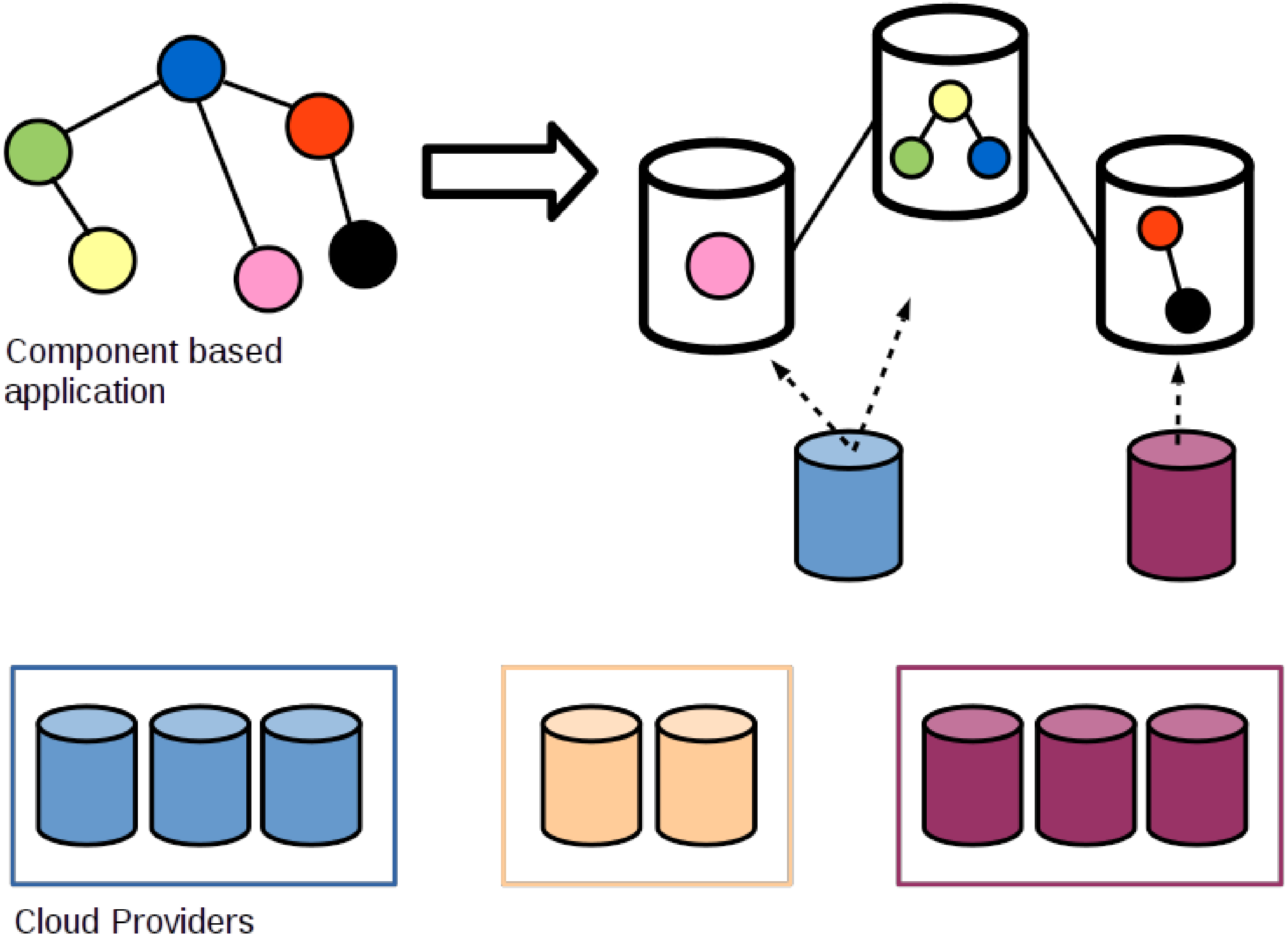
Efficient Heuristics for the Placement of Large Scale Distributed Applications on Multiple Clouds

