Transportation & Logistics

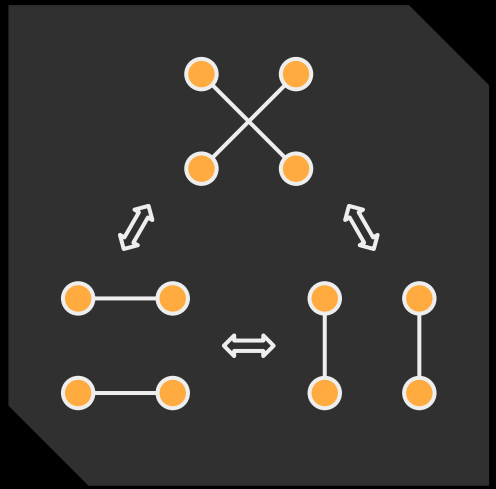
Alex Varga
Mark Lavrentyev

Simulated Annealing

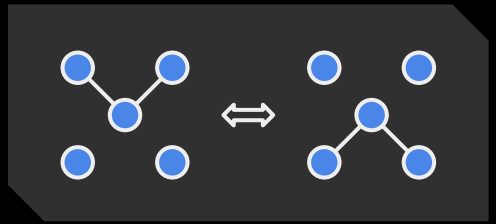
- **Simulated Annealing:**
  - Very quick annealing rate to discourage random steps later (when close)
  - Objective value: distance + penalty for capacity overflow

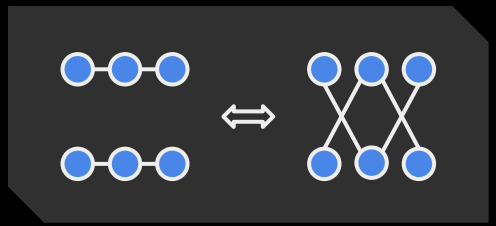
- **Stopping criteria:**
  - Made 10 bad steps in a row
  - Current state is feasible
  - Low variation in the past 1000 steps (less than 0.001% of score)
- Idea: stop when we aren’t improving any more

Neighborhoods



2-Opt





Compound

- Two ways to cross a pair of edges on different routes
- Must be sure not to create a loop if edges are in same route

- Compound operations can be defined in terms of two 2-opt operations
  - Useful later in search

- Solver learns which operations are likely to improve the objective value as it goes

Basic Optimizations & Tunable Parameters

- Used profiler to find slow points
- Memoized capacity, distance functions

- Used multiple processes (1/core)
  - First one done is the answer

Parameters we can tune:

- Learning rate, min weight for each op
- Annealing time, multiplier, and init value
- History size, threshold for stopping

- Capacity overflow multiplier for penalty
  - Initially high to force feasibility
  - Decreases when close to allow more exploration later

Potential Future Work

- Use state-distance metric to avoid over-exploring similar states and spread out search

$$d_S(s_1, s_2) := \sum_{p \in P} \min \Big(\|s_1(p) - s_2(p)\| + \|s_1^{-1}(p) - s_2^{-1}(p)\|, \|s_1(p) - s_2^{-1}(p)\| + \|s_1^{-1}(p) - s_2(p)\| \Big)$$

- Implement larger neighborhoods
  - ***n*-opt**: permutations on *n* edges
  - ***k*-compound *n*-opt**: combinations of k *n*-opt operations
- Other learning techniques for deciding which neighborhood op to use

Summary

- Approach: Simulated annealing local search over full (feas. + infeas.) space
  - Non-linear penalty for infeasibility reliant on capacity overflow

- Neighborhoods: 2-opt, compound 2-opt
  - Much faster solution: 20s vs. 150s on largest instance

- Stopping criteria: keep history of recent past scores
    - If they don’t change much, stop
    - Good at recognizing when done