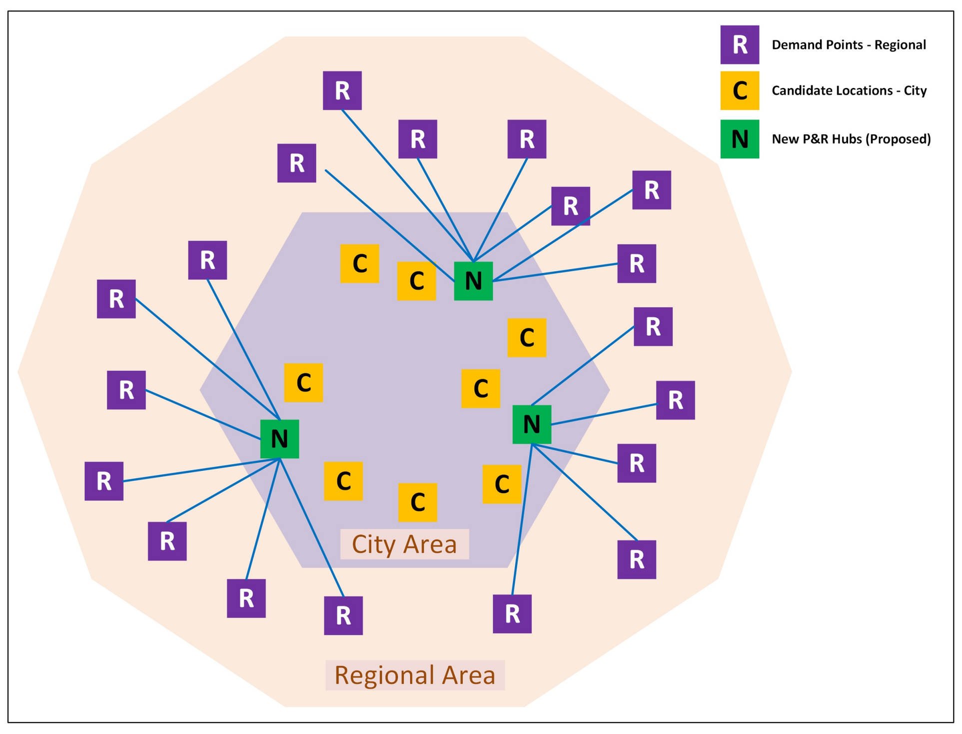
Multi-Agent System for Location of Park-and-Ride Hubs

Sathuta Sellapperuma , Algirdas Lančinskas,
DMSTI, Vilnius University
{sathuta.sellapperuma, algirdas.lancinskas}@mif.vu.lt