Pedro Silva
Frédéric Desprez, Christian Perez

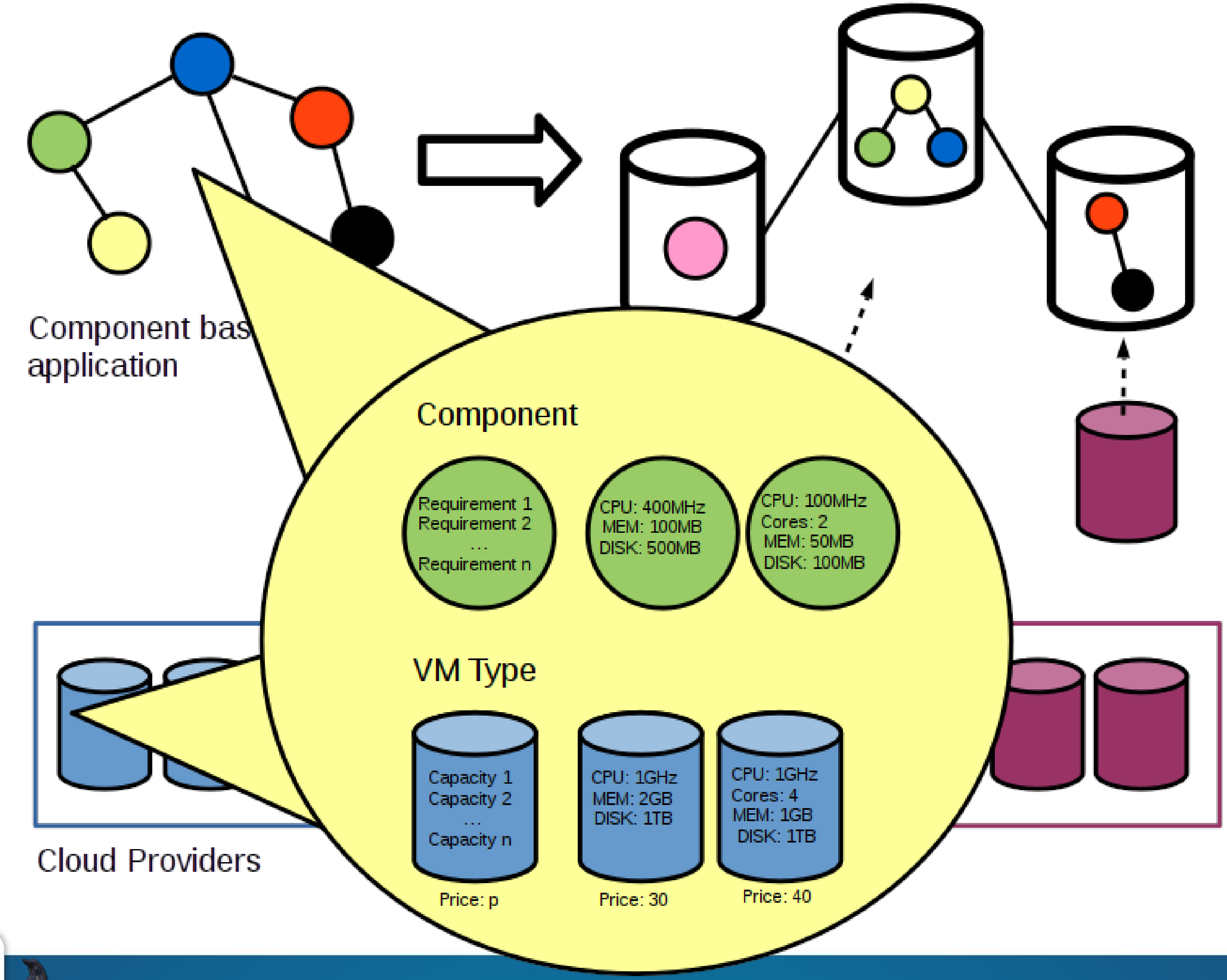
Discovery Meeting @ Rennes

2017, March 30-31

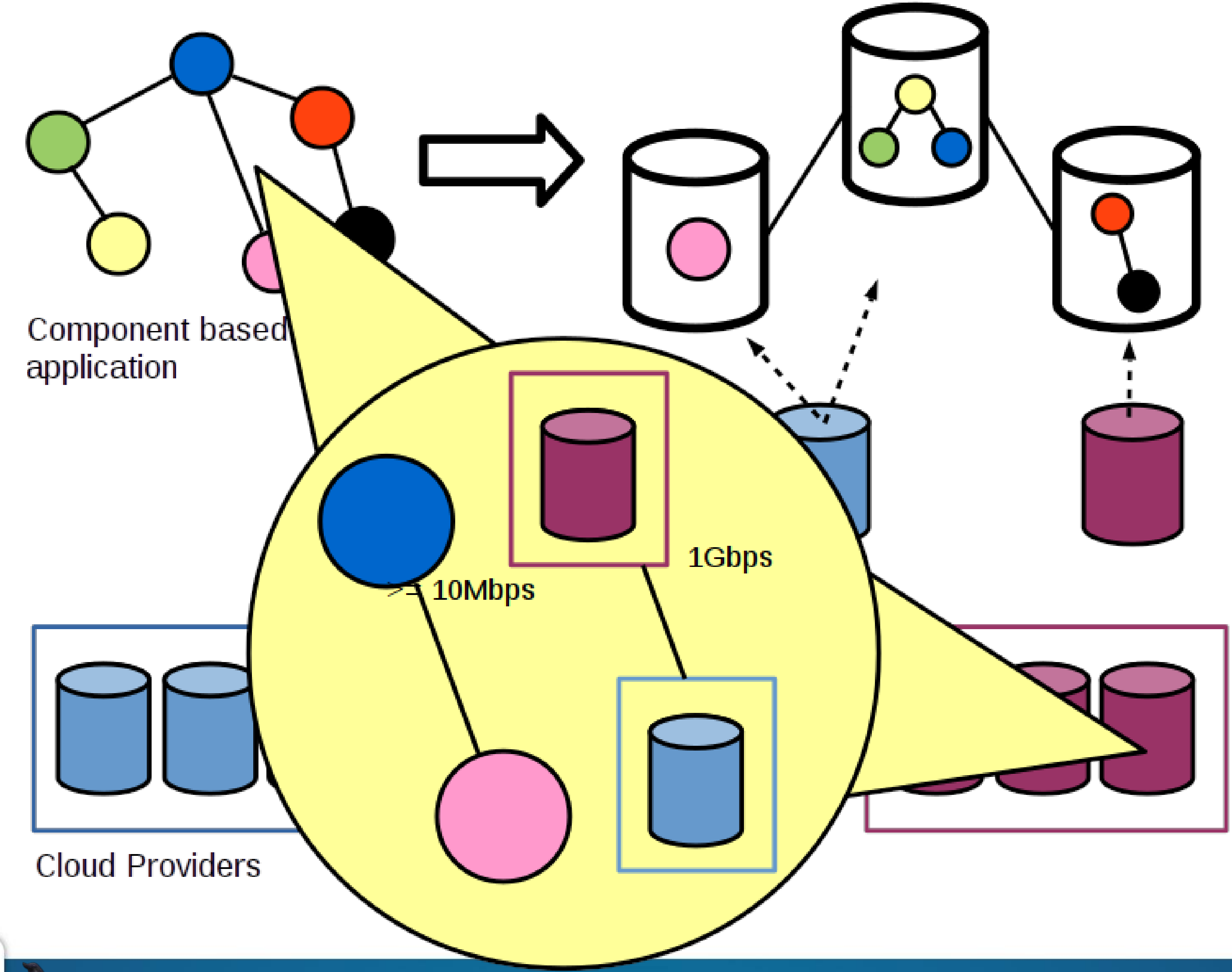
Application Placement on (Multi) Cloud



Resource Constraints



Communication Constraints



Problem Statement / Methodology

Minimize cost and satisfy resource and communication constraints

Problem divided into three sub problems

- Initial placement, communication oblivious, concrete application
- Initial placement, communication aware, concrete application
- Reconfiguration, communication aware, concrete application
- Toward parametric/abstract application

Evaluation through comparison to the state of the art

Approaches to solve the placement problem

Exact Algorithms

- Solvers
- Model details, optimal
- Scalability

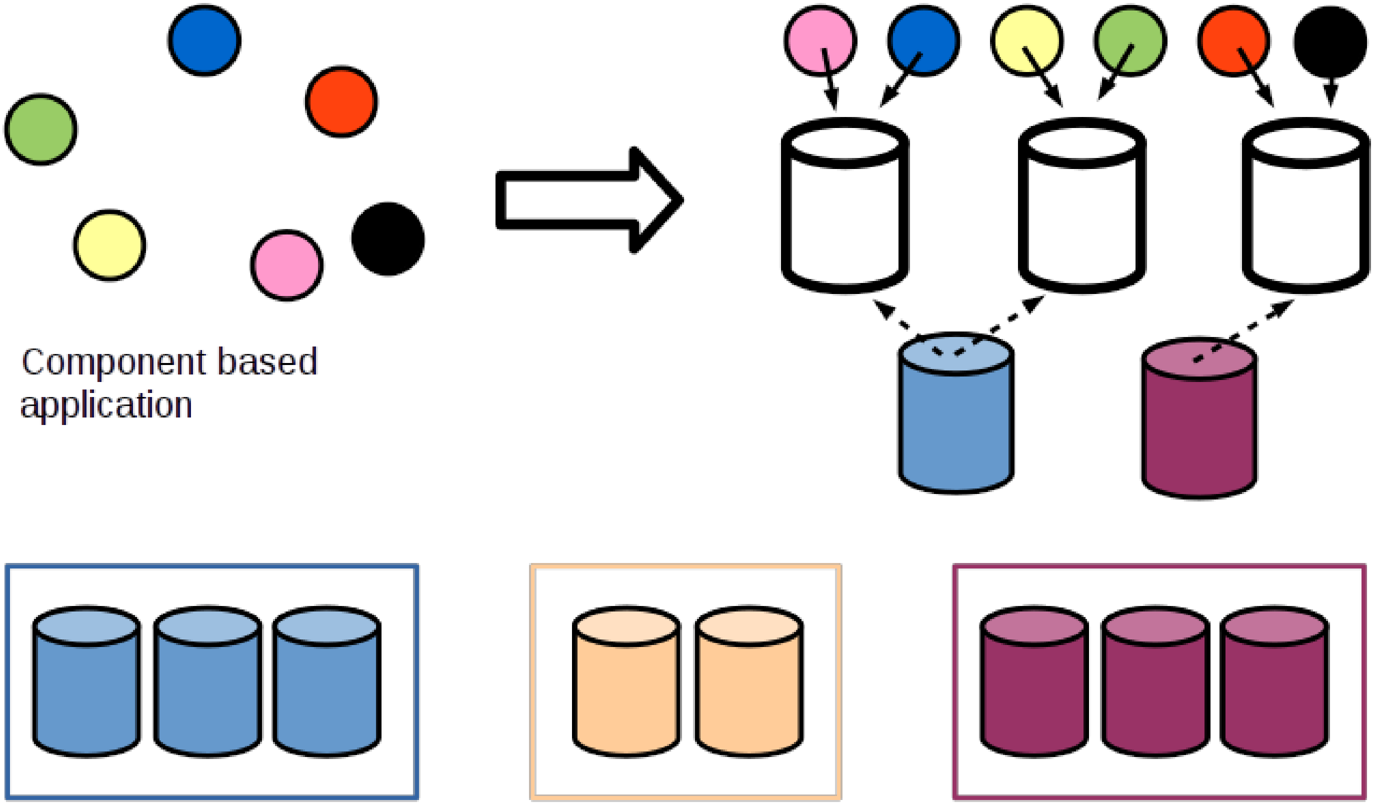
Meta-heuristics

- Genetic algorithm, Simulated annealing, Particle Swarm optimization, etc.
- Model details, generic
- Execution time

Heuristics

- Greedy heuristics, graph based (clustering, partitioning, etc)
- Scalability
- Model details, solution quality

Initial Placement, Communication Oblivious



Component based application

Cloud Providers

P. Silva, C. Pérez, F. Desprez. Efficient Heuristics for Placing Large-Scale Distributed Applications on Multiple Clouds. In 16th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid'16), May 2016, Cartagena, Colombia. 2016, <10.1109/CCGrid.2016.77>. <hal-01301382>

Contribution: Greedy Heuristics

Adaptation of three heuristics

- First Fit Decreasing Priority
  - The size is the maximal normalized value of dimensions.
- First Fit Windowed Multi-Capacity
  - It tries to balance the usage of resources from bins.
- Best Fit Dot Product:
  - Most-adapted bin: Bin that maximizes the dot product between item and bins:

