Meta-Reinforcement Learning for Dynamic Parameter Adaptation in Multi-Round Vehicle Routing

Bojan Dimovski[1]

[1]Faculty of Electrical Engineering and Information Technologies

Outline

Outline

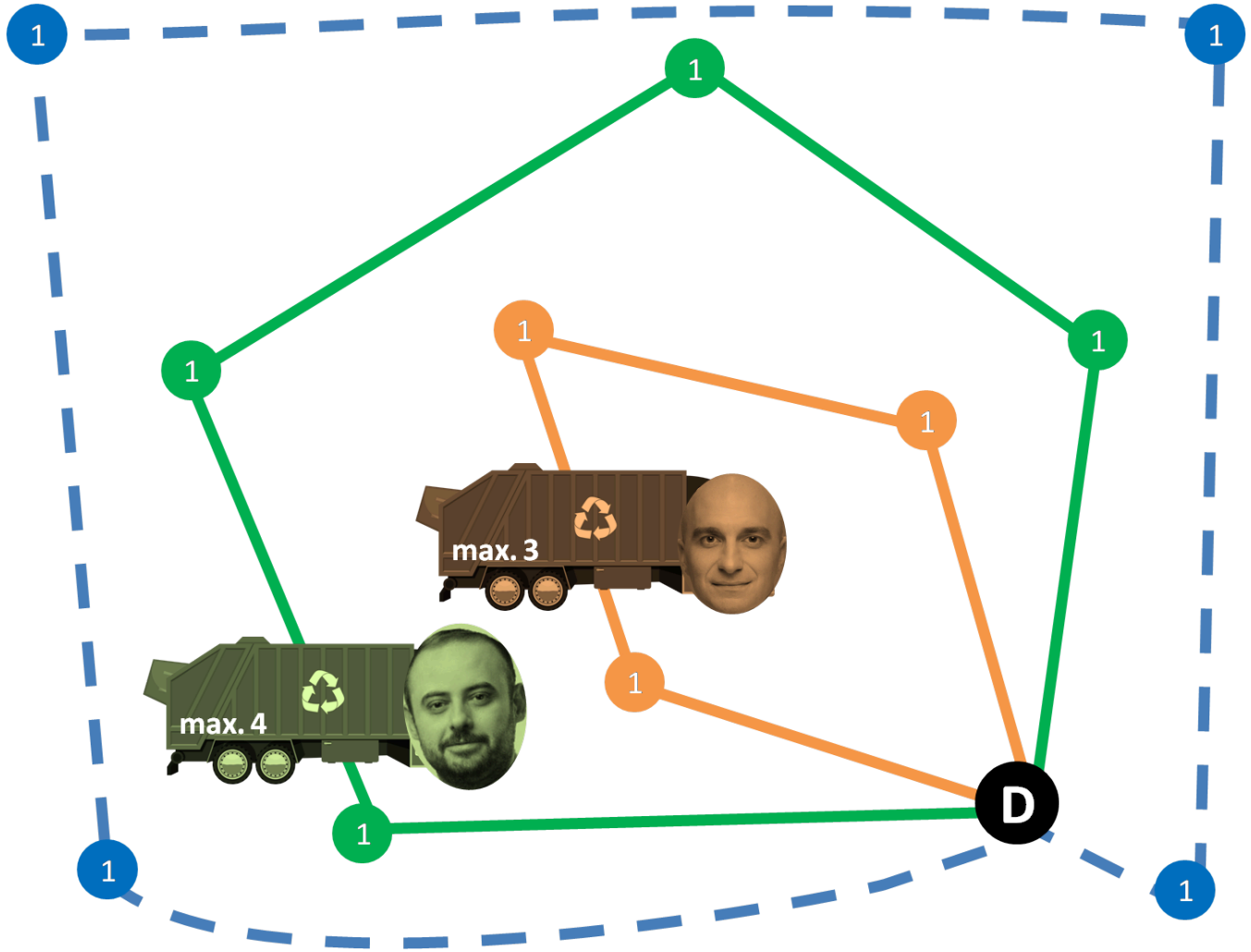
What is a multi-round VRP?