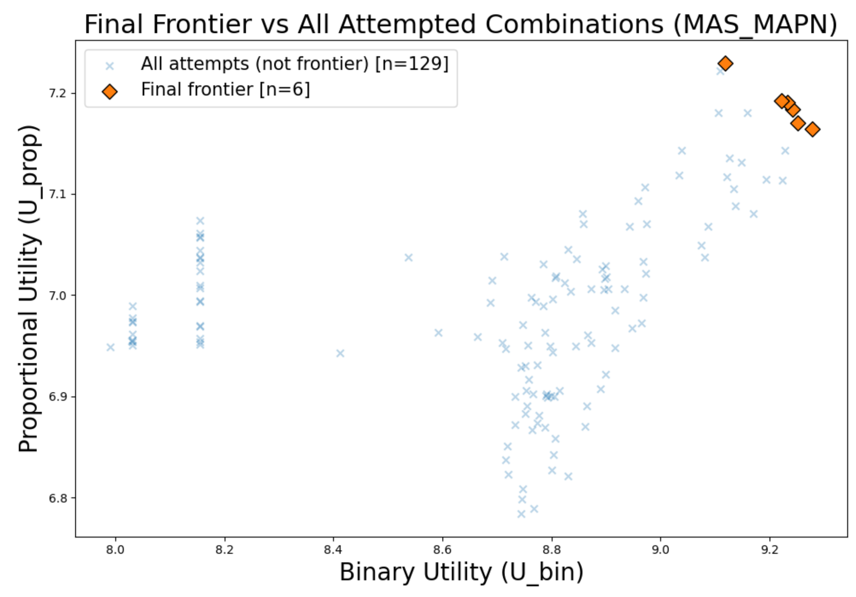
Introduction and Problem Statement

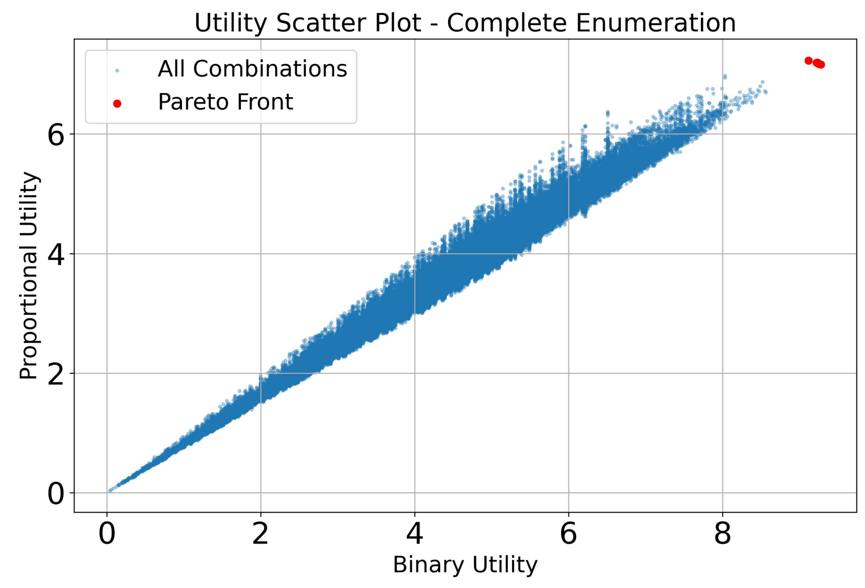
- **Goal:** Identify robust locations for new Park-and-Ride (P&R) hubs in urban networks.
- **Problem:** Customer choices may shift between models (uncertainty).
- **Objective:** Maximise both binary and proportional objectives and ensure solutions stay optimal under both behaviours.



