DM841

Discrete Optimization

Vehicle Routing
Local Search based Metaheuristics

Marco Chiarandini

Department of Mathematics & Computer Science
University of Southern Denmark

Outline

Outline

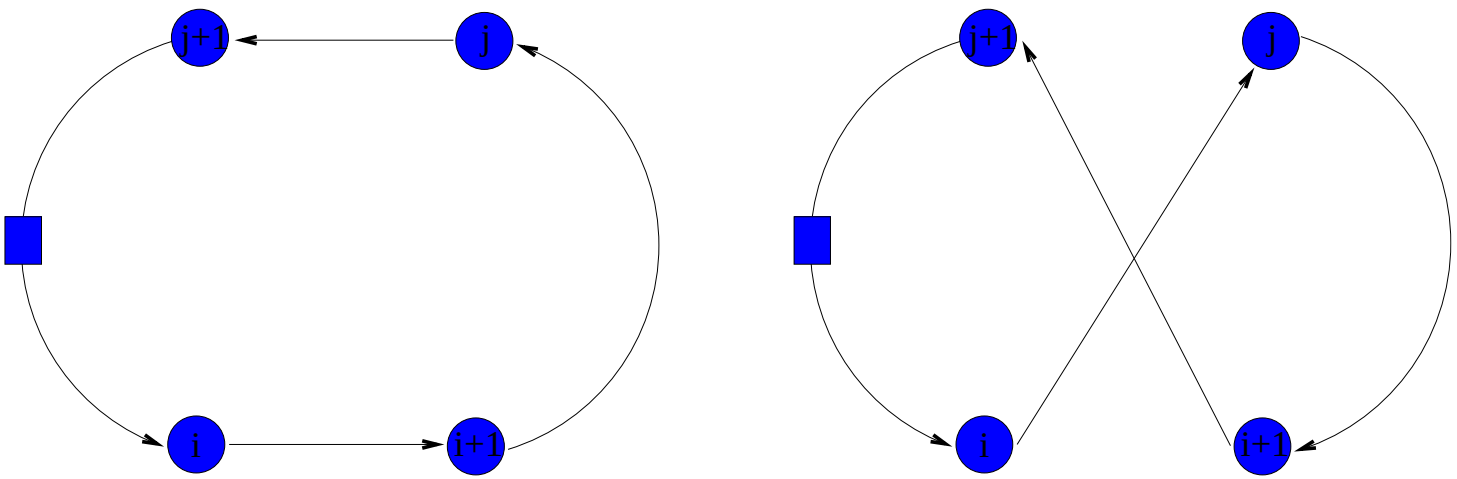
Local Search for CVRP and VRPTW

- Neighborhood structures:
  - Intra-route: 2-opt, 3-opt, Lin-Kernighan (not very well suited), Or-opt (2H-opt)
  - Inter-routes: λ-interchange, relocate, exchange, cross, 2-opt*, b-cyclic k-transfer (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

