Models and Algorithms for the optimal allocation of services in Vehicle-to-Cloud architectures

Master's Thesis by Alessandro Minoli (20202A)

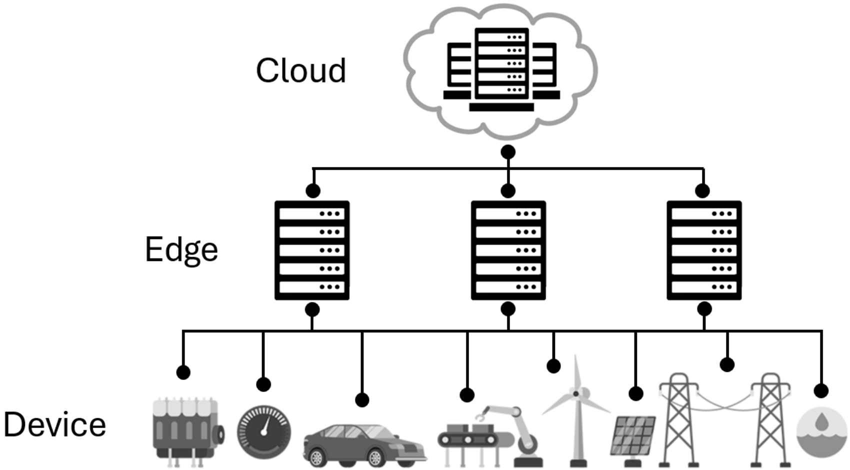
Supervisor : Alberto Ceselli , Co-supervisor : Christian Quadri

11/04/2025

UNIVERSITÀ DEGLI STUDI DI MILANO

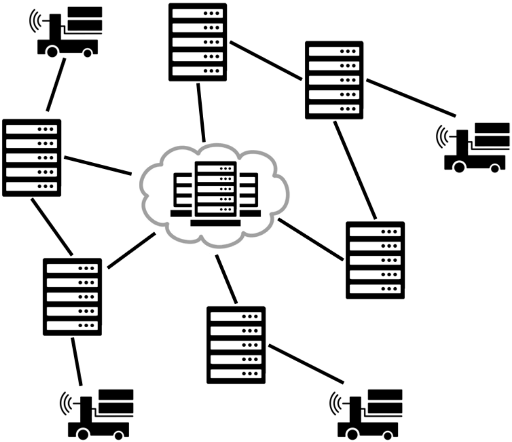
Introduction

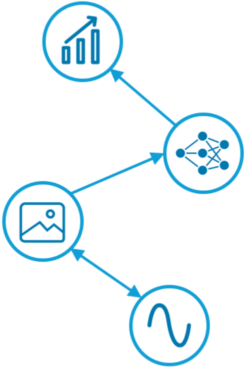
- 3-tier network provisioning architecture



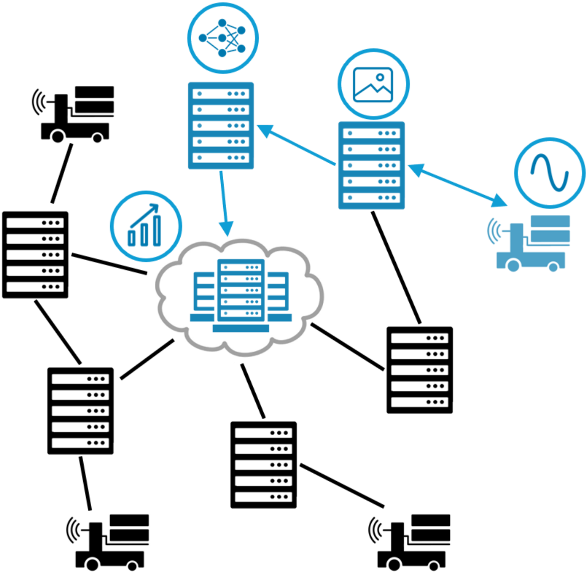
Objective

- *"CAVIA: enabling the Cloud-to-Autonomous-Vehicles continuum for future Industrial Applications"*