To establish the framework of this study, we first define the Static Capacitated Vehicle Routing Problem (SCVRP). We represent an instance as a septuple $\mathcal{P} = \langle V, E, c, d, Q, K, \mathcal{C} \rangle$, where:

- $V = \{v_0, v_1, ..., v_n\}$ is the set of vertices, where v_0 denotes the central depot and $V \setminus \{v_0\}$ represents the customers.
- $E \subseteq V \times V$ is the set of edges connecting vertices.
- $c : E \to \mathbb{R}^+$ defines the edge weights, representing travel costs or distances.
- $d : V \setminus \{v_0\} \to \mathbb{R}^+$ maps each customer to their specific demand.
- Q is the uniform capacity constraint for each vehicle.
- K is the number of available vehicles.
- $\mathcal{C}$ is the objective function to be minimized, typically representing total distance or operational cost.

Or, put simply...

What is a multi-round VRP?



Defining the metrics

- For scenarios in which the optimal solutions are **not** known, the goal is to minimize the cost (be it minimizing dispatch rounds, minimizing sum of the distances of all the routes, some function combining multiple constraints etc.)
- For Augerat et al. Set A instances, the optimal solutions are known
- For testing purposes, the fitness landscape is normalized according to known optimal solutions.

$$\text{gap\_cost}(\sigma) = \text{gc}(\sigma) = \left(\frac{C(\sigma) - C^*}{C^*}\right) \cdot 100$$

$$\text{gap\_dispatch}(\sigma) = \text{gd}(\sigma) = \left(\frac{R(\sigma) - R^*}{R^*}\right) \cdot 100$$

$$\text{penalty}(\sigma) = [\#\text{violations}] \cdot \text{penalty\_factor}$$

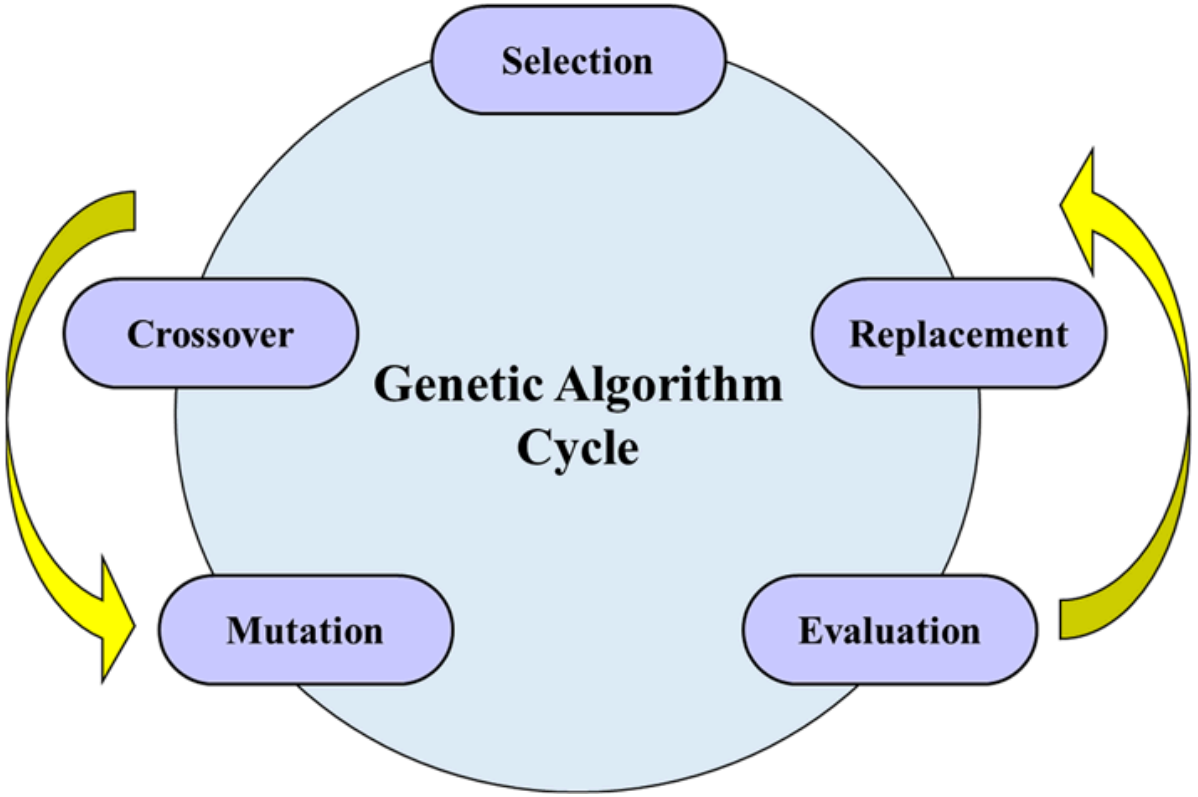
$$\text{fit}(\sigma) = 100 - [w_c\ \text{gc}(\sigma) + (1 - w_c)\text{gd}(\sigma) + \text{p}(\sigma)]$$

Where $w_c \in (0, 1)$ is the weight of the relative cost gap in the final fitness function. For our purposes, we have defined $w_c = 0.7$. Defined as such, $\text{fit}(\sigma)$ can have values in the range $(-\infty, 100]$. Since it is a metric that can be unintuitive, an analogy can be made with $R^2\%$.