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

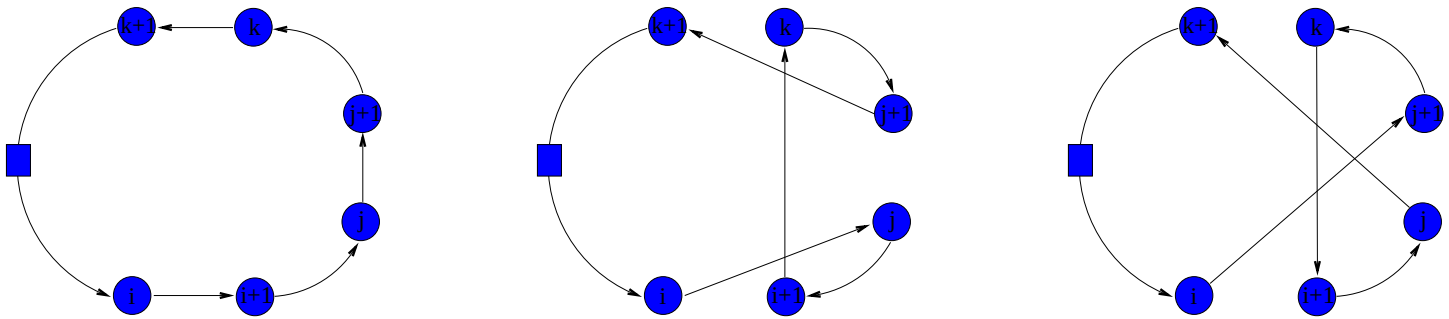


$O(n^2)$ possible exchanges
One path is reversed

Intra-route Neighborhoods

3-opt

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

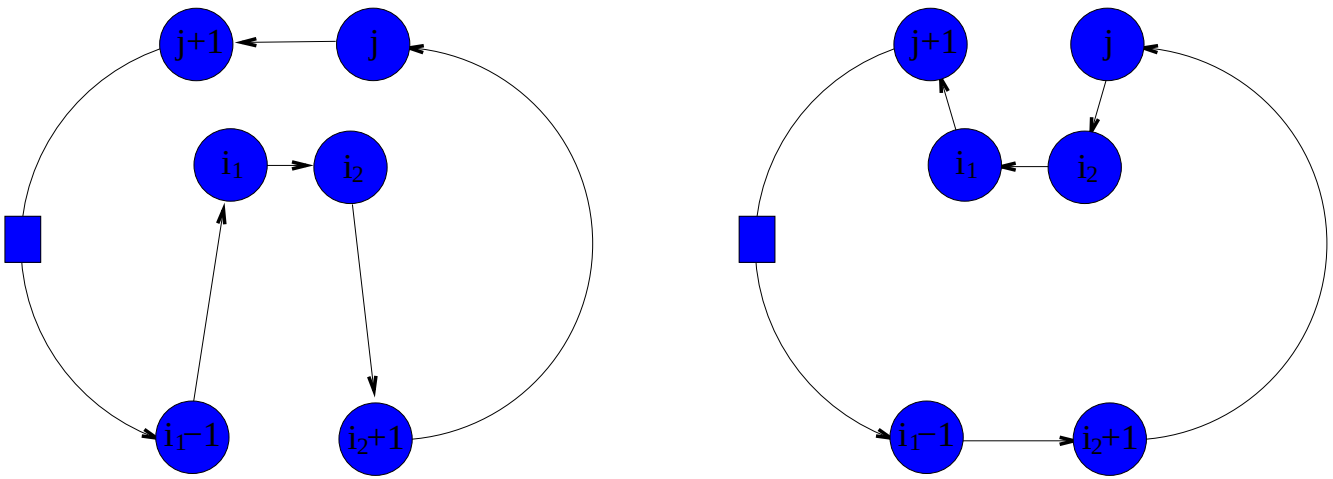


$O(n^3)$ possible exchanges
Paths can be reversed

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

Inter-route Neighborhoods

[Savelsbergh, ORSA (1992)]

Figure 6. The exchange neighborhood.

Inter-route Neighborhoods

[Savelsbergh, ORSA (1992)]

Figure 5. The relocate neighborhood.

