DM87
SCHEDULING,
TIMETABLING AND ROUTING

Lecture 19

Vehicle Routing
Heuristic Methods

Marco Chiarandini

Outline

Outline

Construction Heuristics for VRPTW

Extensions of those for CVRP [Solomon (1987)]

- Savings heuristics (Clarke and Wright)
- Time-oriented nearest neighbors
- Insertion heuristics
- Time-oriented sweep heuristic

Time-Oriented Nearest-Neighbor

- Add the unrouted node "closest" to the depot or the last node added without violating feasibility
- Metric for "closest":

$$c_{ij} = \delta_1 d_{ij} + \delta_2 T_{ij} + \delta_3 v_{ij}$$

d_{ij} geographical distance
T_{ij} time distance
v_{ij} urgency to serve j

Insertion Heuristics

Step 1: Compute for each unrouted costumer u the *best feasible position* in the route:

$$c_1(i(u), u, j(u)) = \min_{p=1,\dots,m} \{c_1(i_{p-1}, u, i_p)\}$$

(c_1 is a composition of increased time and increase route length due to the insertion of u)
(use push forward rule to check feasibility efficiently)

Step 2: Compute for each unrouted customer u which can be feasibly inserted:

$$c_2(i(u^*), u^*, j(u^*)) = \max_u \{\lambda d_{0u} - c_1(i(u), u, j(u))\}$$

(max the benefit of servicing a node on a partial route rather than on a direct route)

Step 3: Insert the customer u^* from Step 2

Outline

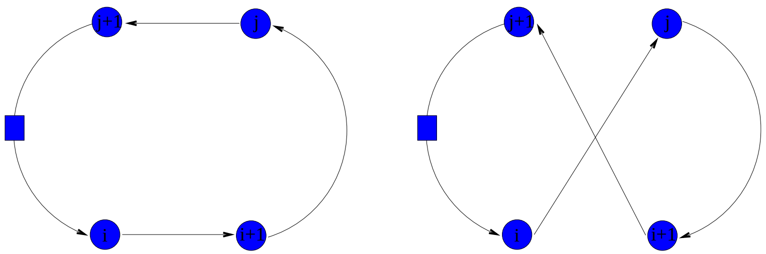
Local Search for CVRP and VRPTW

- Neighborhoods structures:
  - Intra-route: 2-opt, 3-opt, Lin-Kernighan (not very well suited) 2H-opt, Or-opt
  - Inter-routes: λ-interchange, relocate, exchange, cross, 2-opt*, ejection chains, GENI
- Solution representation and data structures
  - They depend on the neighborhood.
  - It can be advantageous to change them from one stage to another of the heuristic

Intra-route Neighborhoods

2-opt

$$\{i, i+1\}\{j, j+1\} \longrightarrow \{i, j\}\{i+1, j+1\}$$

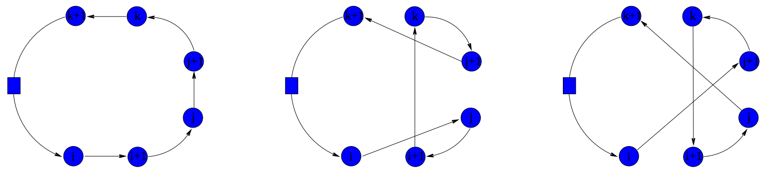


$O(n^2)$ possible exchanges
One path is reversed

Intra-route Neighborhoods

3-opt

$$\{i, i+1\}\{j, j+1\}\{k, k+1\} \longrightarrow \ldots$$

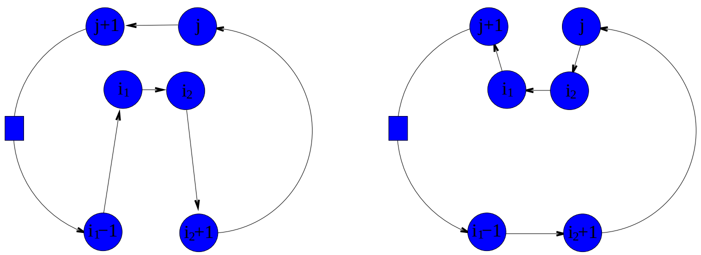


$O(n^3)$ possible exchanges
Paths can be reversed

Intra-route Neighborhoods

Or-opt [Or (1976)]

$$\{i_1-1, i_1\}\{i_2, i_2+1\}\{j, j+1\} \longrightarrow \{i_1-1, i_2+1\}\{j, i_1\}\{i_2, j+1\}$$



sequences of one, two, three consecutive vertices relocated
$O(n^2)$ possible exchanges — No paths reversed

Time windows: Feasibility check

In TSP verifying k-optimality requires $O(n^k)$ time
In TSPTW feasibility has to be tested then $O(n^{k+1})$ time