Defining the metrics

- If $\text{fit}(\sigma) = 100$, then σ is the optimal solution, just like $R^2 = 100\%$ represents a perfect fit of the curve.
- If $\text{fit}(\sigma) \in (0, 100)$ it means that σ is suboptimal, but the solution is not naive.
- If $\text{fit}(\sigma) = 0$, it means the solution σ leads to a solution which costs twice as much as the cost of the optimal solution and the vehicles travel twice as many dispatch rounds, i.e. σ is twice as worse as the optimal solution σ^*, i.e. a naive solution analogous to $R^2 = 0$ which would be tantamount to a dummy regressor predicting the mean for any input.
- If $\text{fit}(\sigma) < 0$ it means the solution σ is not feasible or is so bad that it performs worse than the naive solution which is twice as worse, analogous to $R^2 < 0$ meaning a regressor performing worse than a naive dummy regressor with a mean strategy.

Framework for Vanilla GA



Custom Mutation Types (Part 1)

Inversion (40%)

[1, 2, 3, 4, 5, 6, 7, 8]

[1, 2, 6, 5, 4, 3, 7, 8]

Custom Mutation Types (Part 2)

Inversion (40%)

[1, 2, 3, 4, 5, 6, 7, 8]

[1, 2, 6, 5, 4, 3, 7, 8]

Swap (30%)

[1, 2, 3, 4, 5, 6, 7, 8]

[1, 6, 3, 4, 5, 2, 7, 8]

Custom Mutation Types (Part 3)

Inversion (40%)

[1, 2, 3, 4, 5, 6, 7, 8]

[1, 2, 6, 5, 4, 3, 7, 8]

Swap (30%)

[1, 2, 3, 4, 5, 6, 7, 8]

[1, 6, 3, 4, 5, 2, 7, 8]

Scramble (20%)

[1, 2, 3, 4, 5, 6, 7, 8]

[1, 2, 4, 6, 3, 5, 7, 8]

Custom Mutation Types (Part 4)

Inversion (40%)

[1, 2, 3, 4, 5, 6, 7, 8]

[1, 2, 6, 5, 4, 3, 7, 8]

Swap (30%)

[1, 2, 3, 4, 5, 6, 7, 8]

[1, 6, 3, 4, 5, 2, 7, 8]

Scramble (20%)

[1, 2, 3, 4, 5, 6, 7, 8]

[1, 2, 4, 6, 3, 5, 7, 8]

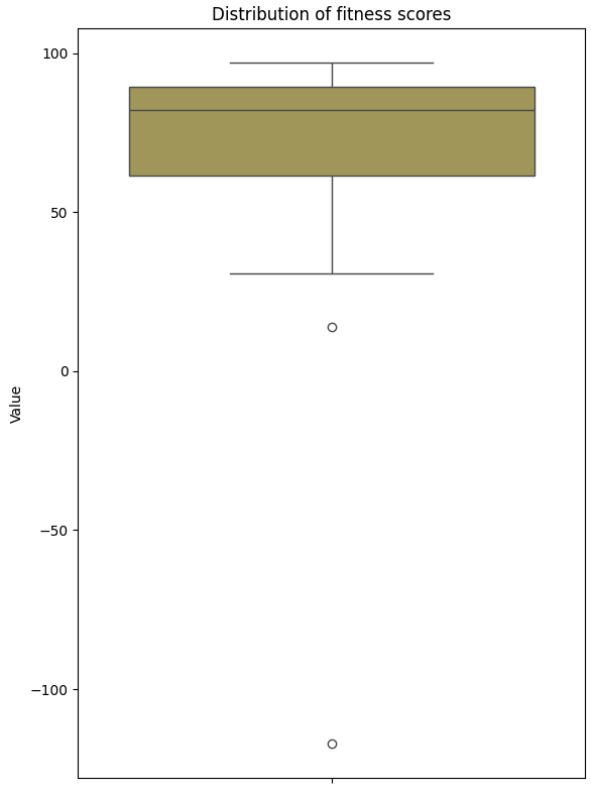
Displace (10%)