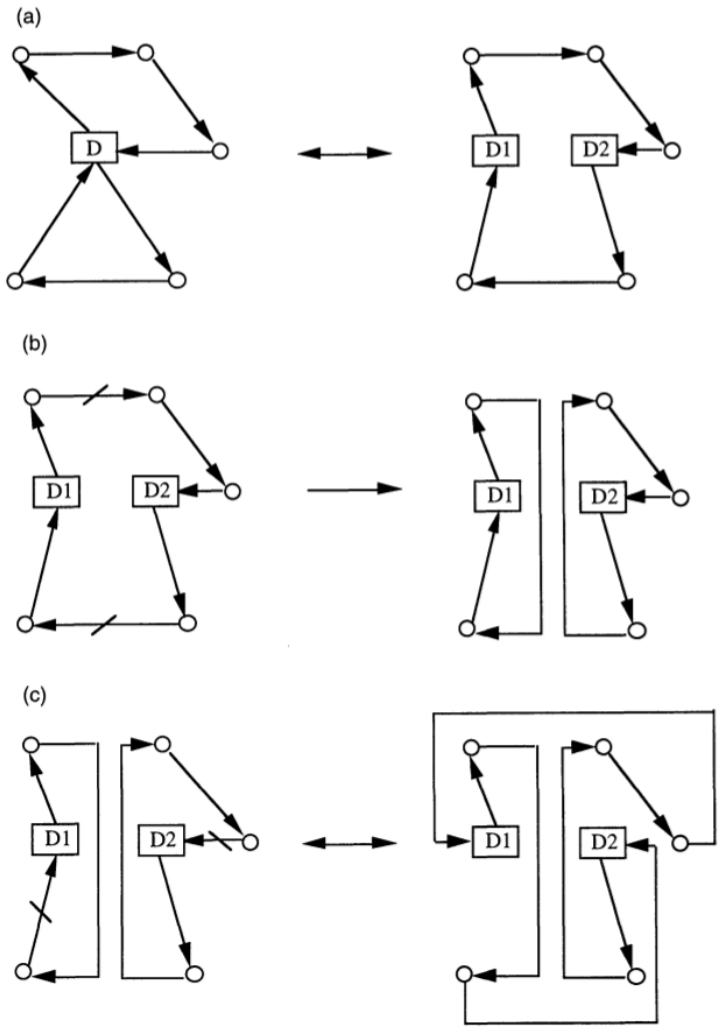
Inter-route Neighborhoods

[Savelsbergh, ORSA (1992)]

Figure 7. The cross neighborhood.

2-opt*

Exchanges 2 pairs of edges between routes.
First transform in TSP by creating M depots



$O(n^2)$ possible moves. (4 edges introduced but 2 are fixed given the other 2)
Preserve orientation and introduce last part of a route on another route.

[Potvin, J.-M. Rousseau, An exchange heuristic for routing problems with time windows Journal of the Operational Research Society, 46 (1995), pp. 1433-1446]

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.

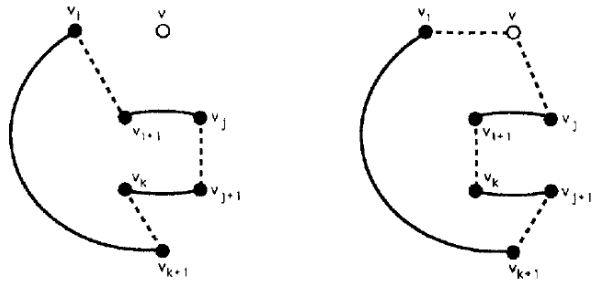


Figure 1. Type I insertion of vertex v between v_i and v_j.

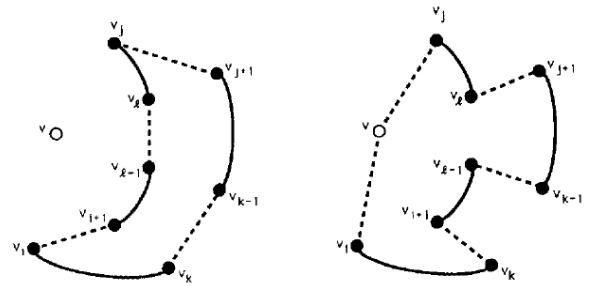


Figure 2. Type II insertion of vertex v between v_i and v_j.

A Local Search

Algorithm 1 Local Improvement(s_{CURR})

```
1: isEnd = false
2: for each route r ∈ s_CURR do updateData(r)
3: while not isEnd do
4:     isEnd = true
5:     for i = 1,...,n and j = 1,...,n do
6:         c_i ←shuffledNodeOrder(i) ; c_j ← shuffledNodeOrder(j) ;
7:         r_i ←getRoute(c_i) ; r_j ← getRoute(c_j) ;
8:         if r_i ≠ r_j and {isImprovingCROSS(c_i,c_j) or isImproving2opt*(c_i,c_j)} then
9:             s_CURR ← performMove(s_CURR;c_i,c_j) ; updateData(r_i,r_j); isEnd = false
10:        if r_i == r_j and {isImprovingOrOpt(c_i,c_j) or isImproving2Opt(c_i,c_j)} then
11:            s_CURR ← performMove(s_CURR;c_i) ; updateData(r_i); isEnd = false;
12: return s_CURR
```