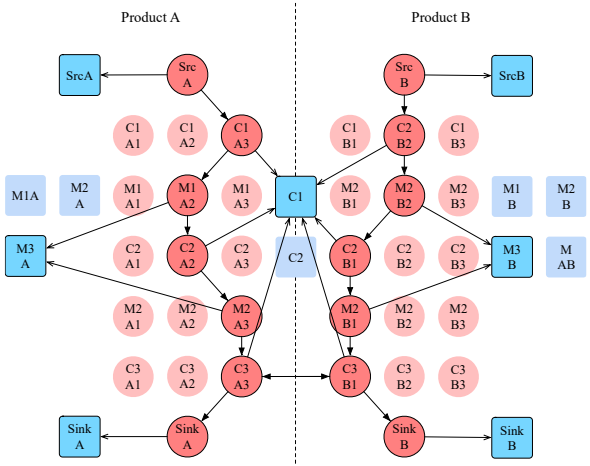
Case Study: Reconfigurable Production Line (RPL)

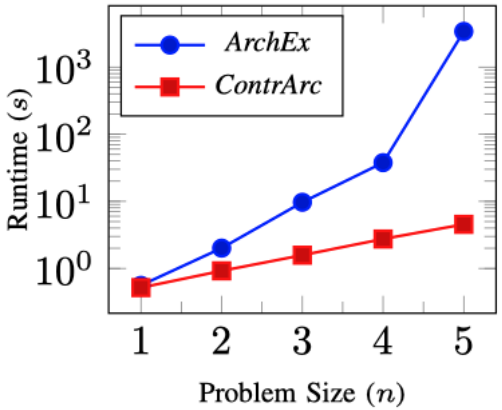


- Components (in red): Source (Src), Machine (M), Conveyor (C), Sink
- Implementations (in blue)
- Assemble lines with the minimum cost such that the total delay to assemble a product is less than l, and the mass flow of product elements are balanced.



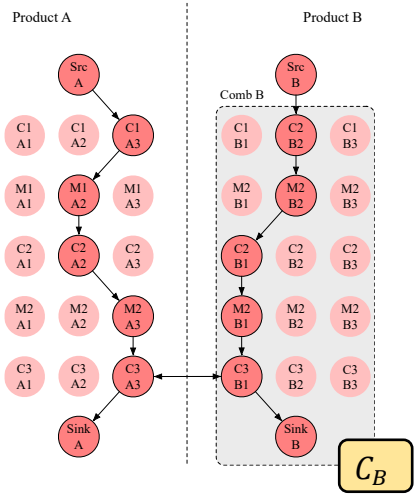
Case Study: Reconfigurable Production Line (RPL)

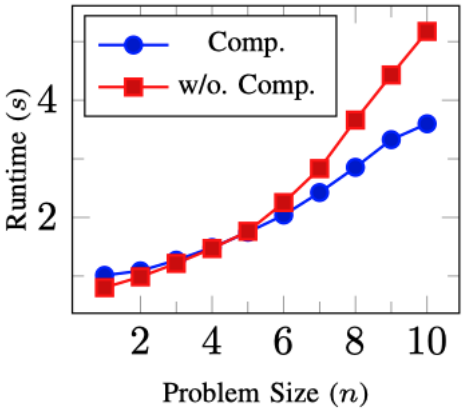




- Up to **two orders of magnitude** acceleration with respect to ***ArchEx*** [K. Dmitrii, et al., 2017]

Case Study: Reconfigurable Production Line (RPL)

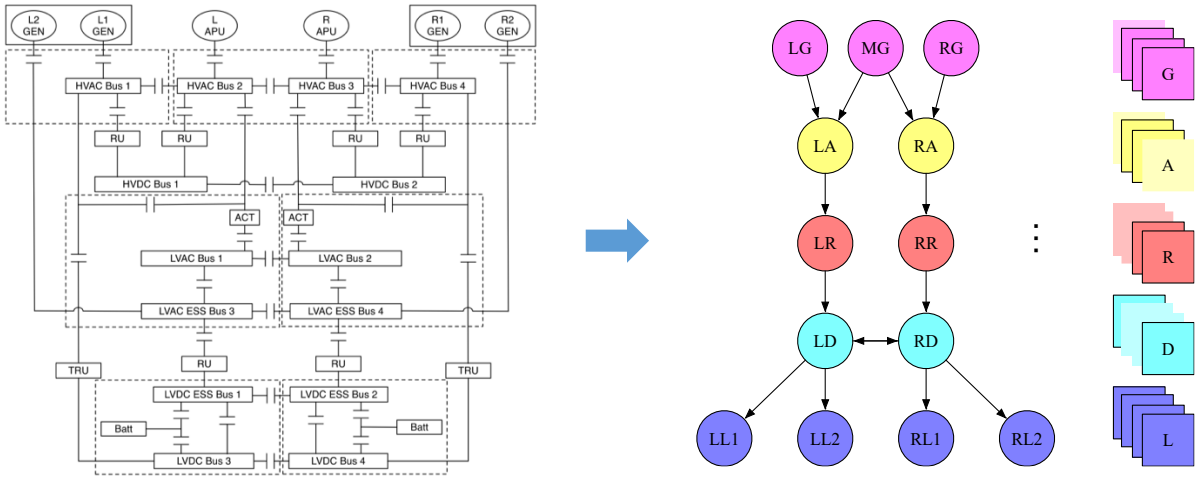




- Up to **two orders of magnitude** acceleration with respect to ***ArchEx*** [K. Dmitrii, et al., 2017]

