Optimization and Simulation

Optimization lab 1: VRPTW construction heuristics

Iliya Markov

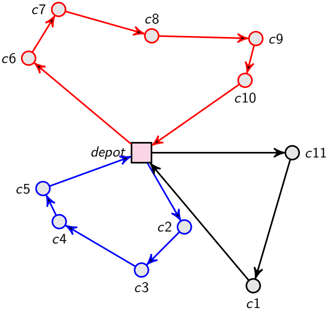
Transport and Mobility Laboratory
School of Architecture, Civil and Environmental Engineering
École Polytechnique Fédérale de Lausanne

April 12, 2016

Overview

Overview

The Vehicle Routing Problem (VRP)



- The Vehicle Routing Problem (VRP) is a combinatorial optimization and integer programming problem that seeks to find the most efficient utilization and routing of a vehicle fleet to service a set of customers subject to constraints.
- It was introduced by Dantzig and Ramser (1959), and is one of the most practically relevant and widely studied problems in Operations Research.
- It has numerous applications in the distribution and collection of goods and the transportation of people.

Overview

The capacitated VRP

- Three-index directed vehicle-flow formulation (Golden et al., 1977):
- Sets:
  - $\mathcal{K}$ is a set of identical vehicles
  - $\mathcal{N}$ is a set of all nodes, where the depot is duplicated as o (origin) and d (destination)
- Parameters:
  - Q is the vehicle capacity
  - q_i is the demand at node i
  - c_{ij} is the travel cost from node i to j
- Variables:
  - $x_{ijk} = 1$ iff vehicle k moves from node i to j; 0 otherwise
  - $y_{ik} = 1$ iff vehicle k visits node i; 0 otherwise
  - u_{ik} is the cumulated demand serviced by vehicle k when arriving at node i

The capacitated VRP

- Objective: minimize total travel cost

$$\text{minimize} \quad \sum_{k \in \mathcal{K}} \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{N}} c_{ij} x_{ijk}$$

The capacitated VRP

- Objective: minimize total travel cost

$$\text{minimize} \quad \sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{N}}\sum_{j\in\mathcal{N}} c_{ij}x_{ijk}$$

- A customer is visited by exactly one vehicle

$$\text{s.t.} \quad \sum_{k\in\mathcal{K}} y_{ik} = 1, \qquad \forall i \in \mathcal{N} \setminus \{o, d\}$$

The capacitated VRP

- Objective: minimize total travel cost

$$\text{minimize} \quad \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}} c_{ij} x_{ijk}$$

- A customer is visited by exactly one vehicle

$$\text{s.t.} \quad \sum_{k\in\mathcal{K}} y_{ik} = 1, \qquad \forall i \in \mathcal{N} \setminus \{o, d\}$$

- Path-flow

$$\text{s.t.} \quad \sum_{j\in\mathcal{N}\setminus\{i\}} x_{ijk} - \sum_{j\in\mathcal{N}\setminus\{i\}} x_{jik} = 0, \qquad \forall i \in \mathcal{N} \setminus \{o, d\}, k \in \mathcal{K}$$

$$\text{s.t.} \quad \sum_{j\in\mathcal{N}\setminus\{o\}} x_{ojk} - \sum_{j\in\mathcal{N}\setminus\{o\}} x_{jok} = 1, \qquad \forall k \in \mathcal{K}$$

The capacitated VRP

- Coupling

$$\text{s.t.} \qquad y_{ik} = \sum_{j \in \mathcal{N} \setminus \{i\}} x_{ijk}, \qquad \forall i \in \mathcal{N} \setminus \{d\}, k \in \mathcal{K}$$

$$\text{s.t.} \qquad y_{dk} = \sum_{i \in \mathcal{N} \setminus \{d\}} x_{idk}, \qquad \forall k \in \mathcal{K}$$

The capacitated VRP

- Coupling

s.t. $$y_{ik} = \sum_{j \in \mathcal{N} \setminus \{i\}} x_{ijk}, \qquad \forall i \in \mathcal{N} \setminus \{d\}, k \in \mathcal{K}$$