Results and Validation

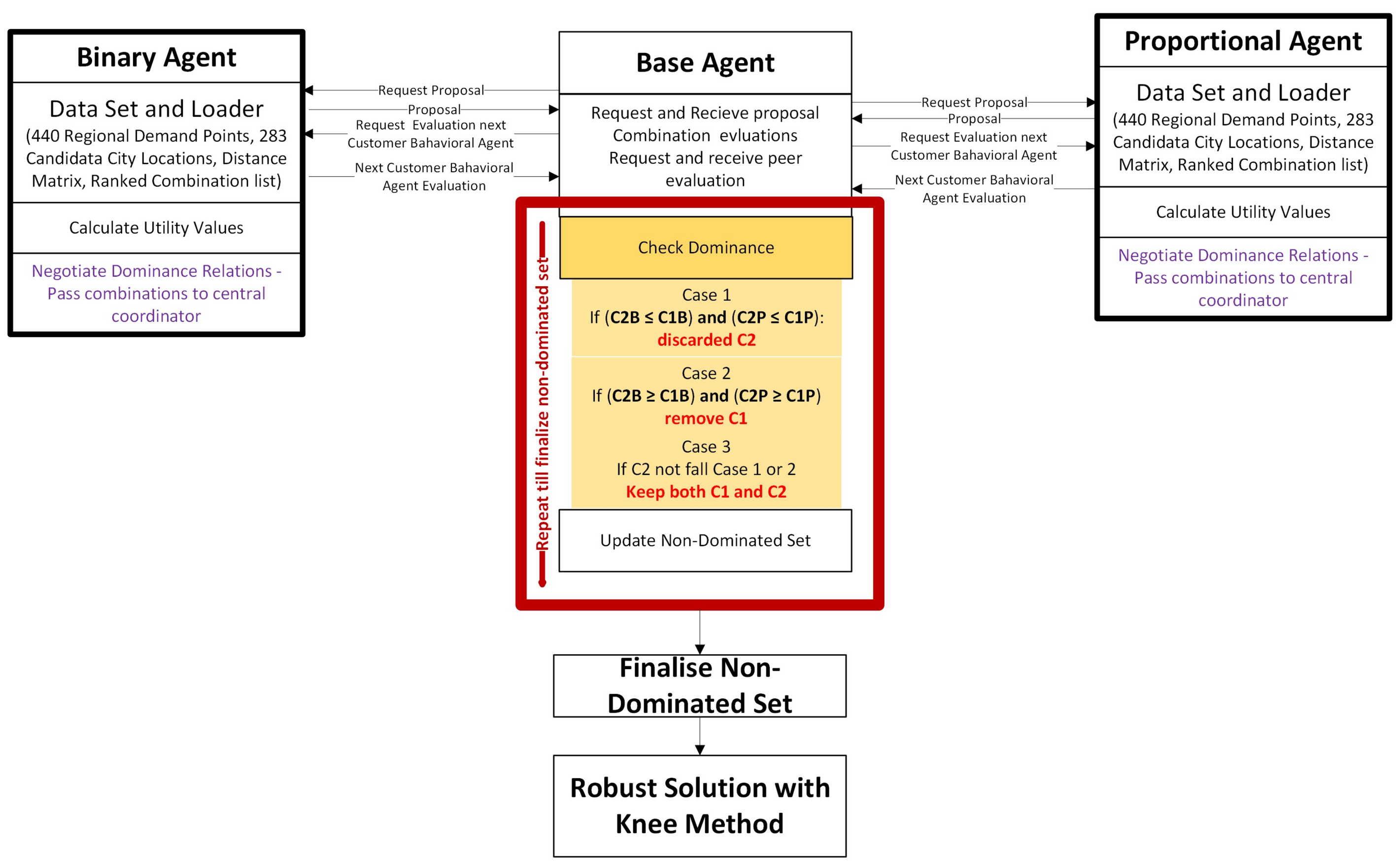
- Locations for 3 new P&R hubass, 6 test instances with different city qualities data sets.
- Demand points: 440 in Vilnius and 283 in the region of Vilnius.