- integer optimization and heuristics for the mapping problem



Problem Data

- $G_{network} = (I, A)$
  - routing paths : $\mathcal{P}_{ij} \;\; \forall i \in I, \forall j \in I \qquad \rightarrow \qquad a_{ij}^{lm} = 1$ if $(l, m) \in \mathcal{P}_{ij}$
  - allocation costs : $c_i \;\; \forall i \in I$

- $G_{app} = (T, D)$

- consumable resources $R = \{\, core\, ,\; bandwidth \,\}$
- properties $S = \{\, has\_camera\, ,\; has\_gpu\, ,\; latency \,\}$

- resource availability $\forall k \in R \cup S$
  - $Q_i^k \;\; \forall i \in I \;,\;\; Q_{ij}^k \;\; \forall (i, j) \in A$
- resource requirement $\forall k \in R \cup S$
  - $q_u^k \;\; \forall u \in T \;,\;\; q_{uv}^k \;\; \forall (u, v) \in D$

- $b_u^i = 0$ if u alone cannot be mapped to i
- $b_{uv}^{ij} = 0$ if u, v alone cannot be mapped to i, j

Mathematical Model

- binary decision variables $x_{ui} \in \{0,1\} \quad \forall u \in T, \forall i \in I$
  which are 1 if micro-service u is mapped to node i, 0 otherwise

$$
\begin{aligned}
\text{minimize} \quad & \sum_{u \in T} \sum_{i \in I} c_i \cdot x_{ui} & \\
\text{s.t.} \quad & \sum_{i \in I} x_{ui} = 1 & \forall u \in T \\
& x_{ui} = 0 & \forall i \in I, \forall u \in T \mid b_u^i = 0 \\
& x_{ui} + x_{vj} \leq 1 & \forall i \in I, \forall j \in I, \forall (u,v) \in D \mid b_{uv}^{ij} = 0 \\
& \sum_{u \in T} q_u^k \cdot x_{ui} \leq Q_i^k & \forall k \in R, \forall i \in I \\
& \sum_{i \in I} \sum_{j \in I} \sum_{(u,v) \in D} q_{uv}^k \cdot a_{ij}^{lm} \cdot x_{ui} \cdot x_{vj} \leq Q_{lm}^k & \forall k \in R, \forall (l,m) \in A
\end{aligned}
$$

- BQP with a polynomial number of variables and constraints

Algorithms - Column Generation (1)

- set of applications N, each $n \in N$ modeled as $G^n_{app} = (T^n, D^n)$
- Dantzig–Wolfe decomposition
- $\overline{\Omega^n} \subseteq \Omega^n = \{$all feasible mappings of app n onto the network$\}$
- LRMP contains linking constraints imposing that the mappings, all together, do not exhaust the available network resources
- continuous decision variables $\quad 0 \leq \theta^n_p \leq 1 \quad \forall n \in N, \forall p \in \overline{\Omega^n}$
- Column Generation technique

Algorithms - Column Generation (2)

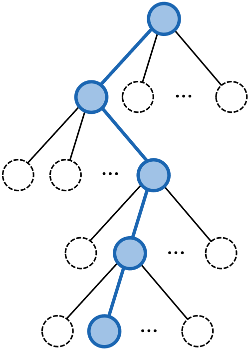
- PPs find negative reduced cost columns to add to the LRMP
- pricing problem PP^n, $\forall n \in N$

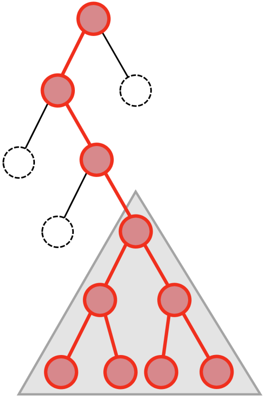
$$\begin{aligned}
\text{minimize } c_{\mathrm{PP}^n} = & \sum_{u \in T^n} \sum_{i \in I} c_i \cdot x_{ui} \\
& + \sum_{k \in R} \sum_{i \in I} \sum_{u \in T^n} [\, q_u^k \cdot \lambda_i^k \,] \cdot x_{ui} \\
& + \sum_{k \in R} \sum_{(l,m) \in A} \sum_{i \in I} \sum_{j \in I} \sum_{(u,v) \in D^n} [\, q_{uv}^k \cdot a_{ij}^{lm} \cdot \mu_{lm}^k \,] \cdot x_{ui} \cdot x_{vj} \\
& - \eta_n \\
\text{s. t.} \quad & \text{all the constraints of the original model are satisfied}
\end{aligned}$$

