Lecture 18
Vehicle Routing

Marco Chiarandini

Outline
1. Vehicle Routing
2. Integer Programming
3. Construction Heuristics
   Construction Heuristics for CVRP

Problem Definition
Vehicle Routing: distribution of goods between depots and customers.
Delivery, collection, transportation.
Examples: solid waste collection, street cleaning, school bus routing, dial-a-ride systems, transportation of handicapped persons, routing of salespeople and maintenance unit.

Vehicle Routing Problems
Input: Vehicles, depots, road network, costs and customers requirements.
Output: Set of routes such that:
 ▶ requirement of customers are fulfilled,
 ▶ operational constraints are satisfied and
 ▶ a global transportation cost is minimized.
Refinement

Road Network
- represented by a (directed or undirected) complete graph
- travel costs and travel times on the arcs obtained by shortest paths

Customers
- vertices of the graph
- collection or delivery demands
- time windows for service
- service time
- subset of vehicles that can serve them
- priority (if not obligatory visit)

Vehicles
- capacity
- types of goods
- subsets of arcs traversable
- fix costs associated to the use of a vehicle
- distance dependent costs
- a-priori partition of customers
- home depot in multi-depot systems
- drivers with union contracts

Operational Constraints
- vehicle capacity
- delivery or collection
- time windows
- working periods of the vehicle drivers
- precedence constraints on the customers

Objectives
- minimization of global transportation cost (variable + fixed costs)
- minimization of the number of vehicles
- balancing of the routes
- minimization of penalties for un-served customers

History:
Dantzig, Ramser “The truck dispatching problem”, Management Science, 1959
Clark, Wright, “Scheduling of vehicles from a central depot to a number of delivery points”. Operation Research. 1964
Vehicle Routing Problems

- Capacitated (and Distance Constrained) VRP (CVRP and DCVRP)
- VRP with Time Windows (VRPTW)
- VRP with Backhauls (VRPB)
- VRP with Pickup and Delivery (VRPPD)
- Periodic VRP (PVRP)
- Multiple Depot VRP (MDVRP)
- Split Delivery VRP (SDVRP)
- VRP with Satellite Facilities (VRPSF)
- Site Dependent VRP
- Open VRP
- Stochastic VRP (SVRP)
- ...

Capacitated Vehicle Routing (CVRP)

**Input:** (common to all VRPs)
- (d)igraph (strongly connected, typically complete) $G(V,A)$, where $V = \{0, \ldots, n\}$ is a vertex set:
  - $0$ is the depot.
  - $V' = V \setminus \{0\}$ is the set of $n$ customers
- $A = \{ (i,j) : i,j \in V \}$ is a set of arcs
- $C$ a matrix of non-negative costs or distances $c_{ij}$ between customers $i$ and $j$ (shortest path or Euclidean distance)
- $d_i$ a non-negative vector of customer demands
- a set of $K$ (identical!) vehicles with capacity $Q$, $d_i \leq Q$

**Task:** Find collection of $K$ circuits with minimum cost, defined as the sum of the costs of the arcs of the circuits and such that:
- each circuit visits the depot vertex
- each customer vertex is visited by exactly one circuit; and
- the sum of the demands of the vertices visited by a circuit does not exceed the vehicle capacity $Q$.

**Note:** lower bound on $K$
- $\lceil d(V')/Q \rceil$
- number of bins in the associated Bin Packing Problem

A feasible solution is composed of:
- a partition $R_1, \ldots, R_m$ of $V$;
- a permutation $\pi_i$ of $R_i \cup \{0\}$ specifying the order of the customers on route $i$.

A route $R_i$ is feasible if $\sum_{l=\pi_i^1}^{\pi_i^m} d_l \leq Q$.

The cost of a given route ($R_i$) is given by: $F(R_i) = \sum_{l=\pi_i^1}^{\pi_i^m} c_{l,l+1}$

The cost of the problem solution is: $F_{VRP} = \sum_{i=1}^{m} F(R_i)$. 
Relation with TSP

- VRP with $K = 1$, no limits, no (any) depot, customers with no demand ⇒ TSP
- VRP is a generalization of the Traveling Salesman Problem (TSP) ⇒ is NP-Hard.
- VRP with a depot, $K$ vehicles with no limits, customers with no demand ⇒ Multiple TSP = one origin and $K$ salesman
- Multiple TSP is transformable in a TSP by adding $K$ identical copies of the origin and making costs between copies infinite.