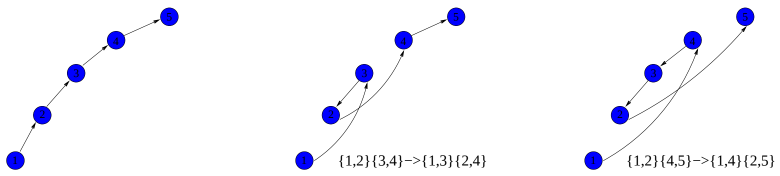
(Savelsbergh 1985) shows how to verify constraints in constant time
Search strategy + Global variables

$\Downarrow$

$O(n^k)$ for k-optimality in TSPTW

Search Strategy

- Lexicographic search, for 2-exchange:
  - $i = 1, 2, \ldots, n-2$ (outer loop)
  - $j = i+2, i+3, \ldots, n$ (inner loop)



Previous path is expanded by the edge $\{j-1, j\}$

Global variables (auxiliary data structure)

- Maintain auxiliary data such that it is possible to:
  - handle single move in constant time
  - update their values in constant time

Ex.: in case of time windows:

- total travel time of a path
- earliest departure time of a path
- latest arrival time of a path

Inter-route Neighborhoods

[Savelsbergh, ORSA (1992)]

Figure 6. The exchange neighborhood.

Inter-route Neighborhoods

[Savelsbergh, ORSA (1992)]

Figure 5. The relocate neighborhood.

Inter-route Neighborhoods

[Savelsbergh, ORSA (1992)]

Figure 7. The cross neighborhood.

Inter-route Neighborhoods

GENI: generalized insertion [Gendreau, Hertz, Laporte, Oper. Res. (1992)]

- select the insertion restricted to the neighborhood of the vertex to be added (not necessarily between consecutive vertices)
- perform the best 3- or 4-opt restricted to reconnecting arc links that are close to one another.

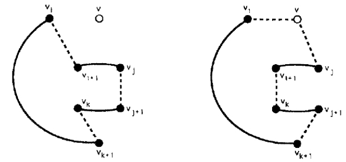


Figure 1. Type I insertion of vertex v between v_i and v_j.

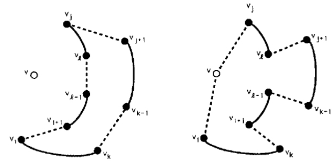


Figure 2. Type II insertion of vertex v between v_i and v_j.

General recommendation: use a combination of 2-opt* + or-opt [Potvin, Rousseau, (1995)]

However,

- Designing a local search algorithm is an engineering process in which learnings from other courses in CS might become important.
- It is important to make such algorithms as much efficient as possible.
- Many choices are to be taken (search strategy, order, auxiliary data structures, etc.) and they may interact with instance features. Often a trade-off between examination cost and solution quality must be decided.
- The assessment is conducted through:
  - analytical analysis (computational complexity)
  - experimental analysis

Table 5.6. *The effect of 3-opt on the Clarke and Wright algorithm.*

	Sequential				Parallel			
Problem	No 3-opt[1]	+ 3-opt FI[2]	+ 3-opt BI[3]	K[4]	No 3-opt[5]	+ 3-opt FI[6]	+ 3-opt BI[7]	K[8]
E051-05e	625.56	624.20	624.20	5	584.64	578.56	578.56	6
E076-10e	1005.25	991.94	991.94	10	900.26	888.04	888.04	10
E101-08e	982.48	980.93	980.93	8	886.83	878.70	878.70	8
E101-10c	939.99	930.78	928.64	10	833.51	824.42	824.42	10
E121-07c	1291.33	1232.90	1237.26	7	1071.07	1049.43	1048.53	7
E151-12c	1299.39	1270.34	1270.34	12	1133.43	1128.24	1128.24	12
E200-17c	1708.00	1667.65	1669.74	16	1395.74	1386.84	1386.84	17
D051-06c	670.01	663.59	663.59	6	618.40	616.66	616.66	6
D076-11c	989.42	988.74	988.74	12	975.46	974.79	974.79	12
D101-09c	1054.70	1046.69	1046.69	10	973.94	968.73	968.73	9
D101-11c	952.53	943.79	943.79	11	875.75	868.50	868.50	11
D121-11c	1646.60	1638.39	1637.07	11	1596.72	1587.93	1587.93	11
D151-14c	1383.87	1374.15	1374.15	15	1287.64	1284.63	1284.63	15
D200-18c	1671.29	1652.58	1652.58	20	1538.66	1523.24	1521.94	19

[1]Sequential savings.
[2]Sequential savings + 3-opt and first improvement.
[3]Sequential savings + 3-opt and best improvement.
[4]Sequential savings: number of vehicles in solution.
[5]Parallel savings.
[6]Parallel savings + 3-opt and first improvement.
[7]Parallel savings + 3-opt and best improvement.
[8]Parallel savings: number of vehicles in solution.