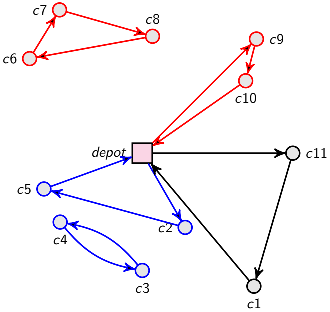
s.t. $$y_{dk} = \sum_{i \in \mathcal{N} \setminus \{d\}} x_{idk}, \qquad \forall k \in \mathcal{K}$$

- Domain

s.t. $$x_{ijk} \in \{0, 1\}, \qquad \forall i, j \in \mathcal{N}, k \in \mathcal{K}$$

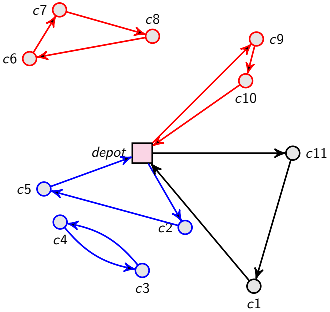
s.t. $$y_{ik} \in \{0, 1\}, \qquad \forall i \in \mathcal{N}, k \in \mathcal{K}$$

We are missing something...



- Look at the solution depicted here.
- The cycles $c6 \longrightarrow c7 \longrightarrow c8 \longrightarrow c6$ and $c3 \longrightarrow c4 \longrightarrow c3$ are referred to as *subtours*.
- Subtours are part of the vehicles' tours that are disconnected from the depot.
- Apparently a solution like this should not exist.

We are missing something...



- Look at the solution depicted here.
- The cycles $c6 \longrightarrow c7 \longrightarrow c8 \longrightarrow c6$ and $c3 \longrightarrow c4 \longrightarrow c3$ are referred to as *subtours*.
- Subtours are part of the vehicles' tours that are disconnected from the depot.
- Apparently a solution like this should not exist.
- **However, is it feasible for the model defined above!**

Subtour elimination constraints

- The constraints we are missing are called *subtour elimination constraints* (SEC).
- Their role is to eliminate the possibility of subtours and to enforce the vehicle capacity constraints.
- SEC can be formulated in different ways, with an impact on the number of SEC and the integrality gap, **and they represent the main difficulty in solving the VRP**.

Subtour elimination constraints

- The constraints we are missing are called *subtour elimination constraints* (SEC).
- Their role is to eliminate the possibility of subtours and to enforce the vehicle capacity constraints.
- SEC can be formulated in different ways, with an impact on the number of SEC and the integrality gap, **and they represent the main difficulty in solving the VRP**.
- One classical example of a SEC formulation, the so-called MTZ-formulation, is due to Miller et al. (1960). It has $\mathcal{O}(n^2)$ variables and constraints but produces a weak linear relaxation of the model.

$$
\begin{array}{lll}
\text{s.t.} & u_{ik} + q_j \leq u_{jk} + Q(1 - x_{ijk}), & \forall i, j \in \mathcal{N}, k \in \mathcal{K} \\
\text{s.t.} & q_i \leq u_{ik} \leq Q, & \forall i \in \mathcal{N}, k \in \mathcal{K}
\end{array}
$$

Complete model

$$\min \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{N}} \sum_{j\in\mathcal{N}} c_{ij} x_{ijk}$$

$$\text{s.t.} \sum_{k\in\mathcal{K}} y_{ik} = 1, \qquad \forall i \in \mathcal{N} \setminus \{o, d\}$$

$$\sum_{j\in\mathcal{N}\setminus\{i\}} x_{ijk} - \sum_{j\in\mathcal{N}\setminus\{i\}} x_{jik} = 0, \qquad \forall i \in \mathcal{N} \setminus \{o, d\}, k \in \mathcal{K}$$

$$\sum_{j\in\mathcal{N}\setminus\{o\}} x_{ojk} - \sum_{j\in\mathcal{N}\setminus\{o\}} x_{jok} = 1, \qquad \forall k \in \mathcal{K}$$

$$y_{ik} = \sum_{j\in\mathcal{N}\setminus\{i\}} x_{ijk}, \qquad \forall i \in \mathcal{N} \setminus \{d\}, k \in \mathcal{K}$$

