Neur2SP: Neural Two-Stage Stochastic Programming

Justin Dumouchelle* · Rahul Patel* · Elias B. Khalil · Merve Bodur

Mechanical & Industrial Engineering
UNIVERSITY OF TORONTO

Two-stage Stochastic Programming (2SP)

Objective: Determine optimal first-stage decisions that minimize sum of first-stage cost and expected second-stage cost.

Challenge: Exact optimization becomes exponentially harder with the number of observed samples (scenarios), *K*.

First-Stage Problem

$$\min_{\mathbf{x}} \left\{ \underbrace{\boxed{\mathbf{c}^\intercal \mathbf{x}}}_{\text{First-stage cost}} + \underbrace{\boxed{\sum_{k=1}^{K} p_k Q(\mathbf{x}, \xi_k)}}_{\text{Expected second-stage cost}} : \mathbf{x} \in \mathcal{X} \right\}$$

First-stage decisions

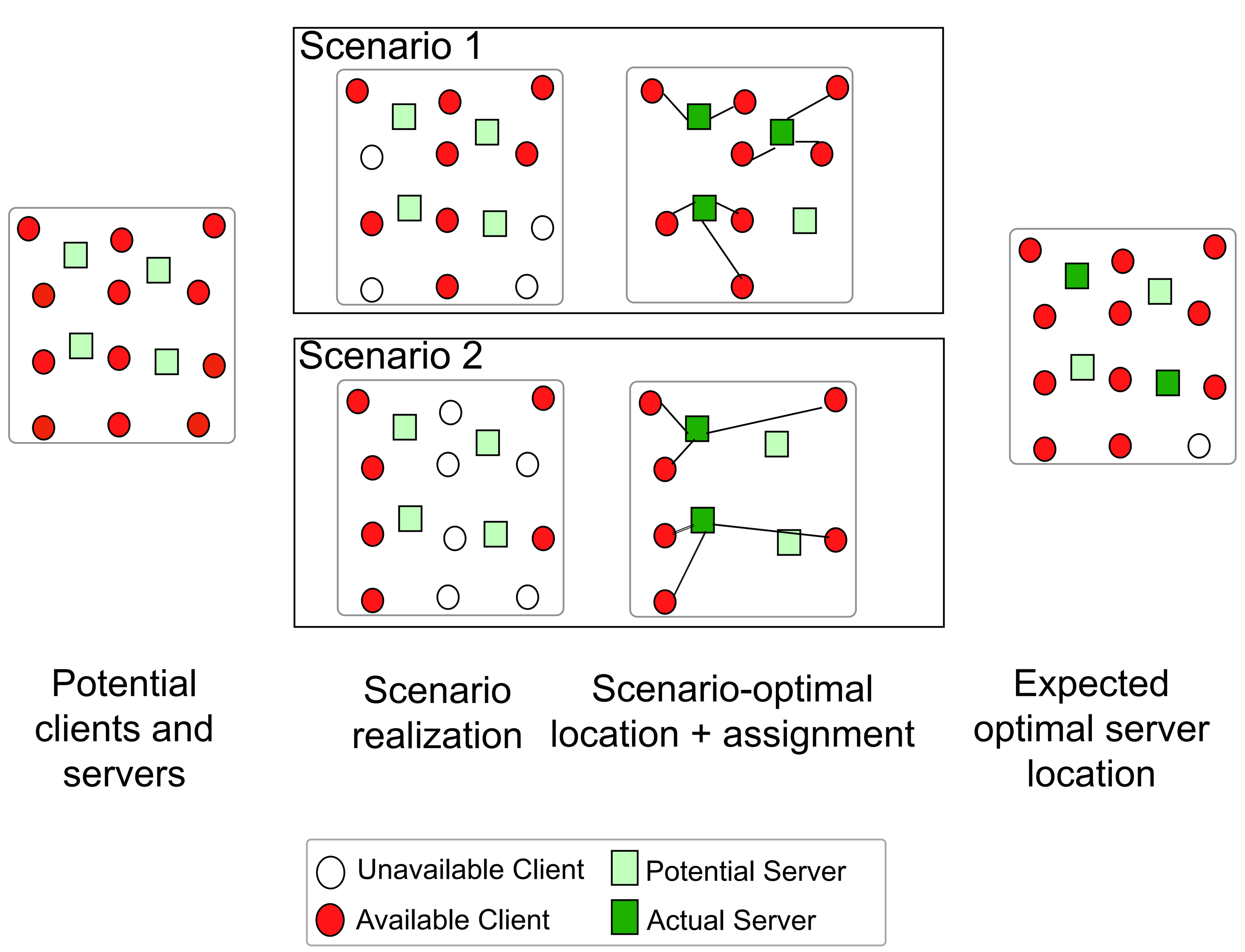
Second-Stage Problem

$$Q(\mathbf{x}, \xi_k) := \min_{\mathbf{y}} \left\{ \underbrace{\boxed{F(\mathbf{y}, \xi)}}_{\text{Second-stage cost}} : \mathbf{y} \in \mathcal{Y}(\mathbf{x}, \xi) \right\}$$

Second-stage decisions

Stochastic Server Location Problem

Objective: Determine the optimal set of servers to construct given uncertainty in client requests.