What is best?

Outline

1. Construction Heuristics for VRPTW

2. Local Search

3. Metaheuristics

4. Other Variants of VRP

Tabu Search for VRPTW [Potvin (1996)]

Initial solution: Solomon's insertion heuristic

Neighborhood: or-opt and 2-opt* (in VNS fashion or neighborhood union)
speed up in or-opt: i is moved between j and $j+q$ if i is one of the h nearest neighbors

Step : best improvement

Tabu criterion: forbidden to reinsert edges which were recently removed

Tabu length: fixed

Aspiration criterion: tabu move is overridden if an overall best is reached

End criterion: number of iterations without improvements

Taburoute

[Gendreau, Hertz, Laporte, 1994]

Neighborhood: remove one vertex from one route and insert with GENI in another that contains one of its p nearest neighbors
Re-optimization of routes at different stages

Tabu criterion: forbidden to reinsert vertex in route

Tabu length: random from $[5, 10]$

Evaluation function: possible to examine infeasible routes + diversification component:

- penalty term measuring overcapacity (every 10 iteration multiplied or divided by 2)
- penalty term measuring overduration
- frequency of movement of a vertex currently considered

Overall strategy: false restart (initially several solutions, limited search for each of them, selection of the best)

False restart:

Step 1: (Initialization) Generate $\lceil\sqrt{n}/2\rceil$ initial solutions and perform tabu search on $W' \subset W = V \setminus \{0\}$ ($|W'| \approx 0.9|W|$) up to 50 idle iterations.

Step 2: (Improvement) Starting with the best solution observed in Step 1 perform tabu search on $W' \subset W = V \setminus \{0\}$ ($|W'| \approx 0.9|W|$) up to $50n$ idle iterations.

Step 3: (Intensification) Starting with the best solution observed in Step 2, perform tabu search up to 50 idle iterations.
Here W' is the set of the $\lceil |V|/2 \rceil$ vertices that have been most often moved in Steps 1 and 2.

Adaptive Memory Procedure

[Rochart and Taillard, 1995]

1. Keep an adaptive memory as a pool of good solutions
2. Some element (single tour) of these solutions are combined together to form new solution (more weight is given to best solutions)
3. Partial solutions are completed by an insertion procedure.
4. Tabu search is applied at the tour level

Granular Tabu Search

[Toth and Vigo, 1995]

Long edges are unlikely to be in the optimal solution

⇓

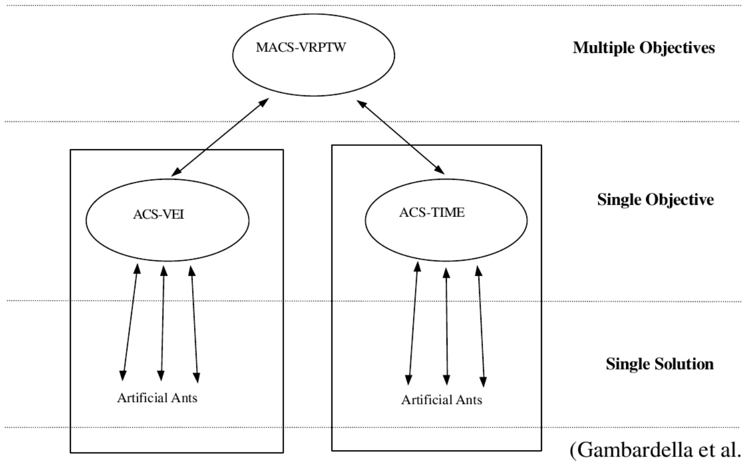
Remove those edges that exceed a granularity threshold ν

$$\nu = \beta \bar{c}$$

- β sparsification parameter
- $\bar{c}$ average length for a solution from a construction heuristic
- adjust β after a number of idle iterations

Ant Colony System [Gambardella et al. 1999]

VRP-TW: in case of vehicle and distance minimization two ant colonies are working in parallel on the two objective functions (colonies exchange pheromone information)



(Gambardella et al. 1999

Constraints: A constructed solution must satisfy i) each customer visited once ii) capacity not exceeded iii) Time windows not violated

Pheromone trails: associated with connections (desirability of order)

Heuristic information: savings + time considerations

Solution construction:

$$p_{ij}^{k} = \frac{\tau_{ij}^{\alpha}\eta_{ij}^{\beta}}{\sum_{l \in N_i^k} \tau_{il}^{\alpha}\eta_{il}^{\beta}} \qquad j \in N_i^k$$

if no feasible, open a new route
or decide routes to merge
if customers left out use an insertion procedure