Adaptations

- Cost-awareness
  - Ordered bin types by size/price ratio
- Heterogeneity of bin types
  - Created an ideal bin type as reference

Problem Classes

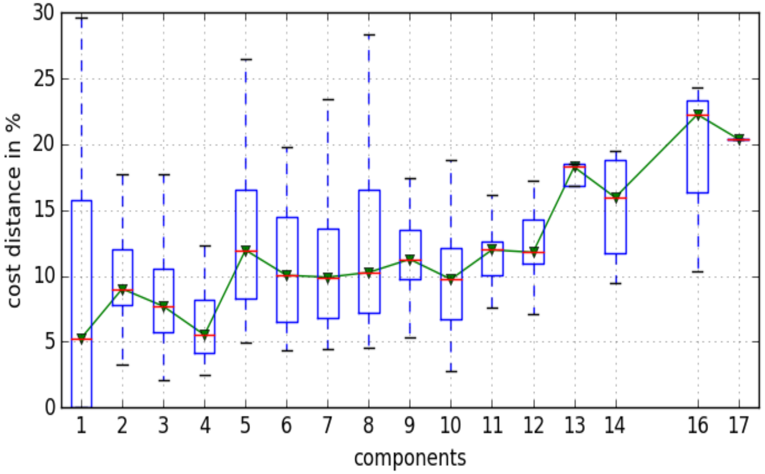
Compared approaches

- **Exact algorithm**: IPL Solver (SCIP solver)
  - 30h per problem
- **Meta heuristics**: Simulated annealing (SA)
  - 10 min per problem
- **Block of greedy heuristics**
  - Only the lowest cost is kept
  - The execution times are summed up

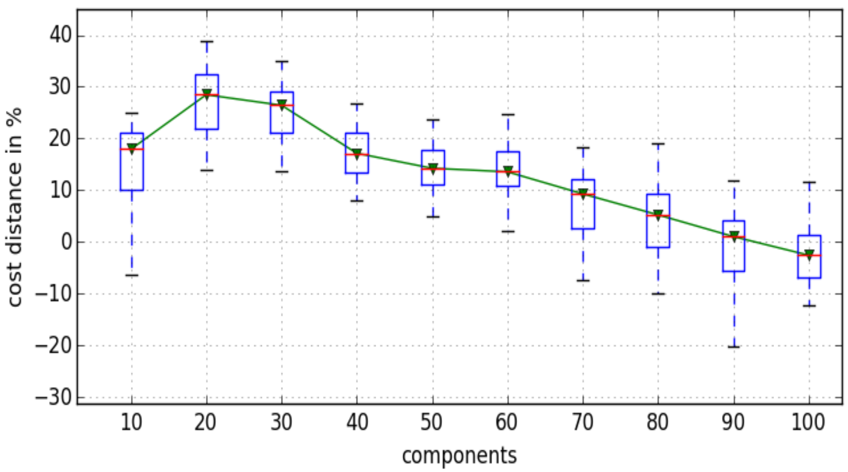
Generated problems (Uniform random)

	Resource/Dimensions	Component/Items	VMs/bin types	Total
A	1, 2, 4, 8	1, 2, …, 19	100, 200, …., 1000	1520
B	1, 2, 4, 8	10, 20, ..., 100	1k, 2k, …., 10 k	800

