DM204, 2010
SCHEDULING, TIMETABLING AND ROUTING

Lecture 29
Vehicle Routing

Marco Chiarandini

Department of Mathematics & Computer Science
University of Southern Denmark

Course Overview

- ✔ Problem Introduction
  - ✔ Scheduling classification
  - ✔ Scheduling complexity
  - ✔ RCPSP
- ✔ General Methods
  - ✔ Integer Programming
  - ✔ Constraint Programming
  - ✔ Heuristics
  - ✔ Dynamic Programming
  - ✔ Branch and Bound
- ✔ Scheduling Models
  - ✔ Single Machine
  - ✔ Parallel Machine and Flow Shop
  - ✔ Job Shop
  - ✔ Resource-Constrained Project Scheduling
- Timetabling
  - ✔ Reservations and Education
  - ✔ Course Timetabling
  - ✔ Workforce Timetabling
  - ✔ Crew Scheduling
- Vehicle Routing
  - Capacited Models
  - Time Windows models
  - Rich Models

Outline

1. Vehicle Routing
2. Integer Programming

Outline

1. Vehicle Routing
2. Integer Programming

Problem Definition

Vehicle Routing: distribution of goods between depots and customers.

Delivery, collection, transportation.

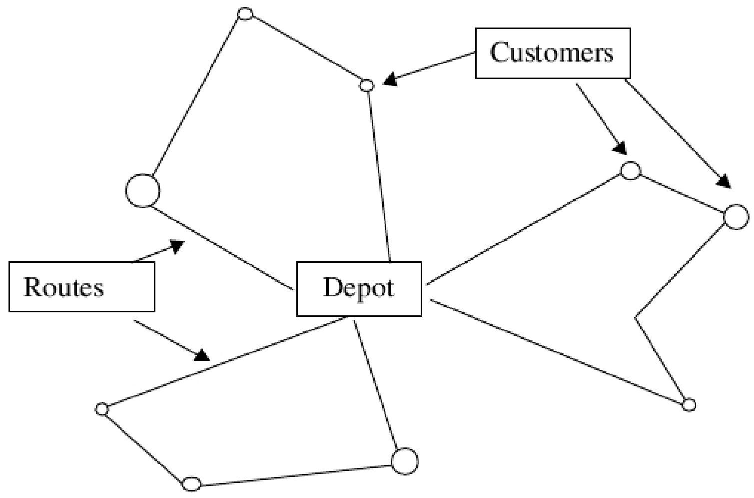
Examples: solid waste collection, street cleaning, school bus routing, dial-a-ride systems, transportation of handicapped persons, routing of salespeople and maintenance unit.

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

- Capacited (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)
- ...

Capacited Vehicle Routing (CVRP)

Input: (common to all VRPs)

- (di)graph (strongly connected, typically complete) $G(V, A)$, where $V = \{0, \ldots, n\}$ is a vertex set:
  - 0 is the depot.
  - $V' = V \backslash \{0\}$ 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) $(c_{ik} + c_{kj} \geq c_{ij} \quad \forall\, i, j \in V)$
- a non-negative vector of costumer demands d_i
- 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 π^i of $R_i \bigcup 0$ specifying the order of the customers on route i.

A route R_i is feasible if $\sum_{i=\pi_1}^{\pi_m} d_i \leq Q$.

The cost of a given route (R_i) is given by: $F(R_i) = \sum_{i=\pi_0^i}^{\pi_m^i} c_{i,i+1}$

