ML-Enhanced Large Neighborhood Search Project Report

Machine Learning for Optimization, WS 2025

Daniel Levin (12433760)

1 Problem: SCF-PDP

The **Selective Capacitated Fair Pickup and Delivery Problem (SCF-PDP)** is a vehicle routing problem where the goal is to design fair and feasible routes for a subset of customer requests. Each customer needs transportation of goods from a pickup location to a corresponding drop-off location.

1.1 Problem Formulation

The problem is modeled on a complete directed graph $G = (V, A)$ where:

- V contains the vehicle depot plus all pickup/drop-off locations
- $A = \{(u, v) : u, v \in V, u \neq v\}$ represents travel routes
- Arc distances $a_{u,v} = \lceil \sqrt{(x_u - x_v)^2 + (y_u - y_v)^2} \rceil$ (rounded Euclidean)

There are n customer requests $CR = \{1, \ldots, n\}$, each defined by pickup location $v_i^{\uparrow}$ and drop-off location $v_i^{\downarrow}$ with demand c_i. A fleet K of n_K identical vehicles with capacity C serves requests starting from the depot.

1.2 Constraints and Objective

A feasible solution assigns routes R_k to each vehicle $k \in K$ such that:

- Vehicle capacity never exceeded along any route
- Each served request handled entirely by one vehicle
- At least γ requests served across all vehicles

The objective minimizes total distance plus fairness penalty:

$$\sum_{k \in K} d(R_k) + \rho \cdot (1 - J(R))$$

where $d(R_k)$ is route k's total distance and $J(R)$ is the Jain fairness index:

$$J(R) = \frac{\left(\sum_{k \in K} d(R_k)\right)^2}{n_K \cdot \sum_{k \in K} d(R_k)^2}$$

The fairness index $J(R) \in (0, 1]$ equals 1 when all routes have equal distance. Parameter ρ controls the distance-fairness trade-off.

2 Baseline: Tuned ALNS

The baseline implementation uses Adaptive Large Neighborhood Search (ALNS) with simulated annealing acceptance, developed and tuned in the Heuristic Optimization Techniques course. This section summarizes the key components.

2.1 Operators

The implementation uses 3 destroy and 3 repair operators, yielding 9 combinations. Each destroy operator removes q requests sampled uniformly from $[10\%, 40\%]$ of served requests.

Destroy Operators:

- **Random**: Removes q randomly selected requests (unbiased exploration)
- **WorstCost**: Removes requests with highest distance contribution
- **LongestRoute**: Removes from longest routes (targets fairness)

Repair Operators:

- **Greedy**: Reinserts using flexible pickup-dropoff construction heuristic
- **RandomGreedy**: Reinserts at random feasible positions
- **ObjectiveAware**: Minimizes full objective including fairness penalty

2.2 Adaptive Weight Mechanism

Operators are selected via roulette wheel with adaptive weights. Scores: 10 for new best, 1 for accepted, 0 for rejected. Weights update every 100 iterations using:

$$\rho_i \leftarrow \rho_i \cdot (1 - \gamma) + \gamma \cdot \frac{s_i}{a_i}$$

where γ is the reaction factor, s_i total score, and a_i applications in the period.

2.3 Parameters

Parameter	Description	Default
`max_iterations`	Maximum iterations	10000
`max_time_seconds`	Runtime limit	300s
`weight_update_period`	Weight update frequency	100
`reaction_factor` (γ)	Adaptation speed	0.1
`min/max_removal_pct`	Removal range	10%-40%
`initial_temperature`	SA start temperature	100.0
`cooling_rate`	Temperature decay	0.99

Table 1: ALNS default parameters

3 Multi-Armed Bandit Operator Selection

3.1 Method

3.2 Results

4 Reinforcement Learning Operator Selection

4.1 Method

4.2 Results

5 Comparison and Analysis

6 Conclusion