- LRMP and PPs are optimized with an integer solver
- Lagrangean relaxation is used to mitigate CG “tailing-off” effect

Algorithms - Heuristics

- constructive metaheuristic for solving PPs
- three matheuristics to get primal bounds from CG information
  - Discrete LRMP
  - Pure Diving
  - Rounding + SubMIPing





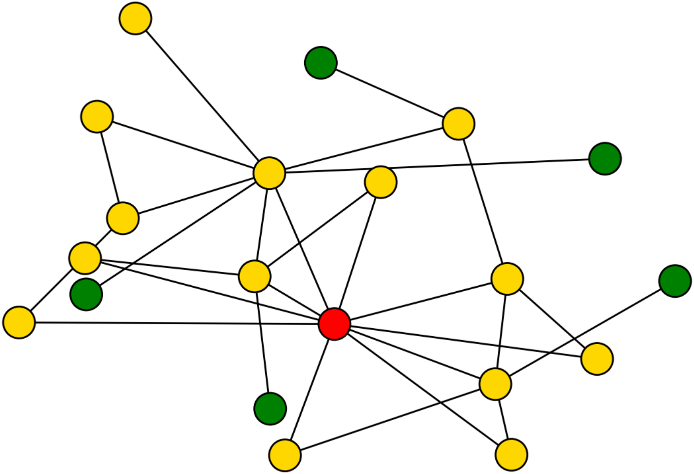
Implementation

- no dataset already available
- random generator of networks , $|I| \in \{30, 40, 50, 60, 70\}$
- random generator of applications , $3 \leq |T| \leq 15$

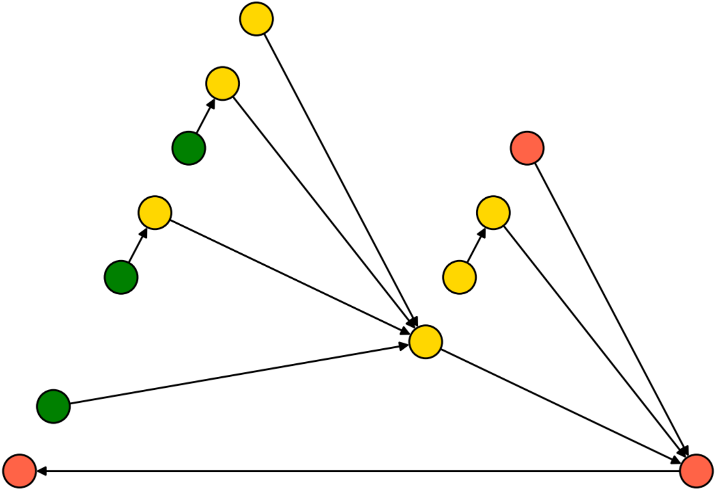
- *Python* programming language
- *Gurobi Optimizer* (version 11.0.2)
- *Apple M1* CPU, 16 GB RAM