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

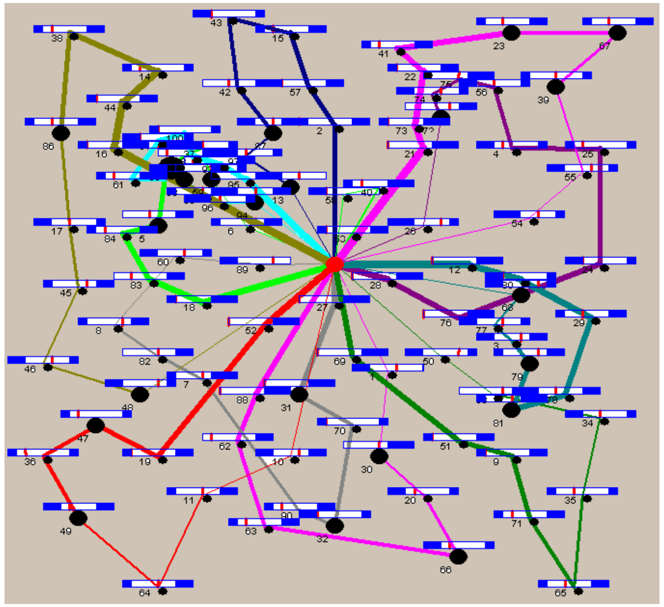
Vehicle Routing with Time Windows (VRPTW)

Further Input:

- each vertex is also associated with a time interval $[a_i, b_j]$.
- 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

Solution Techniques for CVRP

- Integer Programming
- Construction Heuristics
- Local Search
- Metaheuristics
- Hybridization with Constraint Programming

Outline

Basic Models

- arc flow formulation

  integer variables on the edges counting the number of time it is traversed
  one, two or three index variables

- set partitioning formulation

- multi-commodity network flow formulation for VRPTW

  integer variables representing the flow of commodities along the paths traveled by the vehicles and
  integer variables representing times

Two index arc flow formulation

$$
\begin{aligned}
\min \quad & \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij} && && (1)\\
\text{s.t.} \quad & \sum_{i \in V} x_{ij} = 1 && \forall j \in V \setminus \{0\} && (2)\\
& \sum_{j \in V} x_{ij} = 1 && \forall i \in V \setminus \{0\} && (3)\\
& \sum_{i \in V} x_{i0} = K && && (4)\\
& \sum_{j \in V} x_{0j} = K && && (5)\\
& \sum_{i \in S} \sum_{j \notin S} x_{ij} \geq r(S) && \forall S \subseteq V \setminus \{0\}, S \neq \emptyset && (6)\\
& x_{ij} \in \{0,1\} && \forall i,j \in V && (7)
\end{aligned}
$$

One index arc flow formulation

$$
\begin{aligned}
\min \quad & \sum_{e \in E} c_e x_e && && (8)\\
\text{s.t.} \quad & \sum_{e \in \delta(i)} x_e = 2 && \forall i \in V \setminus \{0\} && (9)\\
& \sum_{e \in \delta(0)} x_e = 2K && && (10)\\
& \sum_{e \in \delta S} x_e \geq 2r(S) && \forall S \subseteq V \setminus \{0\}, S \neq \emptyset && (11)\\
& x_e \in \{0,1\} && \forall e \notin \delta(0) && (12)\\
& x_e \in \{0,1,2\} && \forall e \in \delta(0) && (13)
\end{aligned}
$$

Three index arc flow formulation

$$
\begin{aligned}
\min \quad & \sum_{i \in V} \sum_{j \in V} c_{ij} \sum_{k=1}^{K} x_{ijk} && && (14)\\
\text{s.t.} \quad & \sum_{k=1}^{K} y_{ik} = 1 && \forall i \in V \setminus \{0\} && (15)\\
& \sum_{k=1}^{K} y_{0k} = K && && (16)\\
& \sum_{j \in V} x_{ijk} = \sum_{j \in V} x_{jik} = y_{ik} && \forall i \in V, k = 1, \ldots, K && (17)\\
& \sum_{i \in V} d_i y_{ik} \leq C && \forall k = 1, \ldots, K && (18)\\
& \sum_{i \in S} \sum_{j \notin S} x_{ijk} \geq y_{hk} && \forall S \subseteq V \setminus \{0\}, h \in S, k = 1, 1, \ldots, K && (19)\\
& y_{ik} \in \{0,1\} && \forall i \in V, k = 1, \ldots, K && (20)\\
& x_{ijk} \in \{0,1\} && \forall i,j \in V, k = 1, \ldots, K && (21)
\end{aligned}
$$