Evaluation



Solver vs. Heuristics on Class A

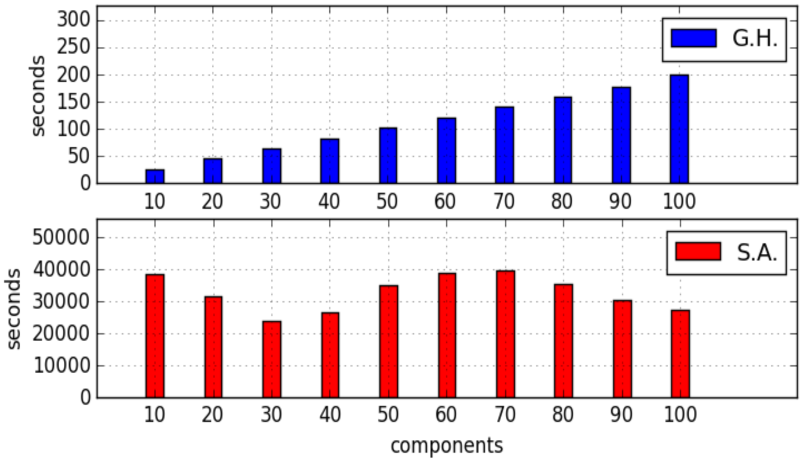


S.A. vs Heuristics on Class B

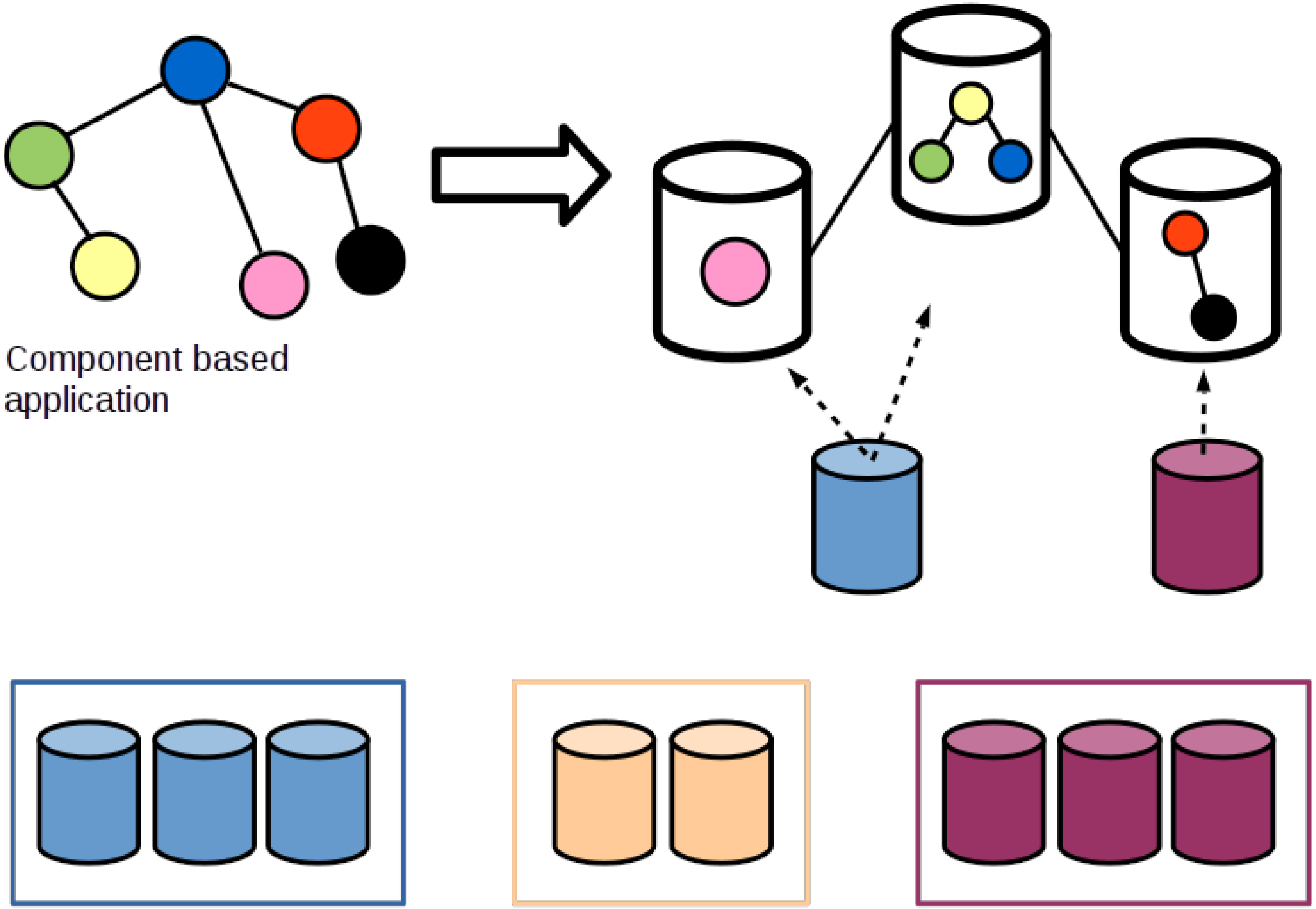
Solver

- Solve around 34% of Class A

	Average (s)	Maximum (s)
BF-DP	1.40	5.70
FFD-P	0.03	0.23
FFD-WMC	0.10	0.35



