The Vehicle Routing Problem

Decision-aid Methodologies in Transportation: Computer Lab 11

Iliya Markov

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

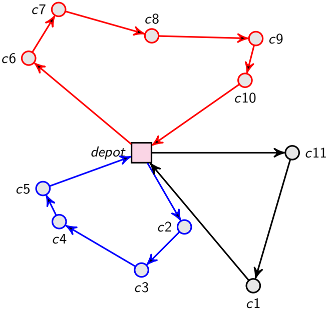
May 5, 2015

Overview

Overview

Introduction

Introduction



- 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

Formulating the capacitated VRP (CVRP)

- We present the three-index directed vehicle-flow formulation of the VRP, modified from Golden, Magnanti and Nguyen (1977). For it, we need to define the following:
- Sets:
  - K is a set of identical vehicles
  - N is a set of all nodes, where the depot is represented by two nodes, o and d, for the start and end point of each tour
- 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

Formulating the capacitated VRP (CVRP)

- Objective: minimize total travel cost

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

Formulating the capacitated VRP (CVRP)

- Objective: minimize total travel cost

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

- A customer is visited by exactly one vehicle

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

Formulating the capacitated VRP (CVRP)

- Objective: minimize total travel cost

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

- A customer is visited by exactly one vehicle

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

- Path-flow

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

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

Formulating the capacitated VRP (CVRP)

- Coupling

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

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

Formulating the capacitated VRP (CVRP)

- Coupling

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

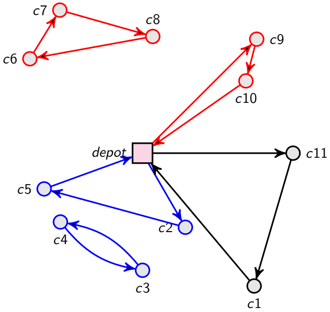
s.t. $$y_{dk} = \sum_{i \in N \setminus \{d\}} x_{idk}, \qquad \forall k \in K$$

- Domain

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

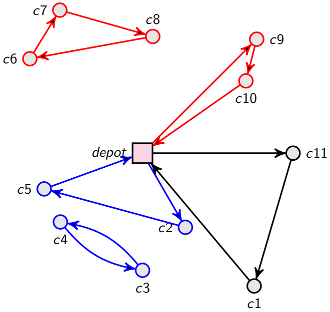
s.t. $$y_{ik} \in \{0, 1\}, \qquad \forall i \in N, k \in K$$

Let's stop and think



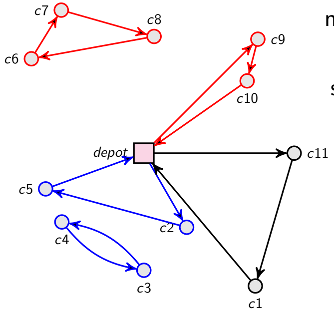
- Look at the solution depicted here.
- The cycles $c6 \longrightarrow c7 \longrightarrow c8 \longrightarrow c6$ and $c3 \longrightarrow c4 \longrightarrow c3$ are referred to as subtours.
- Subtours are part of the vehicles' tours that are disconnected from the depot.
- Apparently a solution like this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

Let's stop and think



- 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 a feasible solution for the model defined above?
- Let's check constraint by constraint.

Let's stop and think



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

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

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

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

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

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

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

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

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.

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.
- In the labs, we will focus on the so-called MTZ-formulation introduced by Miller, Tucker and Zemlin (1960) for the TSP.
- The first set of constraints links the node demand q_i with the cumulated demand u_{ik} in a big-M fashion:
  s.t. $u_{ik} + q_j \leq u_{jk} + Q(1 - x_{ijk}), \quad \forall i, j \in N, k \in K$
- The second set of constraints enforces the vehicle capacity and provides a lower bound for u_{ik}:
  s.t. $q_i \leq u_{ik} \leq Q, \quad \forall i \in N, k \in K$

Complete model

$$
\begin{aligned}
\min \sum_{k\in K}\sum_{i\in N}\sum_{j\in N} c_{ij}x_{ijk} & & \\
\text{s.t.} \sum_{k\in K} y_{ik} &= 1, & \forall i \in N \setminus \{o,d\}\\
\sum_{j\in N\setminus\{i\}} x_{ijk} - \sum_{j\in N\setminus\{i\}} x_{jik} &= 0, & \forall i \in N \setminus \{o,d\}, k \in K\\
\sum_{j\in N\setminus\{o\}} x_{ojk} - \sum_{j\in N\setminus\{o\}} x_{jok} &= 1, & \forall k \in K\\
y_{ik} &= \sum_{j\in N\setminus\{i\}} x_{ijk}, & \forall i \in N \setminus \{d\}, k \in K\\
y_{dk} &= \sum_{i\in N\setminus\{d\}} x_{idk}, & \forall k \in K\\
u_{ik} + q_j &\le u_{jk} + Q(1 - x_{ijk}), & \forall i,j \in N, k \in K\\
q_i \le u_{ik} &\le Q, & \forall i \in N, k \in K\\
x_{ijk} &\in \{0,1\}, & \forall i,j \in N, k \in K\\
y_{ik} &\in \{0,1\}, & \forall i \in N, k \in K
\end{aligned}
$$

Overview

Versions of the VRP

- Many versions of the VRP have been considered in the literature
  - Capacitated VRP
  - VRP with time windows
  - Pickup and delivery VRP
  - VRP with backhauls
  - VRP with split deliveries
  - Periodic VRP
  - Heterogeneous fleet VRP
  - Dial-a-ride problem (DARP)
  - Stochastic VRP
  - Dynamic VRP
  - Inventory routing problem (IRP)
  - etc...
- The interested student is referred to Toth and Vigo (2002) or Toth and Vigo (2014). The full text of the former can be accessed online from the EPFL library website if you are on campus or connected through VPN.

Overview

Solution approaches

- The VRP, and by generalization, all of its extensions are NP-hard.
- Solution approaches can broadly be classified into three categories:
- Exact approaches:
  - Branch-and-bound
  - Branch-and-cut
  - Branch-and-price-and-cut
- Heuristic approaches:
  - Construction heuristics
  - Improvement heuristics
  - Metaheuristics–neighborhood based, population based
- Hybridizations

Overview

Exercise

- You are asked to develop a CPLEX OPL model for solving a VRP problem.
- The problem is the same as what you have seen here, but it has more features and more constraints, which you will have to develop.
- The problem is described in `exercise-session10.pdf`
- You are provided with the model and data file to start from.
- If your model is taking too long to optimize, you can limit the computation time by specifying:

  ```
  execute
  {
      cplex.tilim = 600;
  }
  ```

  in the beginning of your model file.
- This will force CPLEX to stop after 10 minutes and report the best feasible solution found so far.

Overview

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):133–148.

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

Toth, P and Vigo, D. (2002). *The Vehicle Routing Problem.* SIAM.

Toth, P and Vigo, D. (2014). *Vehicle Routing: Problems, Methods, and Applications, Second Edition.* SIAM.