Set Covering Formulation

$\mathcal{R} = \{1, 2, \dots, R\}$ index set of routes

$$a_{ir} = \begin{cases} 1 & \text{if costumer } i \text{ is served by } r \\ 0 & \text{otherwise} \end{cases}$$

$$x_r = \begin{cases} 1 & \text{if route } r \text{ is selected} \\ 0 & \text{otherwise} \end{cases}$$

$$\min \sum_{r \in \mathcal{R}} c_r x_r \quad (22)$$

$$\text{s.t.} \sum_{r \in \mathcal{R}} a_{ir} x_r \geq 1 \qquad \forall i \in V \quad (23)$$

$$\sum_{r \in \mathcal{R}} x_r \leq K \quad (24)$$

$$x_r \in \{0, 1\} \qquad \forall r \in \mathcal{R} \quad (25)$$

$$(26)$$

What can we do with these integer programs?

- plug them into a commercial solver and try to solve them
- preprocess them
- determine lower bounds
  - solve the linear relaxation
  - combinatorial relaxations
    relax some constraints and get an easy solvable problem
  - Lagrangian relaxation
  - polyhedral study to tighten the formulations
- upper bounds via heuristics
- branch and bound
- cutting plane
- branch and cut
- Dantzig Wolfe decomposition
- column generation (via reformulation)
- branch and price

Combinatorial Relaxations

Lower bounding via combinatorial relaxations

Relax: capacity cut constraints (CCC)
or generalized subtour elimination constraints (GSEC) Consider both ACVRP and SCVRP

- Relax CCC in 2-index formulation
  obtain a transportation problem
  Solution may contain isolated circuits and exceed vertex capacity
- Relax CCC in 1-index formulation
  obtain a b-matching problem

$$\min \sum_{e \in E} c_e x_e$$

$$\text{s.t.} \sum_{e \in \delta(i)} x_e = b_i \qquad \forall i \in V\{0\}$$

$$x_e \in \{0, 1\} \qquad \forall e \notin \delta(0)$$

$$x_e \in \{0, 1, 2\} \qquad \forall e \in \delta(0)$$

Solution has same problems as above

- relax in two index flow formulation:

$$\min \sum_{i \in V} \sum_{j \in V} c_{ij} x_{ij}$$

$$\text{s.t.} \sum_{i \in V} x_{ij} = 1 \qquad \forall j \in V \setminus \{0\}$$

$$\sum_{j \in V} x_{ij} = 1 \qquad \forall i \in V \setminus \{0\}$$

$$\sum_{i \in V} x_{i0} = K$$

$$\sum_{j \in V} x_{0j} = K$$

$$\sum_{i \in S} \sum_{i \notin S} x_{ij} \geq r(S) 1 \qquad \forall S \subseteq V \setminus \{0\}, S \neq \emptyset$$

$$x_{ij} \in \{0, 1\} \qquad \forall i, j \in V$$

K-shortest spanning arborescence problem

- relax in two index formulation

$$\begin{aligned}
\min \quad & \sum_{e \in E} c_e x_e \\
\text{s.t.} \quad & \sum_{e \in \delta(i)} x_e = 2 && \forall i \in V \setminus \{0\} \\
& \sum_{e \in \delta(0)} x_e = 2K \\
& \sum_{e \in \delta S} x_e \geq 2r(S) && \forall S \subseteq V \setminus \{0\}, S \neq \emptyset \\
& x_e \in \{0,1\} && \forall e \notin \delta(0)
\end{aligned}$$

  K-tree: min cost set of $n+K$ edges spanning the graph with degree $2K$ at the depot.