Potential clients and servers

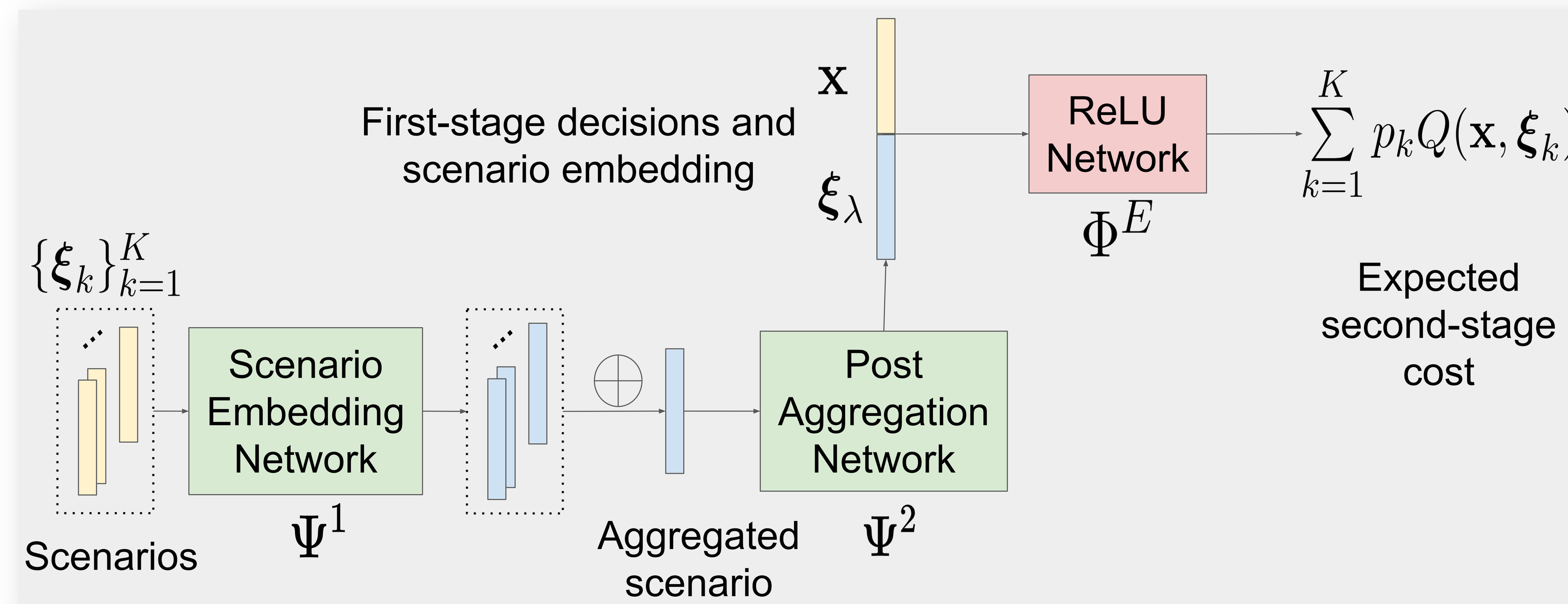
Scenario realization

Scenario-optimal location + assignment

Expected optimal server location

Neural Network Architecture

Can we predict the expected second-stage cost?
A set-based, permutation-invariant model can predict the expected second-stage cost for a variable number of scenarios.



Surrogate Optimization Model

Can we use the trained model to obtain a first-stage solution?