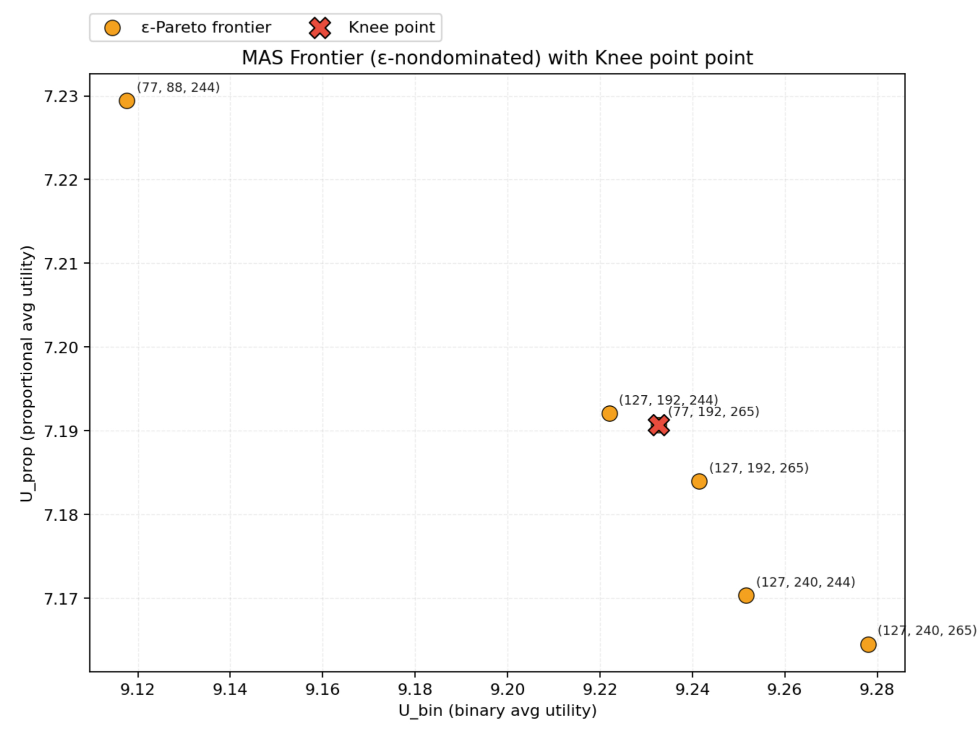
Multi-Agent System

- Multi-Agent System (MAS) uses Mediated Alternating-Proposal Negotiation (MAPN). Behavioral agents propose and evaluate solutions, while the base agent (mediator) manages the set of non-dominated solutions.
- Customer behavioral agents (Binary, Proportional) explores search space and propose solutions.
- Base agent manages entire process and handles communication between customer behavioral agents.