- Lagrangian relaxation of the sub tour constraints iteratively added after violation recognized by separation procedure.
  Subgradient optimization for the multipliers

Branch and Cut

$$\begin{aligned}
\min \quad & \sum_{e \in E} c_e x_e && (27) \\
\text{s.t.} \quad & \sum_{e \in \delta(i)} x_e = 2 && \forall i \in V \setminus \{0\} \quad (28) \\
& \sum_{e \in \delta(0)} x_e = 2K && (29) \\
& \sum_{e \in \delta S} x_e \geq 2\left\lceil \frac{d(S)}{C} \right\rceil && \forall S \subseteq V \setminus \{0\}, S \neq \emptyset \quad (30) \\
& x_e \in \{0,1\} && \forall e \notin \delta(0) \quad (31) \\
& x_e \in \{0,1,2\} && \forall e \in \delta(0) \quad (32)
\end{aligned}$$

Branch and Cut

- Let $LP(\infty)$ be linear relaxation of IP
- $z_{LP(\infty)} \leq z_{IP}$
- Problems if many constraints
- Solve $LP(h)$ instead and add constraints later
- If $LP(h)$ has integer solution then we are done, that is optimal
  If not, then $z_{LP(h)} \leq z_{LP(h+1)} \leq z_{LP(\infty)} \leq z_{IP}$
- Crucial step: separation algorithm given a solution to $LP(h)$ it tell us if some constraints are violated.

If the procedure does not return an integer solution we proceed by branch and bound

Problems with B&C:

i) no good algorithm for the separation phase
it may be not exact in which case bounds relations still hold and we can go on with branching

ii) number of iterations for cutting phase is too high

iii) program unsolvable because of size

iv) the tree explodes

The main problem is (iv).
Worth trying to strengthen the linear relaxation by inequalities that although unnecessary in the integer formulation force the optimal solution of LP and IP to get closer. ➨ Polyhedral studies

On the Set Covering Formulation

Solving the SCP integer program

Branch and bound

- Generate routes such that:
  - they are good in terms of cost
  - they reduce the potential for fractional solutions
- constraint branching [Ryan, Foster, 1981]

$$\exists \text{ constraints } r_1, r_2 : 0 < \sum_{j \in J(r_1, r_2)} x_j < 1$$

$J(r_1, r_2)$ all columns covering r_1, r_2 simultaneously. Branch on:

$$\sum_{j \in J(r_1, r_2)} x_j \leq 0 \qquad\qquad \sum_{j \in J(r_1, r_2)} x_j \geq 1$$

Solving the SCP linear relaxation

Column Generation Algorithm

Step 1 Generate an initial set of columns $\mathcal{R}'$

Step 2 Solve problem P' and get optimal primal variables, $\bar{x}$, and optimal dual variables, $(\bar{\pi}, \bar{\theta})$

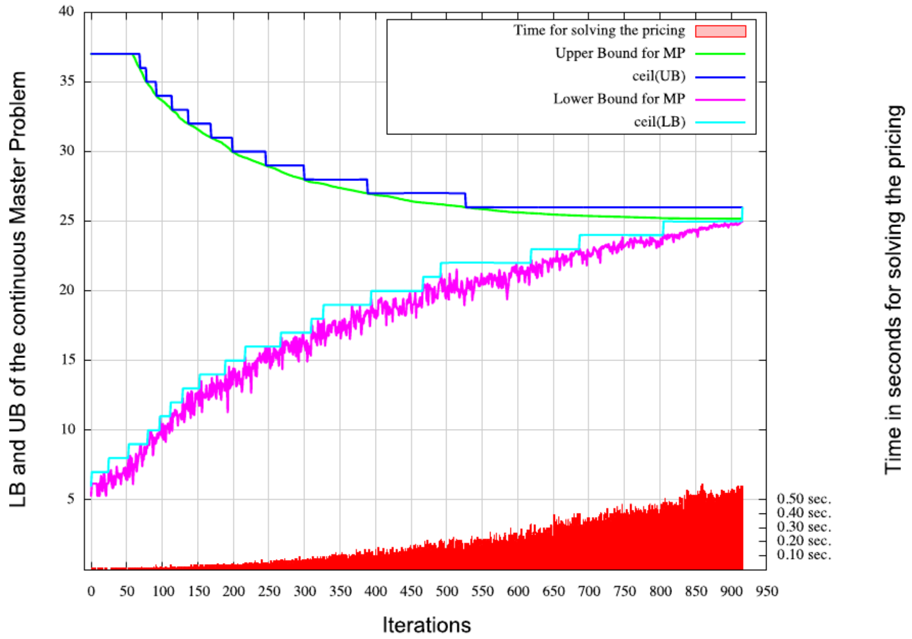
Step 3 Solve problem CG, or identify routes $r \in \mathcal{R}$ satisfying $\bar{c}_r < 0$