Variants of CVRP:

- minimize number of vehicles
- different vehicles $Q_k$, $k = 1, \ldots, K$
- Distance-Constrained VRP: length $t_{ij}$ on arcs and total duration of a route cannot exceed $T$ associated with each vehicle
  - Generally $c_{ij} = t_{ij}$
  - (Service times $s_i$ can be added to the travel times of the arcs: $t'_{ij} = t_{ij} + s_i/2 + s_j/2$)
- Distance constrained CVRP

Vehicle Routing with Time Windows (VRPTW)

Further Input:

- each vertex is also associated with a time interval $[a_i, b_i]$.
- each arc is associated with a travel time $t_{ij}$
- each vertex is associated with a service time $s_i$

Task:
Find a collection of $K$ simple circuits with minimum cost, such that:

- each circuit visit the depot vertex
- each customer vertex is visited by exactly one circuit; and
- the sum of the demands of the vertices visited by a circuit does not exceed the vehicle capacity $Q$.
- for each customer $i$, the service starts within the time windows $[a_i, b_i]$ (it is allowed to wait until $a_i$ if early arrive)

Time windows induce an orientation of the routes.
Variants

- Minimize number of routes
- Minimize hierarchical objective function
- Makespan VRP with Time Windows (MPTW) minimizing the completion time
- Delivery Man Problem with Time Windows (DMPTW) minimizing the sum of customers waiting times

Outline

1. Vehicle Routing
2. Integer Programming
3. Construction Heuristics
   Construction Heuristics for CVRP

Solution Techniques for CVRP

- Integer Programming (only formulations)
- Construction Heuristics
- Local Search
- Metaheuristics
- Hybridization with Constraint Programming

Basic Models

- vehicle flow formulation
  integer variables on the edges counting the number of time it is traversed
two or three index variables
- commodity flow formulation
  additional integer variables representing the flow of commodities along the paths traveled by the vehicles
- set partitioning formulation
VRPTW

Pre-processing

▶ Time windows reduction
  ▶ Increase earliest allowed departure time, $a_k$
  ▶ Decrease latest allowed arrival time $b_k$

▶ Arc elimination
  ▶ $a_i + t_{ij} > b_j \Rightarrow$ arc $(i, j)$ cannot exist
  ▶ $d_i + d_j > C \Rightarrow$ arcs $(i, j)$ and $(j, i)$ cannot exist

Construction Heuristics for CVRP

▶ TSP based heuristics

▶ Savings heuristics (Clarke and Wright)

▶ Insertion heuristics

▶ Cluster-first route-second
  ▶ Sweep algorithm
  ▶ Generalized assignment
  ▶ Location based heuristic
  ▶ Petal algorithm

▶ Route-first cluster-second

Cluster-first route-second seems to perform better
(Note: Distinction Construction Heuristic / Iterative Improvement is often blurred)

Construction heuristics for TSP

They can be used for route-first cluster-second or for growing multiple tours subject to capacity constraint.

▶ Heuristics that Grow Fragments
  ▶ Nearest neighborhood heuristics
  ▶ Double-Ended Nearest Neighbor heuristic
  ▶ Multiple Fragment heuristic (aka, greedy heuristic)

▶ Heuristics that Grow Tours
  ▶ Nearest Addition
  ▶ Farthest Addition
  ▶ Random Addition
  ▶ Clarke-Wright savings heuristic

▶ Heuristics based on Trees
  ▶ Minimum span tree heuristic
  ▶ Christofides’ heuristics

(But recall! Concorde: http://www.tsp.gatech.edu/)
NN (Flood, 1956)

1. Randomly select a starting node
2. Add to the last node the closest node until no more node is available
3. Connect the last node with the first node

Running time $O(N^2)$

Add the cheapest edge provided it does not create a cycle.