[1, 2, 3, 4, 5, 6, 7, 8]

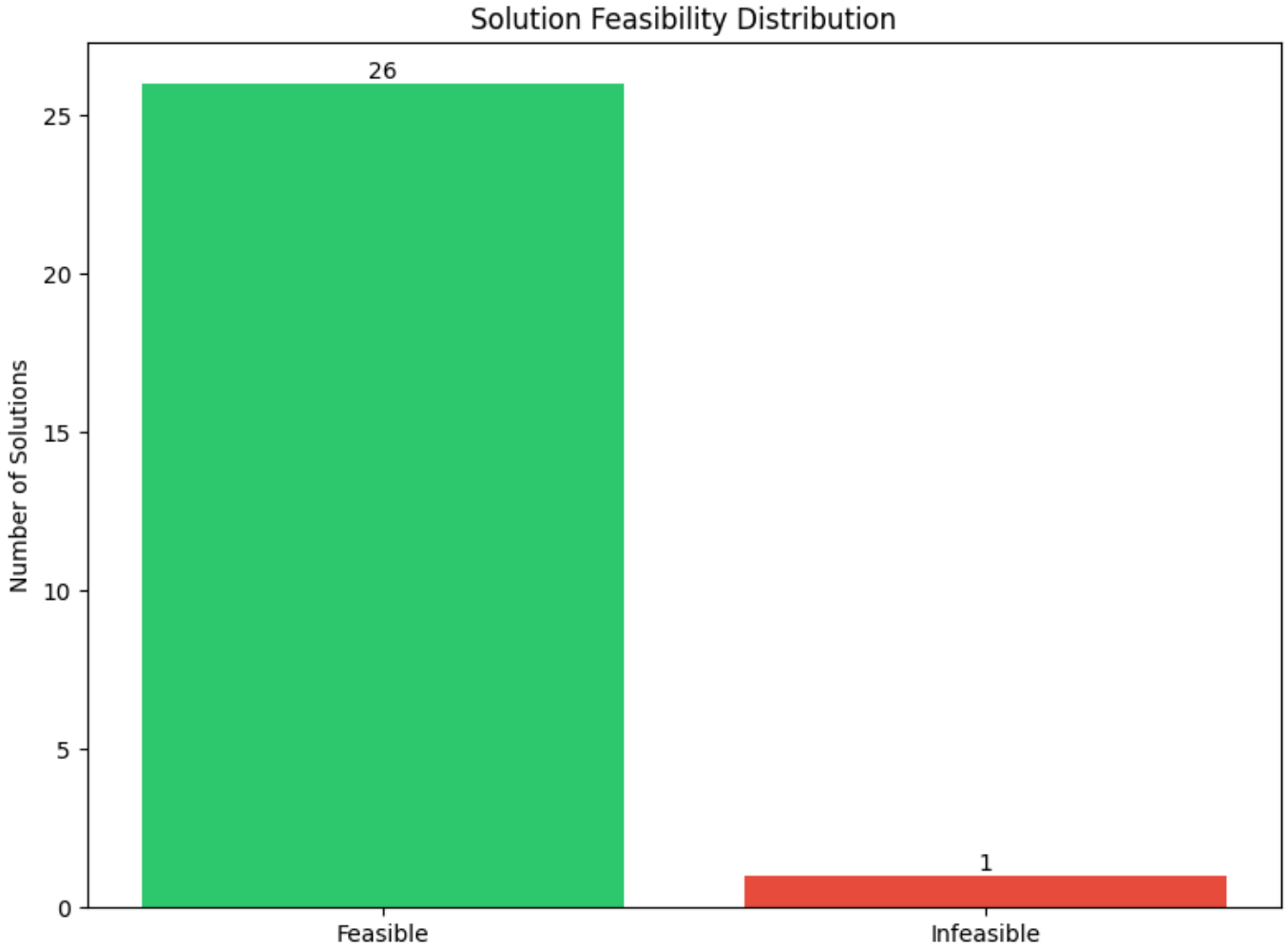
[1, 2, 6, 7, 3, 4, 5, 8]

Vanilla GA performance

- Median = better representation of overall performance of Vanilla GA than mean
- Median = 82.02, Mean = 67.09

