Capacitated Vehicle Routing Problem

Richard Tang & Cristina Hu

Approach 1: ALNS

Initial Solution

$$\begin{aligned} s(i,j) &= 2d(D,i) + 2d(D,j) - [d(D,i) + d(i,j) + d(D,j)] \\ &= d(D,i) + d(D,j) - d(i,j) \end{aligned} \tag{6.12}$$

- Generate initial (*potentially infeasible*) solution
- Attempt 1: Greedy NN
  - Unideal in many cases: crossovers, some are better served by other cars
  - It's very much "local", not taking into account the broader context
- Attempt 2: Clarke-Wright Savings Algorithm
  - Algorithm
    - Assume each customer is served by one vehicle
    - Join customers by the highest saving that can be achieved until capacity is reached
  - Tends to not include the points that are the most isolated
  - Tried enhancements to incorporate demand (Altinel, Öncan 2005) (Doyuran, Çatay 2011), but no noticeable improvement
- Both cases often left high-demand customers unserved that were "isolated"

ALNS

- Destroy & Repair Operators
  - Switching two customers from different routes (one random, the other “closest”)
  - Switch “segments” of customers
  - Random removal - best global insertion
  - 2 opt
- Operator selection
  - Simulated Annealing
  - autofit() temperature
  - Multi-Arm Bandit: GreedyEpsilon (1-epsilon choose best, epsilon explore), each arm is one of the operators
- Accept ➔ when / what criteria to use to accept a new alternative
  - 20%, 10%, 5% ➔ how much worse it can be
  - 50%,25%,10% ➔ how probable it is that we would accept

Approach 2:
Genetic Search

Local Search

- Initial solution
  - Bellman-based Split algorithm (Prins 2004)
  - Potentially infeasible, but fast
  - Penalize excess demand dynamically
- Local Search
  - Randomly apply operators on routes until no improvement
  - Operators:
    - 2-opt / 2-opt*
    - Swap
    - Relocate

Genetic Algorithm

- Inspired by (Vidal 2021)
- Population management
  - Maintain feasible + infeasible populations
    - To encourage exploration, maintain 20% feasible by adjusting excess demand penalty
  - Compute “diversity” by route’s broken-pairs distance
  - Crossover using parts of each parent, then re-compute routes w/ Split algorithm (Oliver et. al. 1987)
- Genetic algorithm
  - Generate initial population via Split + LS
  - Repeatedly crossover and optimize new offspring
    - Parallelize offspring optimization
  - When a new generation (λ offspring) produced, select elite (top μ) offspring by diversity + objective

$$f_{\mathcal{P}}(S) = f_{\mathcal{P}}^{\phi}(S) + \left(1 - \frac{n^{\text{ELITE}}}{|\mathcal{P}|}\right) f_{\mathcal{P}}^{\text{DIV}}(S).$$

Final Approach: Portfolio!

Portfolio “Algorithm”

- Concurrently run ALNS and Genetic Search
- Select best result