- The ReLU network can be embedded into an integer program.
- This formulation mitigates the curse of dimensionality from the number of scenarios.

$$\min_{\mathbf{x}} \quad \mathbf{c}^\intercal \mathbf{x} + \boxed{\beta} \quad \textbf{Predicted Objective}$$

NN constraints

$$\text{s.t.} \quad \sum_{i=1}^{d_0} w_{ij}^0 [\mathbf{x}, \xi_\lambda]_i + b_j^0 = \hat{h}_j^1 - \check{h}_j^1 \qquad \forall\, j \in [d_1],$$

$$\sum_{i=1}^{d_{m-1}} w_{ij}^{m-1} \hat{h}_i^{m-1} + b_j^{m-1} = \hat{h}_j^m - \check{h}_j^m \qquad \forall\, m \in [\ell - 1], j \in [d_m],$$

$$\sum_{i=1}^{d_\ell} w_{ij}^\ell \hat{h}_i^\ell + b^\ell \leq \beta$$

ReLU constraints

$$z_j^m = 1 \Rightarrow \hat{h}_j^m = 0 \qquad \forall\, m \in [\ell - 1], j \in [d_m],$$

$$z_j^m = 0 \Rightarrow \check{h}_j^m = 0 \qquad \forall\, m \in [\ell - 1], j \in [d_m],$$

Variables for NN

$$z_j^m \in \{0, 1\} \qquad \forall\, m \in [\ell - 1], j \in [d_m],$$

$$\hat{h}_j^m, \check{h}_j^m \geq 0 \qquad \underbrace{\forall\, m \in [\ell - 1], j \in [d_m],}_{}$$