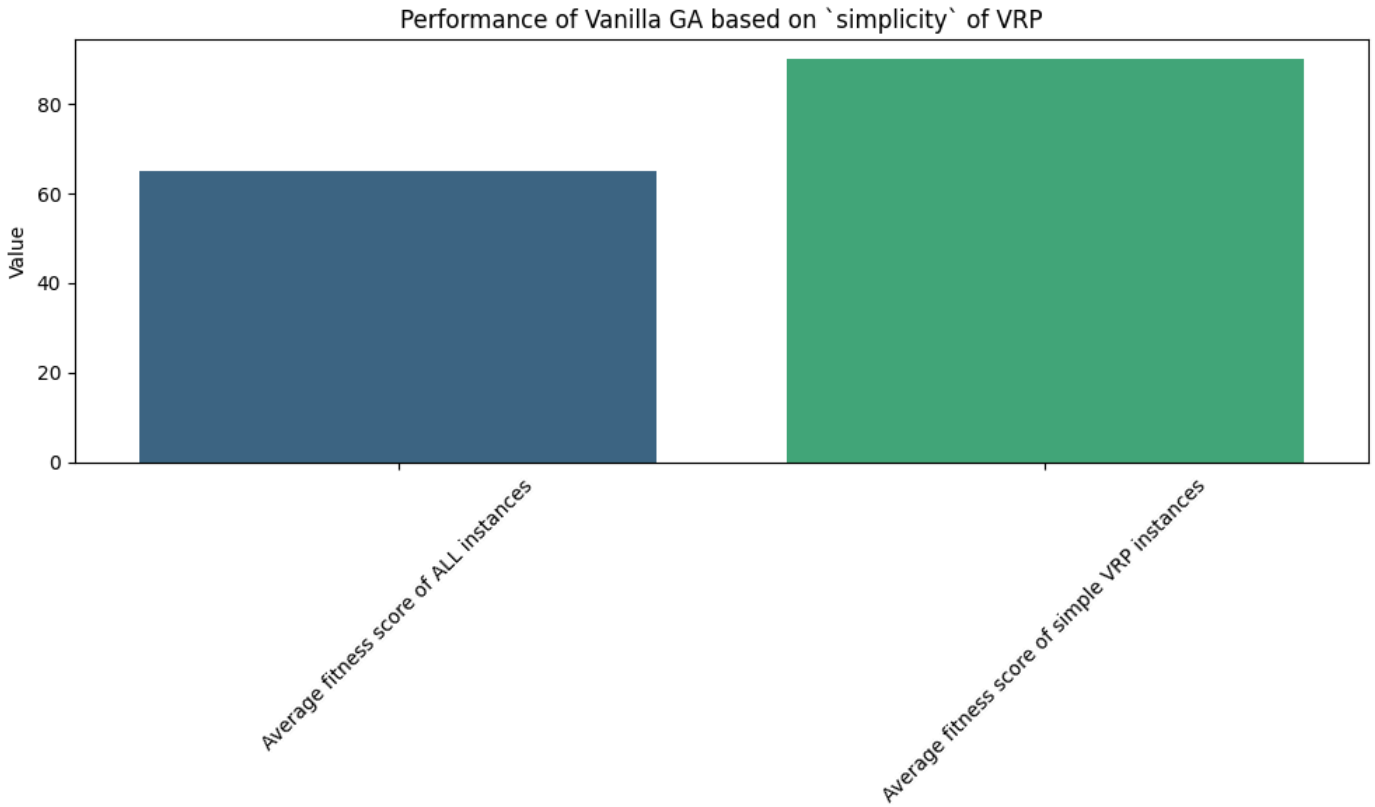


Vanilla GA performance



Vanilla GA performs quite well for low-dimensional VRPs!!!