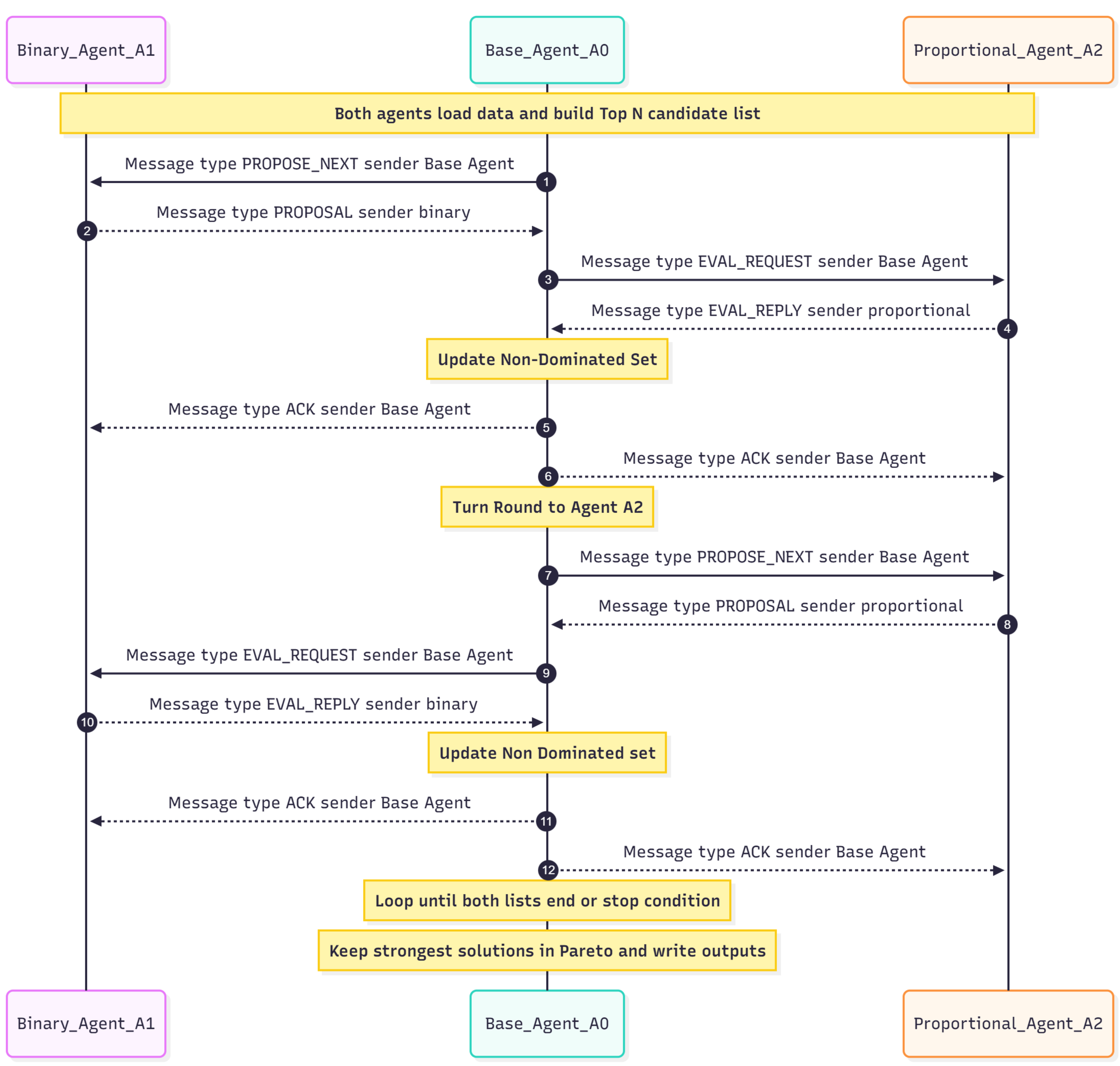
Robust Solution Selection

- After MAS negotiation, six **robust selection methods** were applied on the Pareto frontier:
  - *Distance-based* → Manhattan, Euclidean, (Chebyshev)
  - *Ranking-based* → TOPSIS, VIKOR
  - *Curvature-based* → **Knee Point**



Agent Communication

- **Lightweight FIPA-ACL** protocol with 6 fields: `performative`, `sender`, `receiver`, `content`, `reply_with`, and `in_reply_to`
- **Performative types**: `pn`, `pr`, `erq`, `erp`, `ack`, `nack`.



Agent Learning Logic

The Objective

- The agent must select locations for new facilities from a set S of location candidates.
- The objective is
  (1) to maximize the utility function $U(S)$ and
  (2) to rank candidate locations according to their fitness.

Ranking of Candidate Locations

- All candidate locations have rank values.
- Ranks represent probability to sample a candidate to form a solution.
- Ranks are automatically adjusted at runtime of the algorithm.

Sampling Candidate Locations

- Candidates are chosen one-by-one using a softmax probability with max shift:

$$P(i|S) = \frac{\exp(R_i - R_{max})}{\sum_{j \notin S} \exp(R_j - R_{max})}$$

R_i is the rank of i-th candidate location;
S is the set of candidates already selected;
$R_{max} = \max_{j \notin S} R_j$.

Reward and Advantage

- After constructing a full set S, the agent receives reward $Rwd = U(S)$.
- The advantage compares this reward to the baseline $Adv = Rwd - B$.
- The baseline is updated by $B \leftarrow (1-\beta) \cdot B + \beta \cdot Rwd$, where $\beta \in (0, 1]$.

Learning Update

- For each selected location i with sampling probability p_i the rank is updated by

$$R_i \leftarrow R_i + \alpha \cdot Adv(1 - p_i), \text{ where } \alpha \in (0, 1].$$

- Over time, the agent learns to prefer locations that improve utility.

Final Decision

- After the training phase, the agent constructs the final solution using a greedy rule:

$$S^* = \text{the set of } n \text{ candidates with the highest rank values.}$$

RCL
Research Council of Lithuania

This research has received funding from the Research Council of Lithuania (LMTLT), agreement No S-ITP-24-8
Project title: Robust Optimization for Green Infrastructure Design using Artificial Intelligence