Pheromone update:

Global $\quad \tau_{ij} \leftarrow \tau_{ij} + \rho \Delta \tau_{ij}^{bs} \qquad \forall (i,j) \in T^{bs}$

Local $\quad \tau_{ij} \leftarrow (1-\epsilon)\tau_{ij} + \epsilon \tau_{o}^{bs} \qquad \forall (i,j) \in T^{bs}$

Outline

Vehicle Routing with Backhauls (VRPB)

Further Input from CVRP:

- a partition of customers:
  $L = \{1, \ldots, n\}$ Lineahaul customers (deliveries)
  $B = \{n+1, \ldots, n+m\}$ Backhaul customers (collections)
- precedence constraint:
  in a route, customers from L must be served before customers from B

Task: Find a collection of K simple circuits with minimum costs, 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.
- in any circuit all the linehaul customers precede the backhaul customers, if any.

Vehicle Routing with Pickup and Delivery (VRPPD)

Further Input from CVRP:

- each customer i is associated with quantities d_i and p_i to be delivered and picked up, resp.
- for each customer i, O_i denotes the vertex that is the origin of the delivery demand and D_i denotes the vertex that is the destination of the pickup demand

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

- each circuit visit the depot vertex
- each customer vertex is visited by exactly one circuit; and
- the current load of the vehicle along the circuit must be non-negative and may never exceed Q
- for each customer i, the customer O_i when different from the depot, must be served in the same circuit and before customer i
- for each customer i, the customer D_i when different from the depot, must be served in the same circuit and after customer i

Multiple Depots VRP

Further Input from CVRP:

- multiple depots to which customers can be assigned
- a fleet of vehicles at each depot

Task:
Find a collection of K simple circuits for each depot with minimum costs, such that:

- each circuit visit the depot vertex
- each customer vertex is visited by exactly one circuit; and
- the current load of the vehicle along the circuit must be non-negative and may never exceed Q
- vehicles start and return to the depots they belong

Vertex set $V = \{1, 2, \ldots, n\}$ and $V_0 = \{n+1, \ldots, n+m\}$
Route i defined by $R_i = \{l, 1, \ldots, l\}$

Periodic VRP

Further Input from CVRP:

- planning period of M days

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

- each circuit visit the depot vertex
- each customer vertex is visited by exactly one circuit; and
- the current load of the vehicle along the circuit must be non-negative and may never exceed Q
- A vehicle may not return to the depot in the same day it departs.
- Over the M-day period, each customer must be visited l times, where $1 \leq l \leq M$.

Three phase approach:

1. Generate feasible alternatives for each customer.
   Example, $M = 3$ days $\{d1, d2, d3\}$ then the possible combinations are: $0 \rightarrow 000$; $1 \rightarrow 001$; $2 \rightarrow 010$; $3 \rightarrow 011$; $4 \rightarrow 100$; $5 \rightarrow 101$; $6 \rightarrow 110$; $7 \rightarrow 111$.

Customer	Diary Demand	Number of Visits	Number of Combinations	Possible Combinations
1	30	1	3	1,2,4
2	20	2	3	3,4,6
3	20	2	3	3,4,6
4	30	2	3	1,2,4
5	10	3	1	7

2. Select one of the alternatives for each customer, so that the daily constraints are satisfied. Thus, select the customers to be visited in each day.
3. Solve the vehicle routing problem for each day.

Split Delivery VRP

Constraint Relaxation: it is allowed to serve the same customer by different vehicles. (necessary if $d_i > Q$)

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

- each circuit visit the depot vertex
- the current load of the vehicle along the circuit must be non-negative and may never exceed Q

Note: a SDVRP can be transformed into a VRP by splitting each customer order into a number of smaller indivisible orders [Burrows 1988].

Inventory VRP

Input:

- a facility, a set of customers and a planning horizon T
- r_i product consumption rate of customer i (volume per day)
- C_i maximum local inventory of the product for customer i
- a fleet of M homogeneous vehicles with capacity Q

Task:
Find a collection of K daily circuits to run over the planing horizon with minimum costs and such that:

- each circuit visit the depot vertex
- no customer goes in stock-out during the planning horizon
- the current load of the vehicle along the circuit must be non-negative and may never exceed Q

Other VRPs

VRP with Satellite Facilities (VRPSF)

Possible use of satellite facilities to replenish vehicles during a route.

Open VRP (OVRP)

The vehicles do not need to return at the depot, hence routes are not circuits but paths

Dial-a-ride VRP (DARP)

- It generalizes the VRPTW and VRP with Pick-up and Delivery by incorporating time windows and maximum ride time constraints
- It has a human perspective
- Vehicle capacity is normally constraining in the DARP whereas it is often redundant in PDVRP applications (collection and delivery of letters and small parcels)