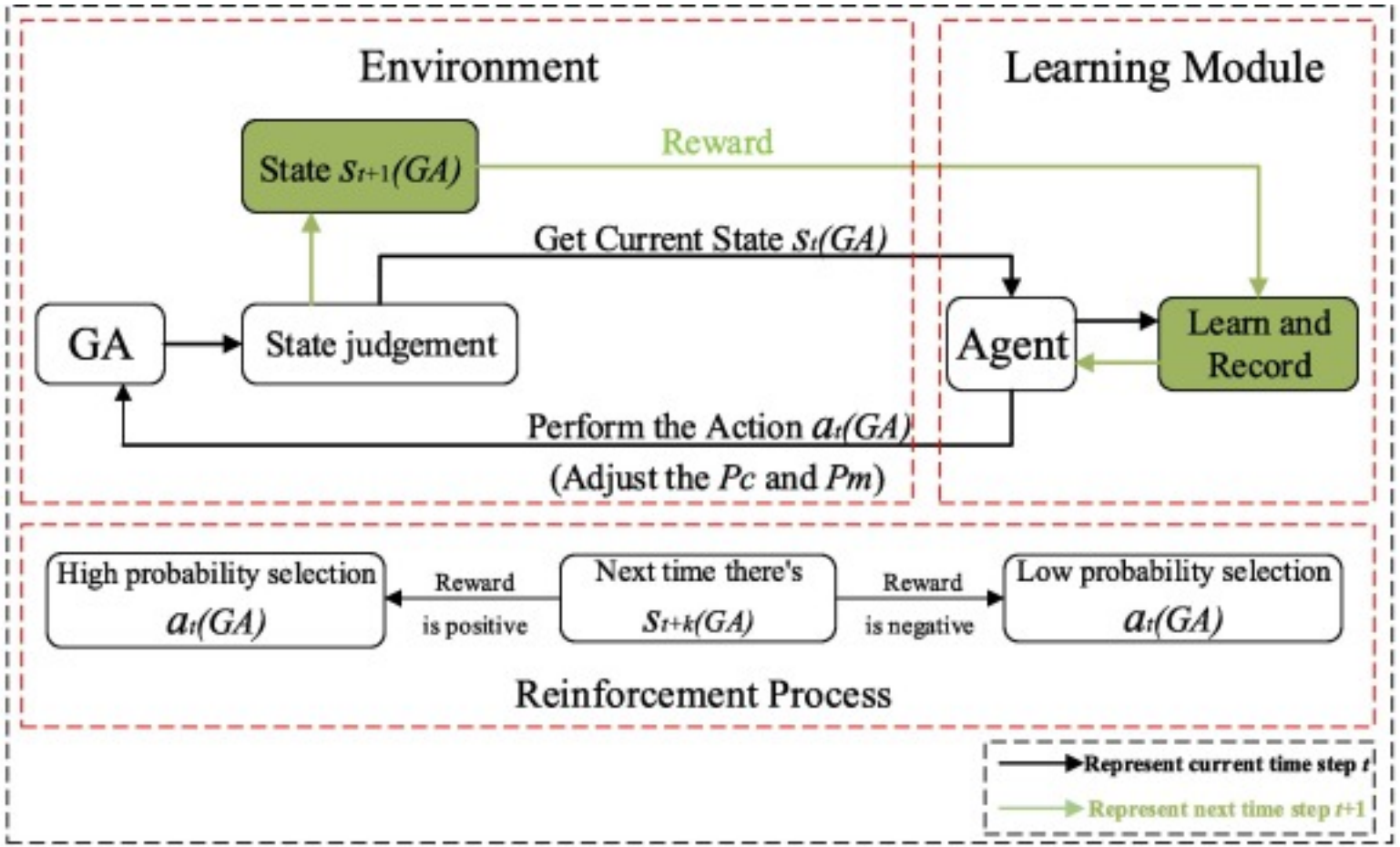
- Median best-found fitness score for **all** instances: 82.02
- Median best-found fitness score for **simple** instances: 91.14