- code available at : `https://github.com/nemolino/master-thesis`

- example of random network , $|I| = 20$

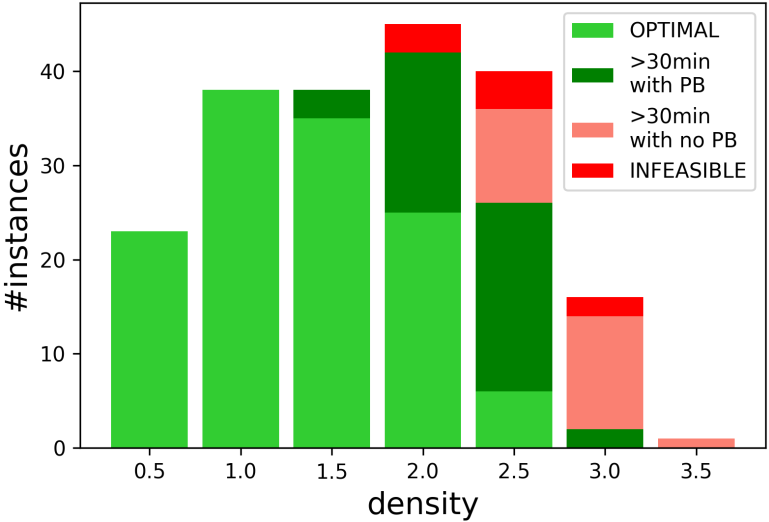


- example of random application with tree topology , $|T| = 12$

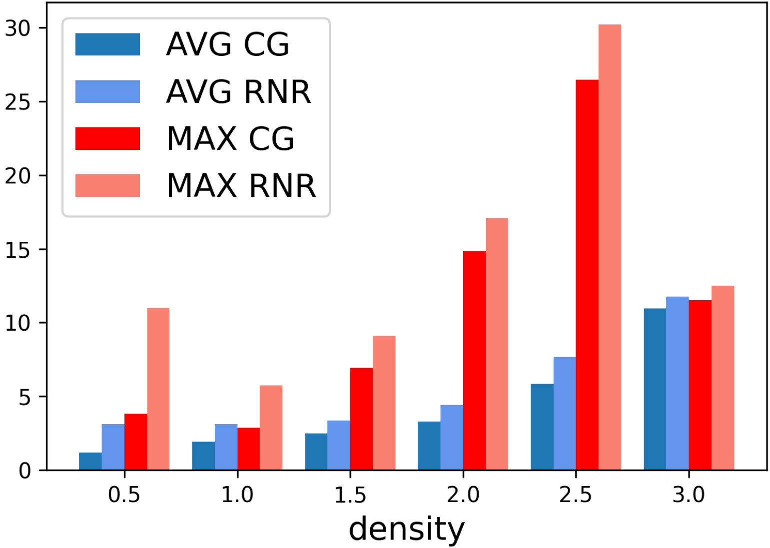


Computational results

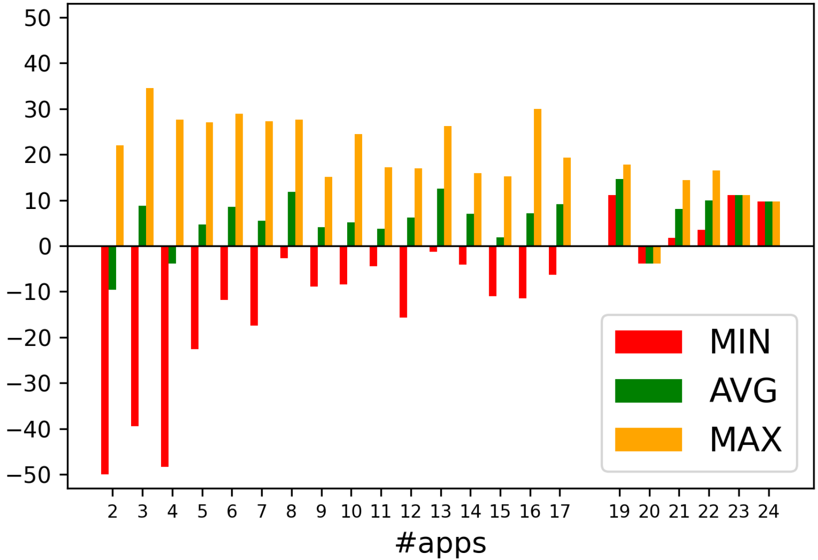
- solve status of *Gurobi*



- % gap of RNR and CG bounds
  with respect to the optimal solution (or primal bound)



- % time improvement of CG with heuristic pricing with respect to basic CG



- #successes of matheuristics