utt
université de technologie Troyes
Institut Charles Delaunay

cnrs

Tour splitting algorithms for vehicle routing problems

Prof. Christian PRINS

christian.prins@utt.fr

Institute Charles Delaunay (ICD) – UTT
12 rue Marie Curie, CS 42060, 10004 Troyes Cedex, France

Outline

1. Introduction to vehicle routing problems
2. Brief history of route-first cluster-second methods
3. Basic splitting procedure
4. Application to constructive heuristics
5. Application to metaheuristics
6. Extensions to other vehicle routing problems

Part 1

Introduction to
Vehicle Routing Problems

Vehicle routing problems – VRPs

Important research area initiated by "The truck dispatching problem" (Dantzig & Ramser, 1959).

Exponential growth: 480 references for 1960-1999, 863 for 2000-2006, 3545 for 2007-2013 (Scopus).

Important applications in logistics (not only).

Important laboratory-problems. Laporte (2009): "The study of the VRP has given rise to major developments in the fields of exact algorithms and heuristics. In particular, sophisticated mathematical programming approaches and powerful metaheuristics for the VRP have been put forward in recent years."

Capacitated VRP – CVRP

The archetype of capacitated node routing problems:

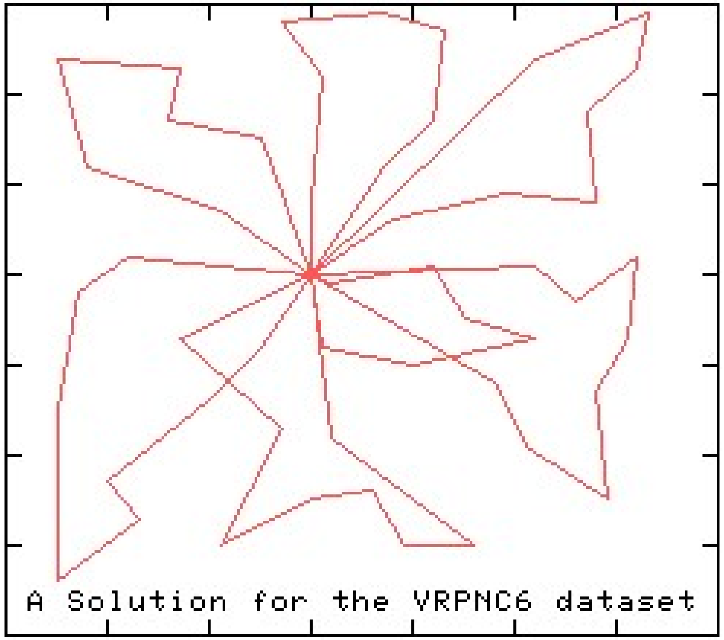
- a complete undirected network with $n+1$ nodes
- a depot (node 0) with identical vehicles of capacity Q
- other nodes 1 to n are customers with demands q_i
- each edge (i,j) has a traversal cost c_{ij}.

Goal: find a least-cost set of routes to visit all customers.

NP-hard: the Traveling Salesman Problem, known to be NP-hard, is a particular case with one vehicle.

Exact methods can reach $n = 200$ (Pecin et al., 2014). However, heuristics are required for most real instances.

Capacitated VRP - CVRP



Christofides-Mingozzi-Toth instance CMT-6, $n = 50$.
Optimal solution: total length 555.43 for 7 routes.

Capacitated Arc Routing Problem

Or CARP (waste collection, meter reading, etc.):

- undirected network $G = (V, E)$, in general not complete
- depot-node with identical vehicles of capacity Q
- subset E_R of m required edges (i, j) with demands q_{ij}
- edge costs c_{ij}
- for instance, street segments with amounts of waste.

Goal: find a least-cost set of routes to serve all required edges, in any direction. Edges can be traversed several times, including one traversal for service.

NP-hard. Exact methods $m = 100$ (Bode & Irnich, 2012).

Two strategies for VRP heuristics

VRP = partitioning problem + sequencing problem.