$$y_{dk} = \sum_{i\in\mathcal{N}\setminus\{d\}} x_{idk}, \qquad \forall k \in \mathcal{K}$$

$$u_{ik} + q_j \leq u_{jk} + Q(1 - x_{ijk}), \qquad \forall i, j \in \mathcal{N}, k \in \mathcal{K}$$

$$q_i \leq u_{ik} \leq Q, \qquad \forall i \in \mathcal{N}, k \in \mathcal{K}$$

$$x_{ijk} \in \{0, 1\}, \qquad \forall i, j \in \mathcal{N}, k \in \mathcal{K}$$

$$y_{ik} \in \{0, 1\}, \qquad \forall i \in \mathcal{N}, k \in \mathcal{K}$$

Overview

Adding time windows (VRPTW)

- Time windows are constraints that appear very often in practice.
- They introduce a time dimension to the problem and restrict the start-of-service time at node i between a_i and b_i.

Adding time windows (VRPTW)

- Time windows are constraints that appear very often in practice.
- They introduce a time dimension to the problem and restrict the start-of-service time at node i between a_i and b_i.
- Let t_{ij} (parameter) denote the travel time from node i to node j and T_{ik} (decision variable) denote the start-of-service time for vehicle k at node i. Then:

$$\begin{array}{lll} \text{s.t.} & a_i \leq T_{ik} \leq b_i & \forall i \in \mathcal{N}, k \in \mathcal{K} \\ \text{s.t.} & T_{ik} + t_{ij} \leq T_{ij} + M(1 - x_{ijk}) & \forall i \in \mathcal{N}, k \in \mathcal{K} \end{array}$$

- If a vehicle arrives at node i before a_i, it waits until the time window opens to start service. However, it cannot arrive after b_i.

Adding time windows (VRPTW)

- Time windows are constraints that appear very often in practice.
- They introduce a time dimension to the problem and restrict the start-of-service time at node i between a_i and b_i.

- Let t_{ij} (parameter) denote the travel time from node i to node j and T_{ik} (decision variable) denote the start-of-service time for vehicle k at node i. Then:

$$\begin{array}{lll} \text{s.t.} & a_i \leq T_{ik} \leq b_i & \forall i \in \mathcal{N}, k \in \mathcal{K} \\ \text{s.t.} & T_{ik} + t_{ij} \leq T_{ij} + M(1 - x_{ijk}) & \forall i \in \mathcal{N}, k \in \mathcal{K} \end{array}$$

- If a vehicle arrives at node i before a_i, it waits until the time window opens to start service. However, it cannot arrive after b_i.

- Based on our policy, we may want to change the objective function to include a duration aspect.

Overview

Solving the VRPTW

- Solution methods for the VRP in general and the VRPTW in particular can broadly be classified into exact, heuristic and hybrid.
- There are two main categories of exact methods: branch-and-cut and branch-and-price, both with many variations.
- Heuristic methods for the VRP started with construction and improvement procedures, and later evolved into more complex metaheuristics such as tabu search, genetic algorithms, variable neighborhood search, adaptive large neighborhood search, etc.
- They are often combined with exact methods for subproblems.

Overview

Exercise for the first lab

- Solomon (1987) designed the first systematic procedures for building good quality initial VRPTW solutions. His paper is available here.
- He also built a testbed of instances which is now one of the classical VRP benchmark sets.
- Your task consists of:
  - Reading the paper (very short).
  - Implementing the first insertion heuristic (Section 1.3).
  - Testing it on Solomon's benchmark instances, which you can find on the website. There is also a help file explaining the structure of the instance files.
  - Comparing your results to Solomon's.

Bibliography

Dantzig, G. and Ramser, R. (1959). The truck dispatching problem. *Management Science*, 6:80–91.

Golden, B. L., Magnanti, T. L., and Nguyen, H. Q. (1977). Implementing vehicle routing algorithms. *Networks*, 7(2):113–148.

Miller, D. L., Tucker, A. W., and Zemlin, R. A. (1960). Integer programming formulations of traveling salesman problems. *Journal of the ACM*, (7):326–329.

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2):254–265.