Reinforcement Learning Agent as a meta-strategist

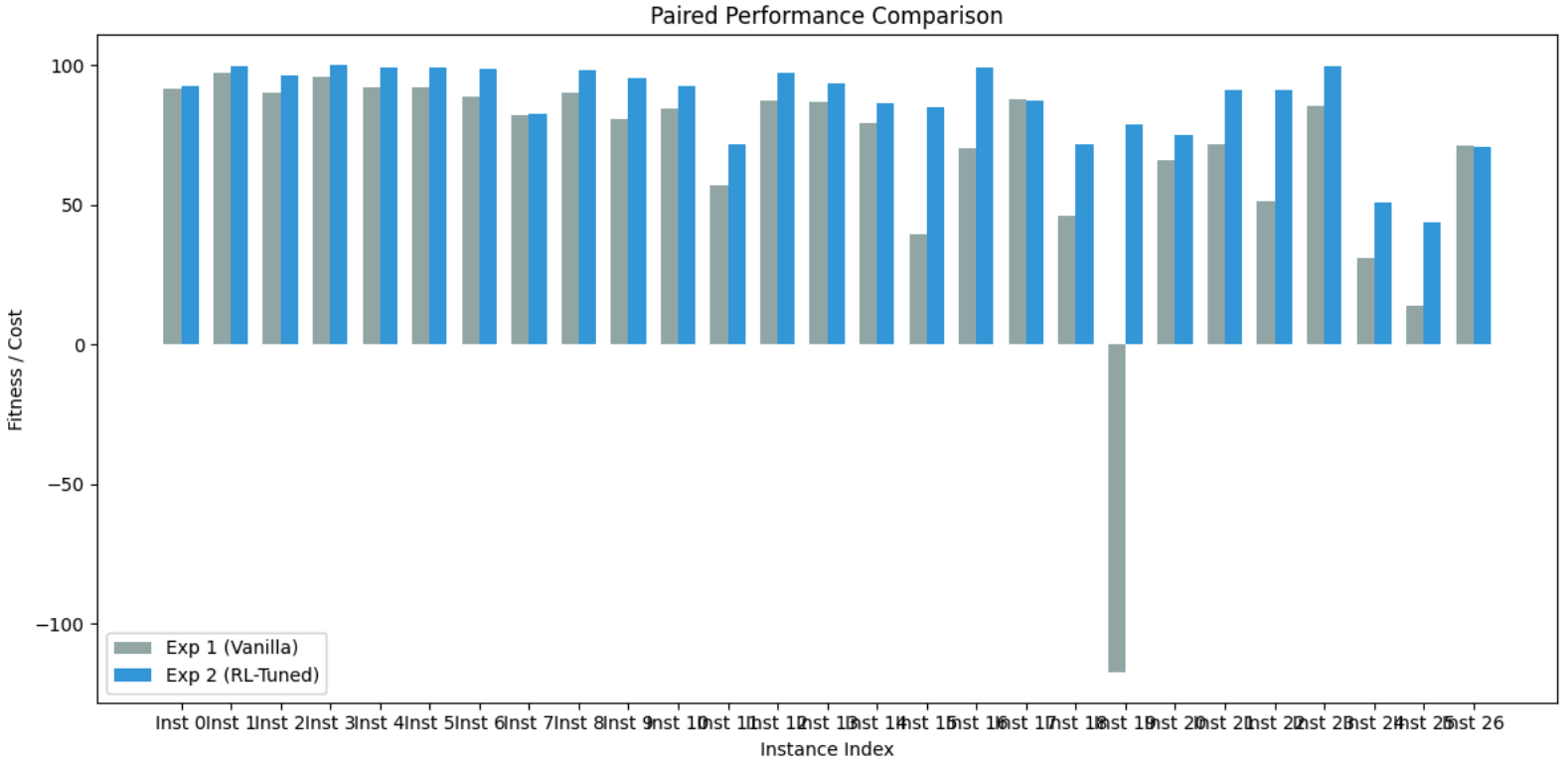
- Uses Stable Baselines3 (SB3) to implement a PPO agent as a high-level meta-optimizer.
- Agent monitors population state, diversity, and stagnation to guide the GA.
- Interventions occur at fixed intervals defined by `n_check_gen` to minimize computational overhead.
- Periodically adjusts mutation probabilities and crossover rates based on real-time performance.
- Balances exploration and exploitation to prevent local optima and speed up convergence.
- Standardizes the optimization process across different VRP instances without manual tuning.