- **Compositional Exploration**: Partition the system and synthesize each line independently under appropriate assumptions

Case Study: Electrical Power Network (EPN) [Nuzzo et al., 2014]



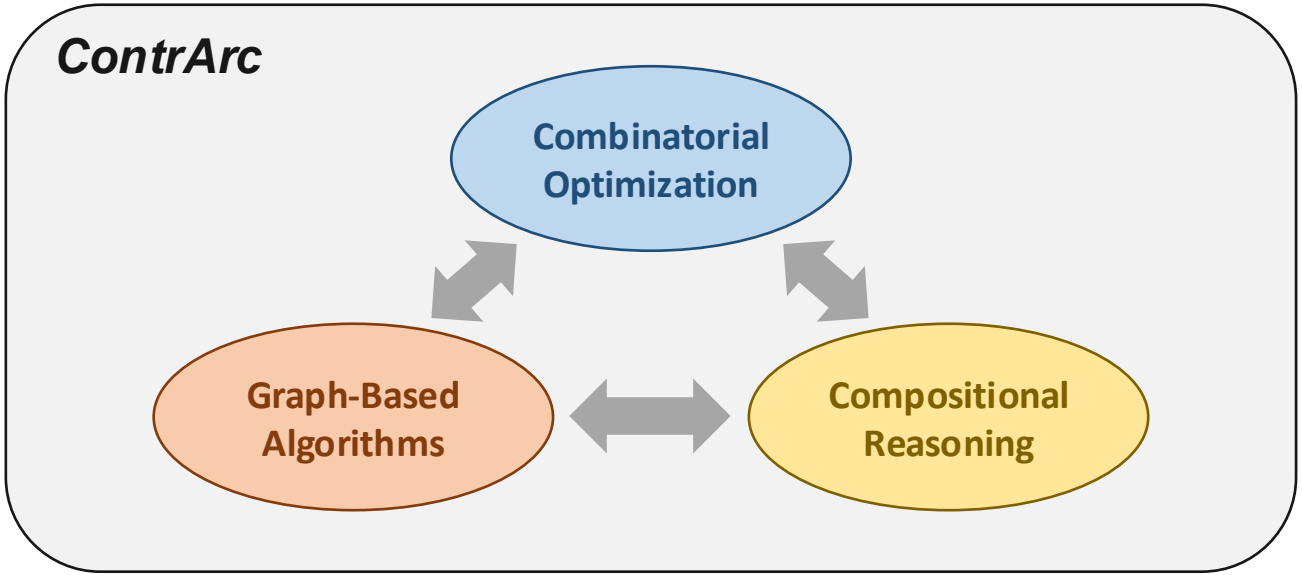
- Components (on left (L) and right(R) side): generator (GEN), AC bus, rectifier unit (RU), DC bus, load (L)
- Each component has four implementations
- Find the lowest cost EPN to power a set of critical loads, each requiring at least power p, with a delay less than l

Case Study: Electrical Power Network (EPN)

Max # in $\mathcal{T}$ (L,R,APU)	# of variables	# of constraints	w/o. decomposition		w/o. subgraph isomorphism		Complete ContrArc	
			Time (s)	# of iterations	Time (s)	# of iterations	Time (s)	# of iterations
1,0,0	454	195	0.57	3	0.58	3	0.56	3
2,0,0	1178	592	4.78	8	10.53	28	2.50	4
3,0,0	2280	1281	50.21	12	84.77	104	8.52	6
4,0,0	3868	2352	6.31×10^3	18	4.45×10^3	231	20.55	4
1,1,0	1138	576	11.18	22	10.72	24	9.15	24
2,1,0	2374	1383	4.09×10^3	93	4.82×10^2	320	27.12	20
2,2,0	4004	2508	2.73×10^4	152	5.59×10^3	1581	1.55×10^2	34
1,1,1	1294	666	62.79	85	13.89	30	16.26	31
2,1,1	2604	1532	1.57×10^2	56	1.99×10^2	168	40.94	26
2,2,1	4320	2726	2.35×10^3	60	3.87×10^3	1353	1.06×10^2	23
Average			4.04×10^3	50.9	1.07×10^3	384.1	38.67	17.5
Ratio			104.36	2.91	27.68	21.95	**1.00**	**1.00**

- **Compositional refinement checking** enables about **two orders of magnitude** acceleration
- **Subgraph isomorphism**-based analysis enables about **20 times less iterations** on average

Conclusions



Future Work:

- Extensions to support a broader set of requirements
- Integration of other graph-based algorithms with optimization methods