$$\mathbf{x} \in \mathcal{X}$$

\# hidden layers, hidden layer dimension

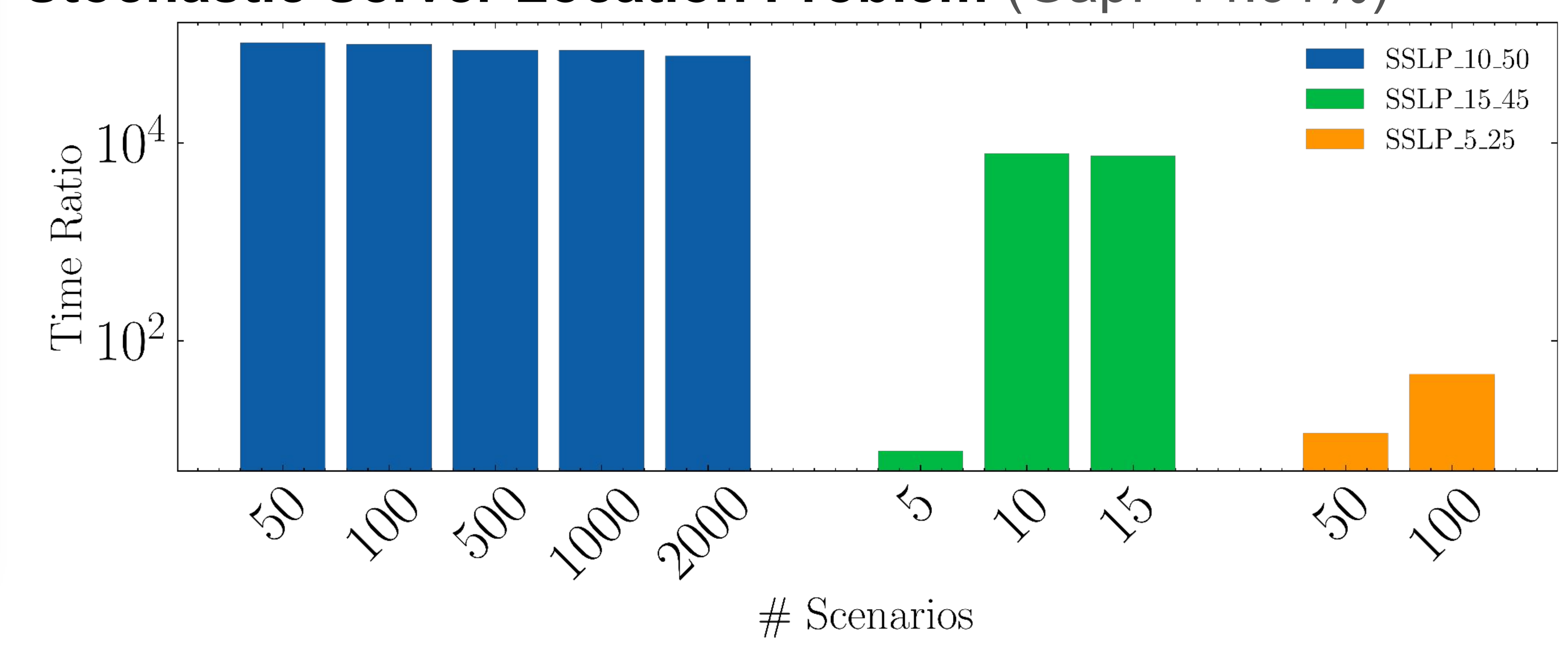
NEURAL INFORMATION PROCESSING SYSTEMS

Results

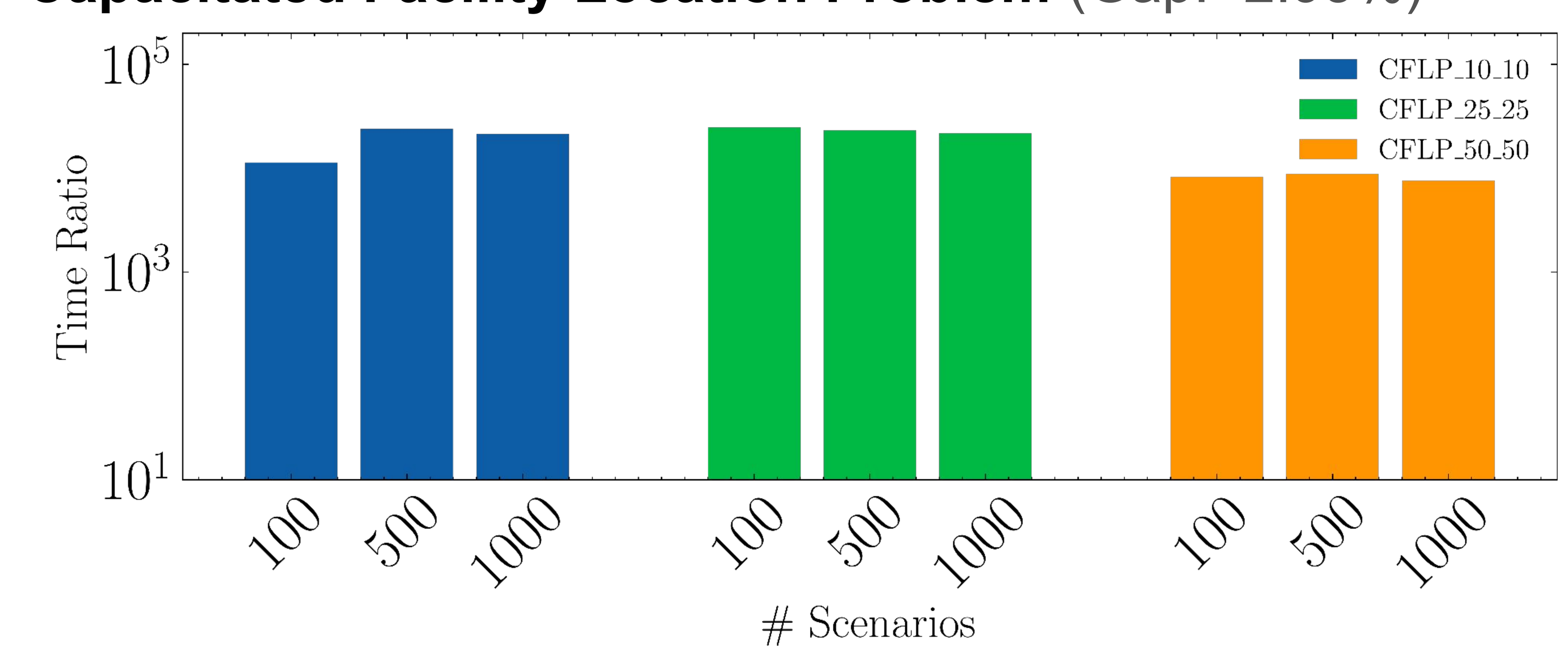
Gap: Mean % difference in solution quality relative to baseline (lower is better).

Bars: Reduction in computing time over baseline (higher is better).