Step 4 For every $r \in \mathcal{R}$ with $\bar{c}_r < 0$ add the column r to $\mathcal{R}'$ and go to Step 2

Step 5 If no routes r have $\bar{c}_r < 0$, i.e., $\bar{c}_{min} \geq 0$ then stop.

In most of the cases we are left with a fractional solution

Convergence in CG



[plot by Stefano Gualandi, Milan University]

Solving the SCP integer program:

- cutting plane
- branch and price

Cutting Plane Algorithm

Step 1 Generate an initial set $\mathcal{R}'$ of columns

Step 2 Solve, using column generation, the problem P' (linear programming relaxation of P)

Step 3 If the optimal solution to P' is integer stop.
Else, generate cutting plane separating this fractional solution.
Add these cutting planes to the linear program P'

Step 4 Solve the linear program P'. Goto Step 3.

Is the solution to this procedure overall optimal?

Cuts

Intersection graph $G = (V, E)$ where V are the routes and E is made by the links between routes that intercept
Independence set in G is a collection of routes where each customer is visited only once.

Clique constraints

$$\sum_{r \in K} \bar{x}_r \leq 1 \qquad \forall \text{ cliques } K \text{ of } G$$

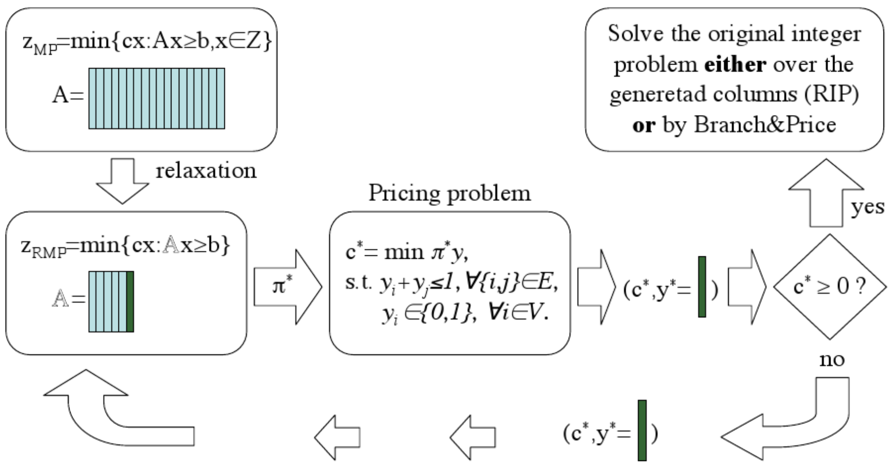
Cliques searched heuristically

Odd holes

Odd hole: odd cycle with no chord

$$\sum_{r \in H} \bar{x}_r \leq \frac{|H| - 1}{2} \qquad \forall \text{ odd holes } H$$

Generation via layered graphs



[illustration by Stefano Gualandi, Milan Un.]
(the pricing problem is for a GCP)

Branch and price

- it must be possible to incorporate information on the node in the column generation procedure
- easy to incorporate $x_r = 1$, just omit nodes in S_r from generation; but not clear how to impose $x_r = 0$.
- different branching: on the edges: $x_{ij} = 1$ then in column generation set $c_{ij} = -\infty$; if $x_{ij} = 0$ then $c_{ij} = \infty$