FA

1. Select a node and its farthest and build a tour of two nodes
2. Insert in the tour the farthest node $v$ until no more node is available

FA is more efficient than NA because the first few farthest points sketch a broad outline of the tour that is refined after.

Running time $O(N^3)$
1. Find a minimum spanning tree \( O(N^2) \)
2. Append the nodes in the tour in a depth-first, inorder traversal

Running time \( O(N^2) \)

\[ A = \frac{\text{MST}(I)}{\text{OPT}(I)} \leq 2 \]

**Construction Heuristics Specific for VRP**

Clarke-Wright Saving Heuristic (1964)

1. Start with an initial allocation of one vehicle to each customer (0 is the depot for VRP or any chosen city for TSP)

Sequential:
2. Consider in turn route \((0, i, \ldots, j, 0)\) determine \(s_{ki}\) and \(s_{jl}\)
3. Merge with \((k, 0)\) or \((0, l)\)
Construction Heuristics Specific for VRP

Clarke-Wright Saving Heuristic (1964)

1. Start with an initial allocation of one vehicle to each customer (0 is the depot for VRP or any chosen city for TSP)

Parallel:
2. Calculate saving \( s_{ij} = c_{0i} + c_{0j} - c_{ij} \) and order the saving in non-increasing order
3. scan \( s_{ij} \) merge routes if i) \( i \) and \( j \) are not in the same tour ii) neither \( i \) and \( j \) are interior to an existing route iii) vehicle and time capacity are not exceeded

Matching Based Saving Heuristic

1. Start with an initial allocation of one vehicle to each customer (0 is the depot for VRP or any chosen city for TSP)
2. Compute \( s_{pq} = t(S_p) + t(S_q) - t(S_p \cup S_q) \) where \( t(\cdot) \) is the TSP solution
3. solve a max weighted matching on the \( S_k \) with weights \( s_{pq} \) on edges. A connection between a route \( p \) and \( q \) exists only if the merging is feasible.

Insertion Heuristic

\[ \alpha(i, k, j) = c_{ik} + c_{kj} - \lambda c_{ij} \]
\[ \beta(i, k, j) = \mu c_{0k} - \alpha(i, k, j) \]

1. construct emerging route \((0, k, 0)\)
2. compute for all \( k \) unrouted the feasible insertion cost:
\[ \alpha^*(i_k, k, j_k) = \min \{ \alpha(i, k, j) \} \]
if no feasible insertion go to 1 otherwise choose \( k^* \) such that
\[ \beta^*(i^*_k, k^*, j^*_k) = \max \{ \beta(i_k, k, j_k) \} \]
Cluster-first route-second: Sweep algorithm [Wren & Holliday (1971)]

1. Choose $i^*$ and set $\theta(i^*) = 0$ for the rotating ray
2. Compute and rank the polar coordinates $(\theta, \rho)$ of each point
3. Assign customers to vehicles until capacity not exceeded. If needed start a new route. Repeat until all customers scheduled.


1. Choose $j_k$ at random for each route $k$
2. For each point compute
   \[
   d_{ik} = \min \{ c_{0,i} + c_{i,j_k} + c_{j_k,0}, c_{0,j_k} + c_{j_k,i} + c_{i,0} \} - (c_{0,j_k} + c_{j_k,0})
   \]
3. Solve GAP with $d_{ik}$, $Q$ and $q_i$


1. Determine seeds by solving a capacitated location problem (k-median)
2. Assign customers to closest seed

(better performance than insertion and saving heuristics)
Cluster-first route-second: Petal Algorithm

1. Construct a subset of feasible routes
2. Solve a set partitioning problem

Route-first cluster-second [Beasley]

1. Construct a TSP tour over all customers
2. Choose an arbitrary orientation of the TSP; partition the tour according to capacity constraint; repeat for several orientations and select the best
Alternatively, solve a shortest path in an acyclic digraph with costs on arcs: $d_{ij} = c_{0i} + c_{0j} + l_{ij}$ where $l_{ij}$ is the cost of traveling from $i$ to $j$ in the TSP tour.
(not very competitive)

Exercise

Which heuristics can be used to minimize $K$ and which ones need to have $K$ fixed a priori?