Initial Placement, Communication Aware

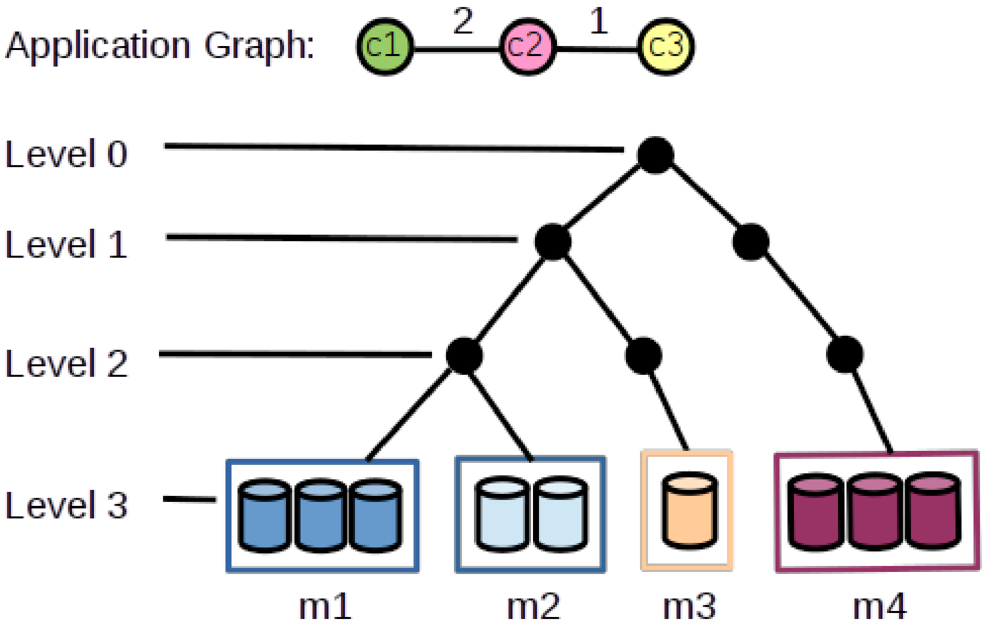


Component based application

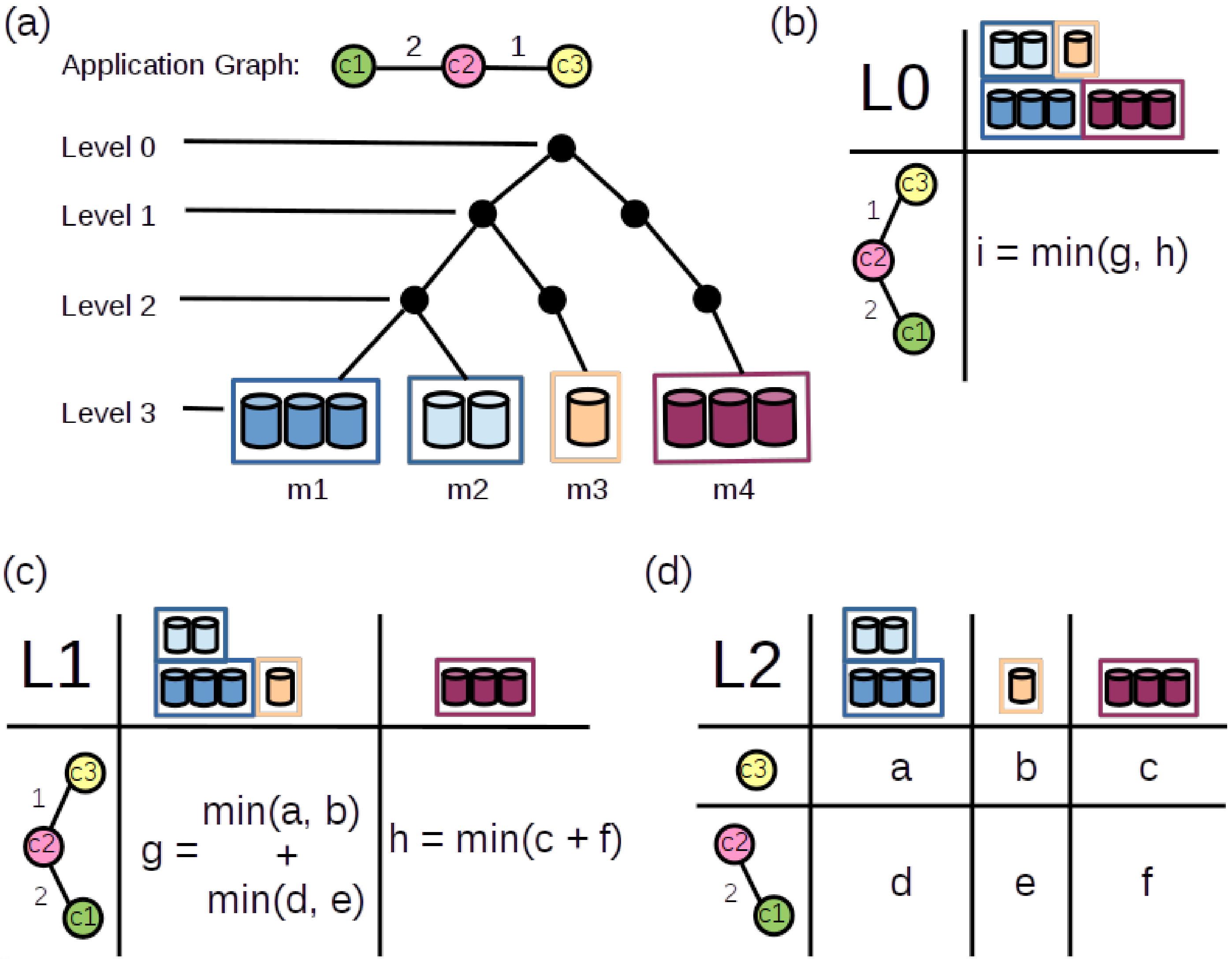
Cloud Providers

P. Silva, C. Perez, F. Desprez, An Efficient Communication Aware Heuristic for Multiple Cloud Application Placement Submitted to Euro-Par 2017