Reinforcement Learning Agent as a meta-strategist

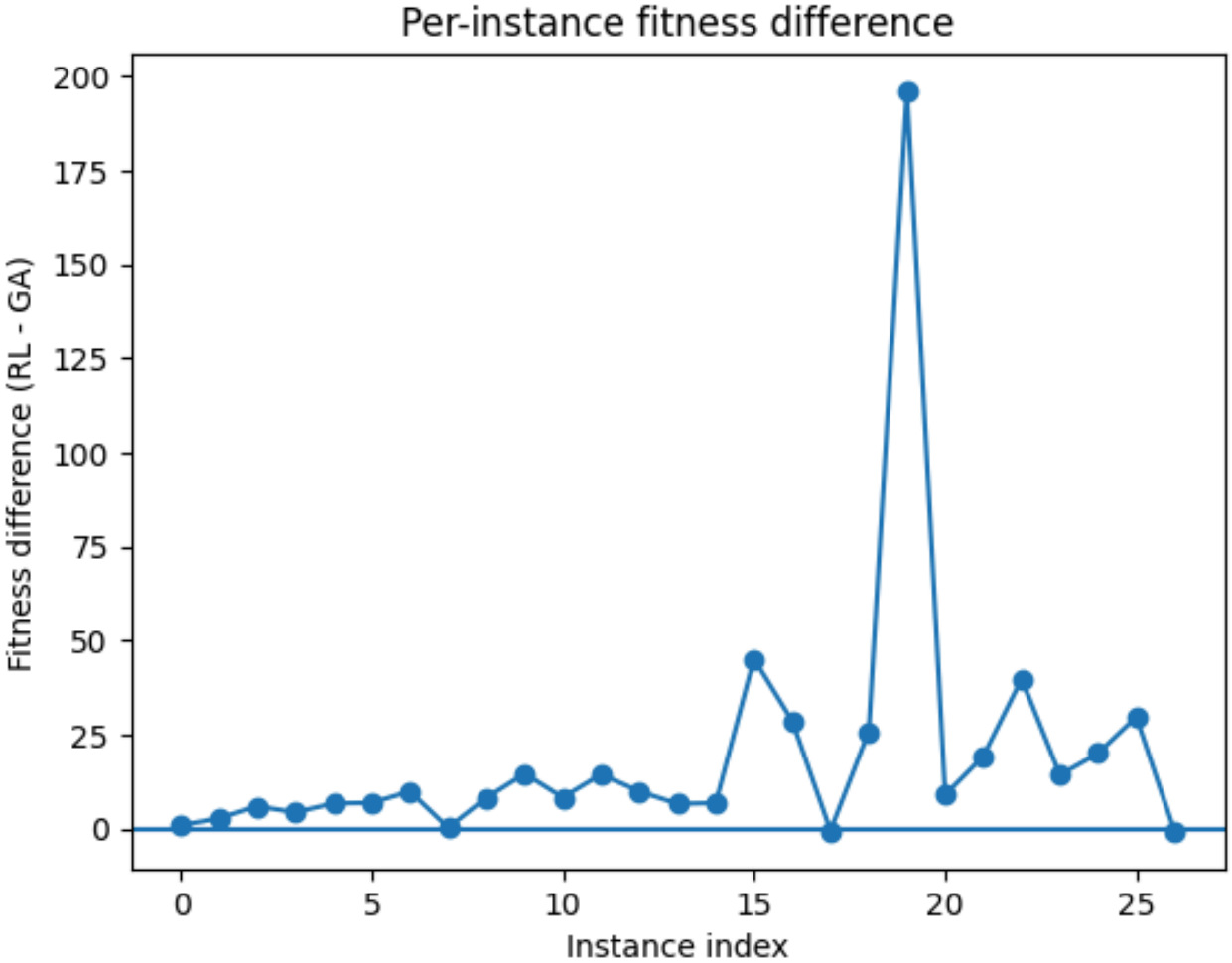


Results & Comparison

- RL-powered adaptive GA outperforms Vanilla GA in 25/27 instances
- RL-powered adaptive GA has **0** infeasible solutions



Results & Comparison

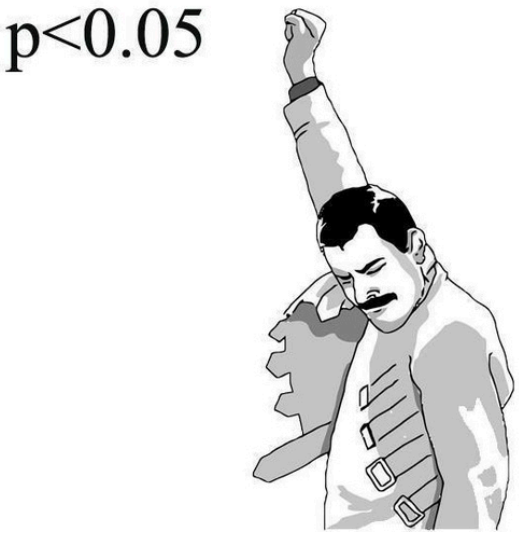


Test for statistical significance

- Paired instances
- Nonparametric (not normally distributed)

$\Rightarrow$ Wilcoxon signed-rank test

$p = 1.49 \cdot 10^{-7} < \alpha := 0.01 \Rightarrow$ DIFFERENCE IS STATISTICALLY SIGNIFICANT



However...

Concluding remarks

Concluding remarks

- RL agents achieve significant gains but the training overhead (e.g., even a light-weight model of 10,000 steps) is often prohibitive. (took me 2 days to train😔)
- Complexity of PPO and neural networks can lead to generalization issues in high-dimensional instances. (e.g. `A-n80-k10`)
- Resource intensity is difficult to justify when compared to a well-tuned, lightweight Vanilla GA.
- A simple rule-based Adaptive GA can replicate RL benefits without the training time.
- Heuristic triggers (e.g., boosting mutation when diversity < 0.05) offer transparent, real-time adaptation.
- Transitioning to deterministic rules preserves flexibility while ensuring industrial accessibility.

Any questions?