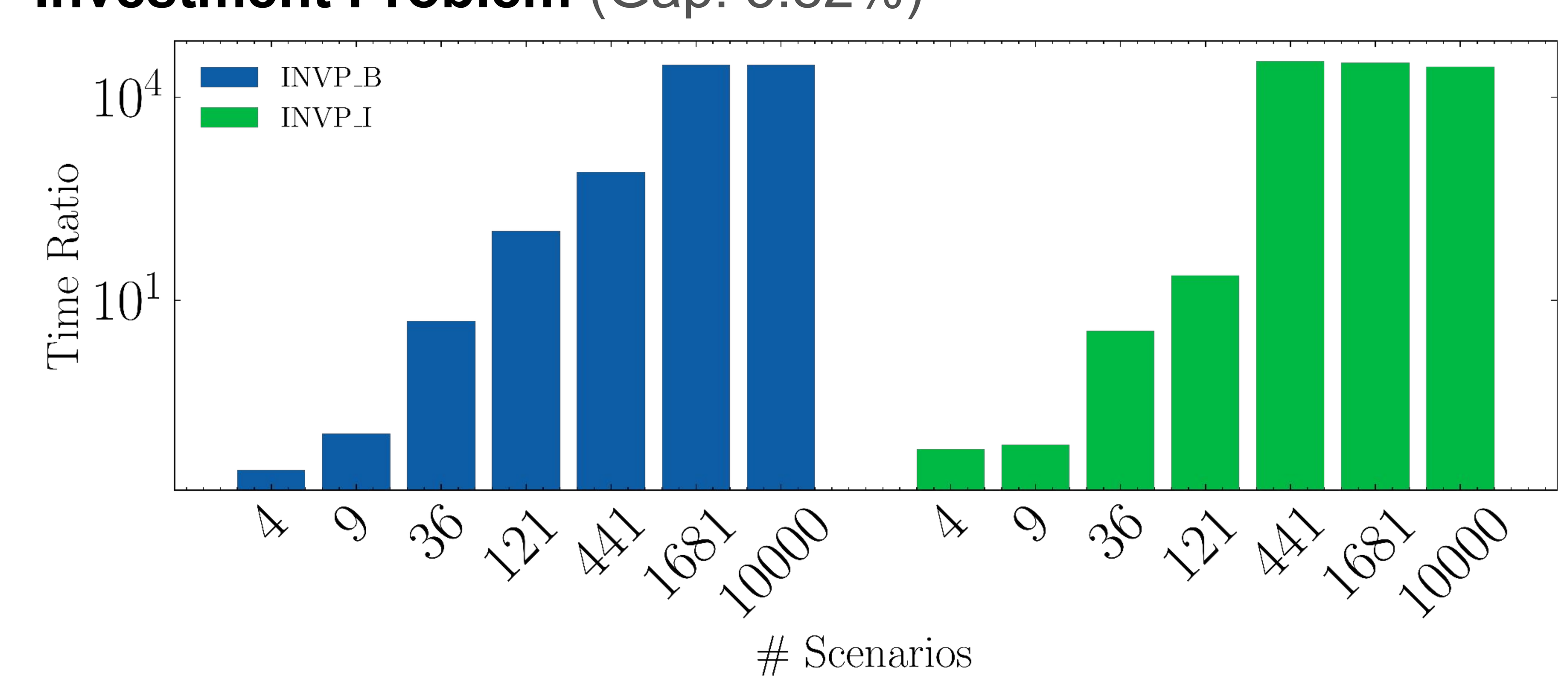
Stochastic Server Location Problem (Gap: -14.91%)



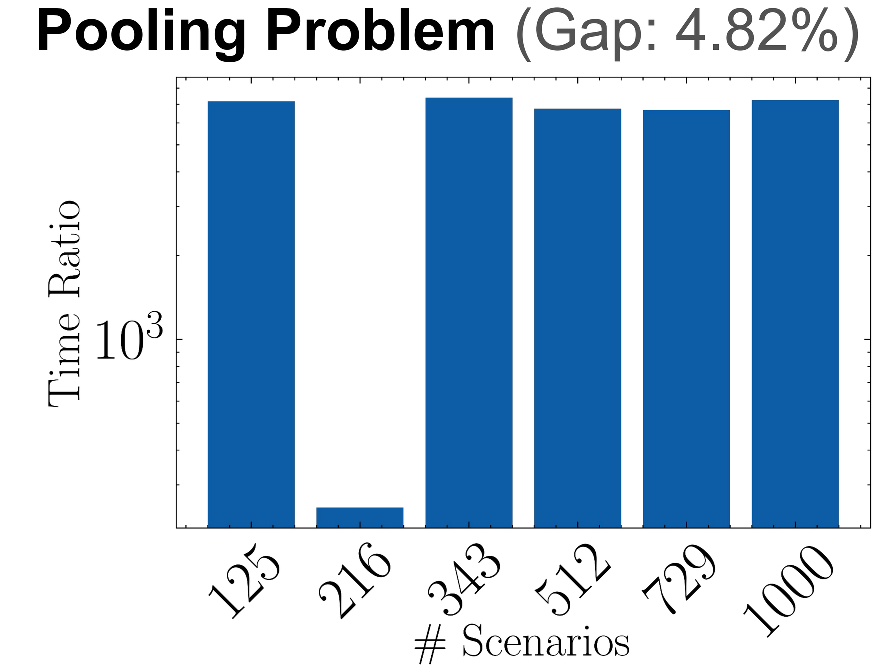
Capacitated Facility Location Problem (Gap: -2.93%)



Investment Problem (Gap: 3.82%)



Pooling Problem (Gap: 4.82%)



Project Page