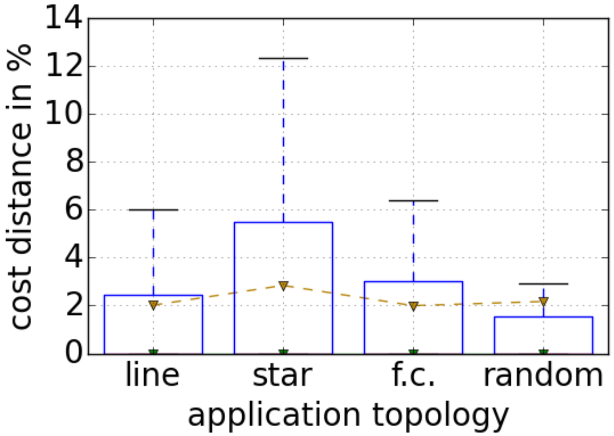
A Hierarchical Resource Model



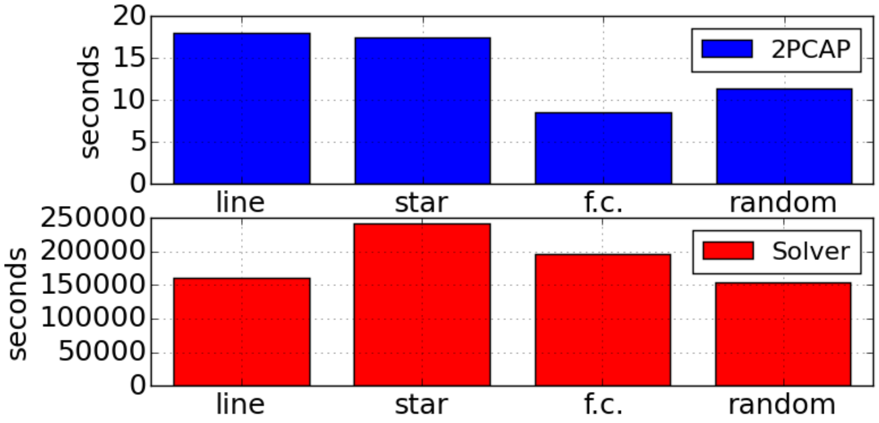
A Greedy Heuristic: 2PCAP



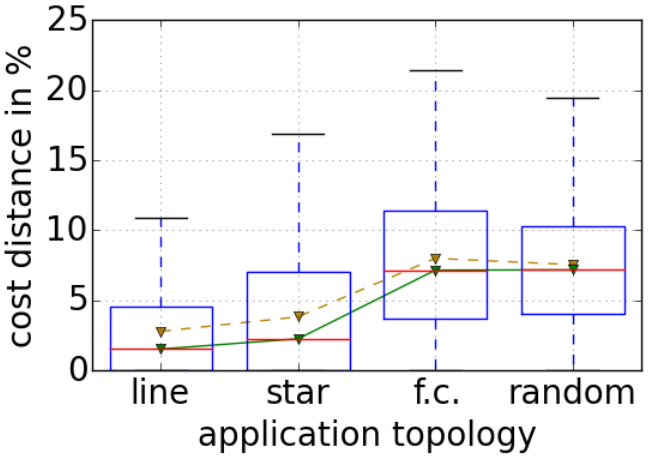
Evaluation: Exact Solver vs SA vs 2PCAP



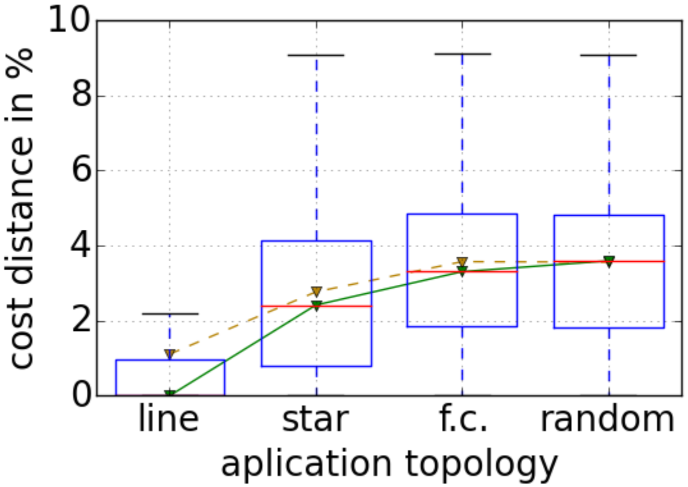
Cost: 2PCAP vs MIP - Small



Exec. Time: 2PCAP vs MIP - Small

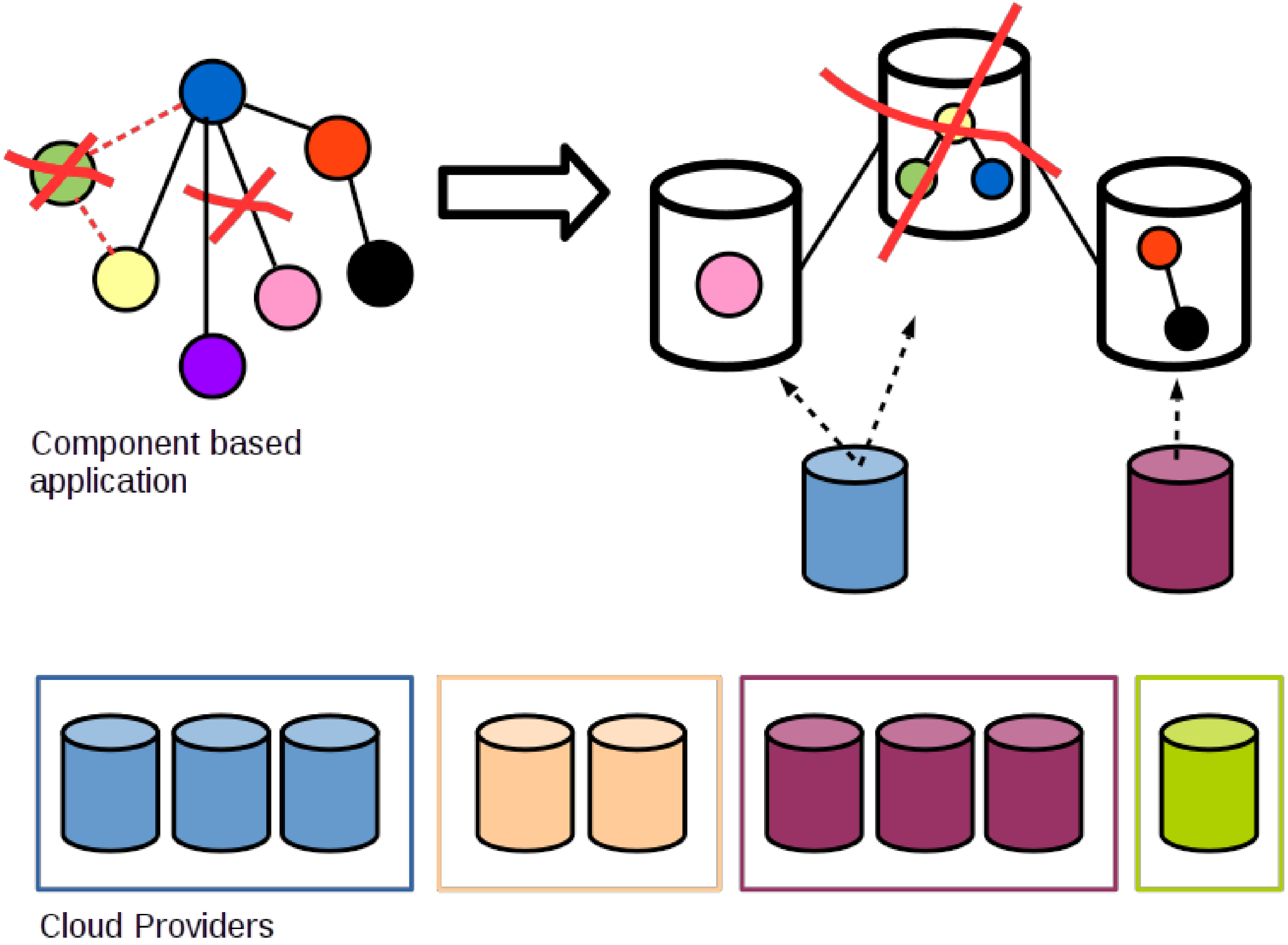


2PCAP vs Baseline Large (No communication)



2PCAP + S.A. - Large

Reconfiguration (WIP)



Component based application

Cloud Providers

Current Status

Adaptation of 2PCAP

- Adding a migrating cost
- Not the smartest one but based on the cheap computation of a solution

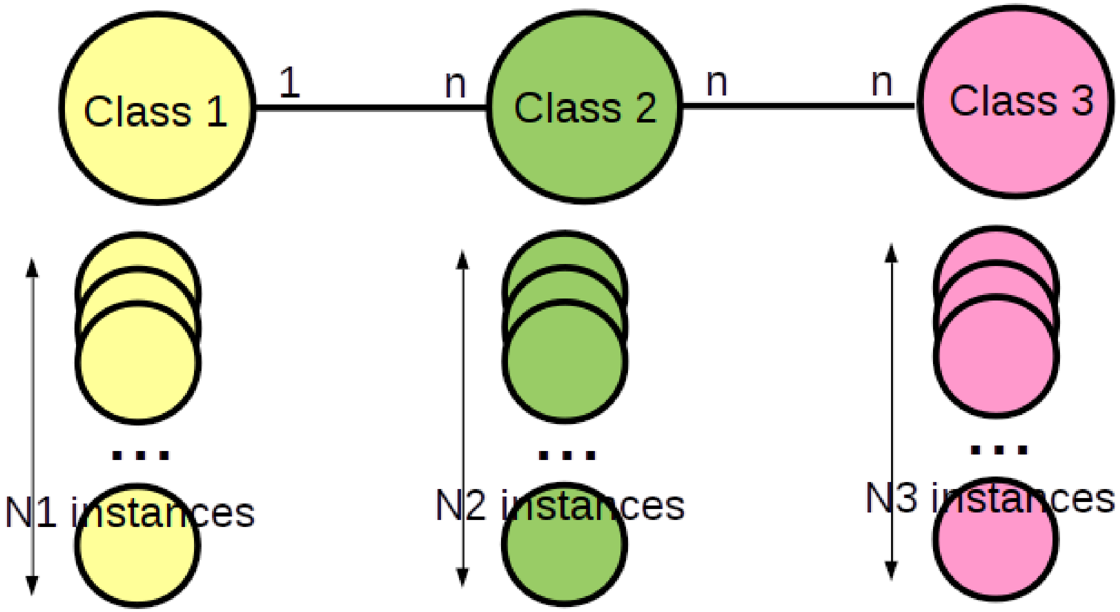
First version of the algorithm implemented

Preparing experiments for evaluation

Parametrizing Application Model

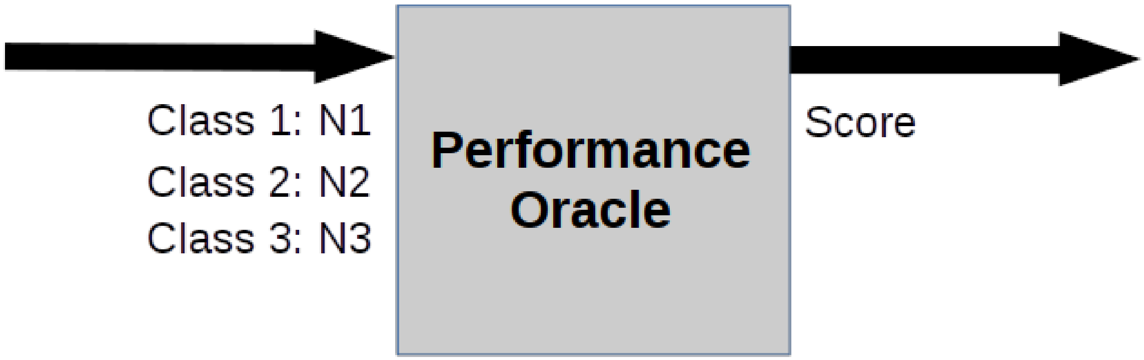
Component Cardinality

- How many instances of each component class needs to satisfy a given performance constraint?
- Need a performance model
- Need a connector model



Status

- 1st modeling done
- 2PCAP modified
- Implementation in progress



Class 1: N1
Class 2: N2
Class 3: N3

Performance Oracle

Score

Conclusion and Relationship with Discovery

Heuristics to quickly compute a solution

- Large number of VM types (10,000’)
- Moderate application size (100’)
- Parametrized number of dimension (~8)
- Hierarchical communication constraint model
- Any concrete application graph
    - Extension toward component cardinality in progress

Relationship with Discovery

- Set of efficient algorithms to deal with initial placement
- Direction of future work
    - Application model and Discovery model
    - Hierarchal modeling and localization constraints
    - Distributed reconfiguration vs “centralized” heuristics
- Determine use cases to understand objective functions