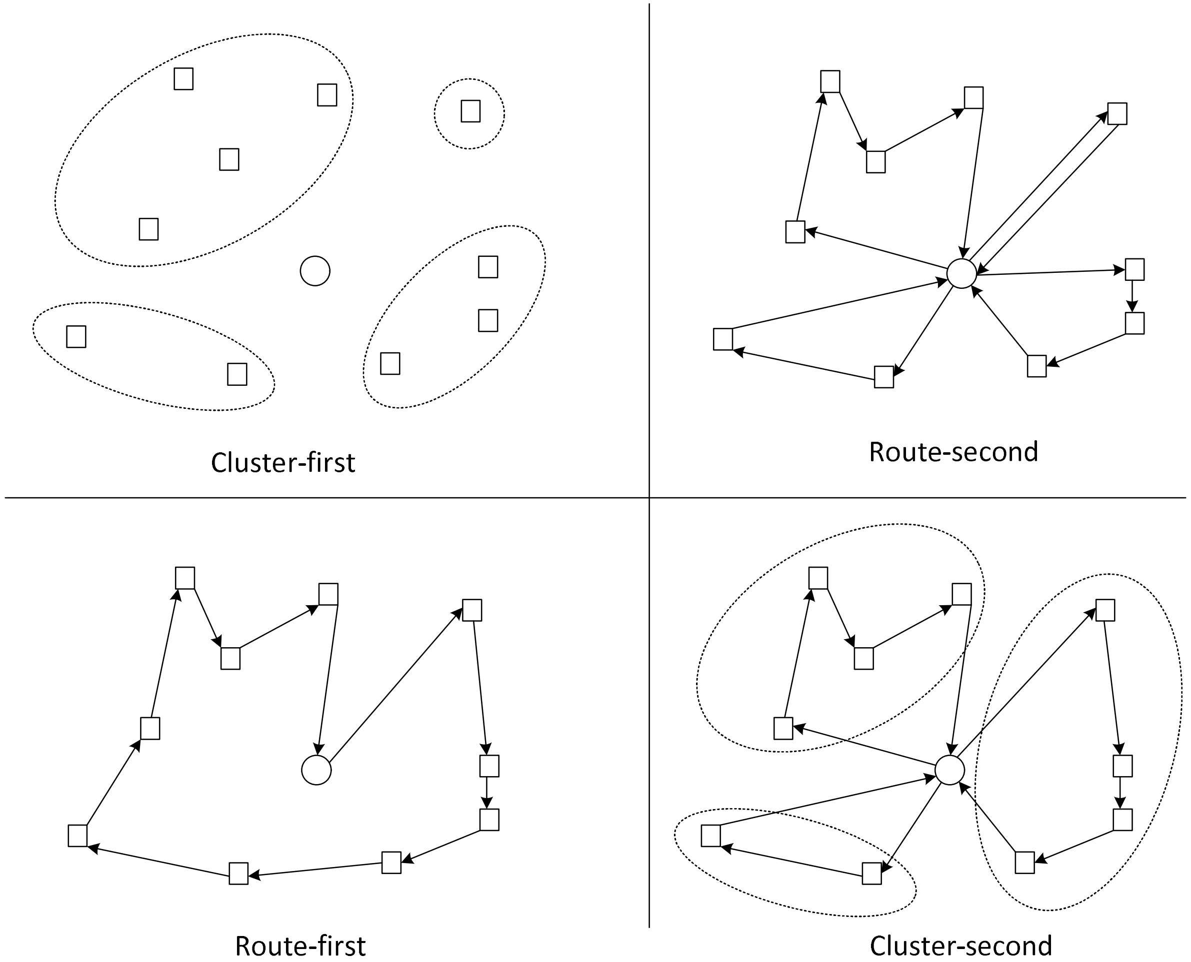
If partition first $\rightarrow$ "cluster-first route-second heuristics":

1. Build groups of nodes, one per vehicle
2. Solve one traveling salesman problem (TSP) per group

Sequence first $\rightarrow$ Route-first cluster-second heuristics:

1. Relax vehicle capacity to solve a TSP
2. This gives a TSP tour T, often called "giant tour"
3. Split this tour into trips satisfying capacity constraints.

Two strategies for VRP heuristics



Two strategies for VRP heuristics

Cluster-first route-seconds are well known (Gillett and Miller sweep heuristic, 1974) and instinctively employed by professional logisticians.

In contrast, route-first cluster-second approaches have been cited as a curiosity for a long time.

In a survey on VRP heuristics (2002), Laporte and Semet even wrote: "We are not aware of any computational experience showing that route-first cluster-second heuristics are competitive with other approaches."

Two strategies for VRP heuristics

So my goal is to show you that route-first cluster-second methods can give very good results on various VRPs.

Quite often, the TSP tour and its cost are not really used: we have an ordering of customers T (e.g., a priority list) and we want to split it optimally (subject to the ordering) into feasible routes.

So, I prefer to call VRP algorithms based on this principle

"order-first split-second methods"

Part 2

Brief history of route-first cluster-second methods

Brief history

Beasley (1983) shows that any TSP algorithm can be recycled for the CVRP, using an optimal splitting procedure called *Split*. But no numerical evaluation.

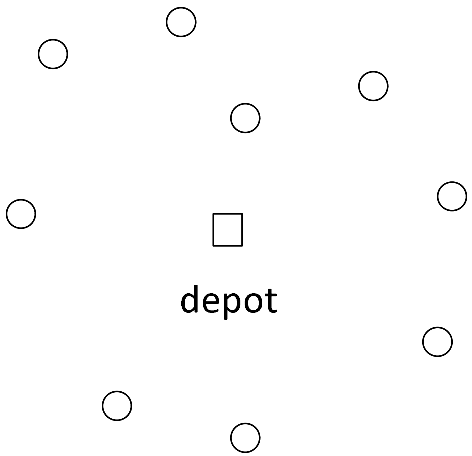
Ulusoy (1985) adapts *Split* to a CARP with heterogeneous vehicles. Results are provided for one instance only.

Theoretical results on worst deviations to the optimum:

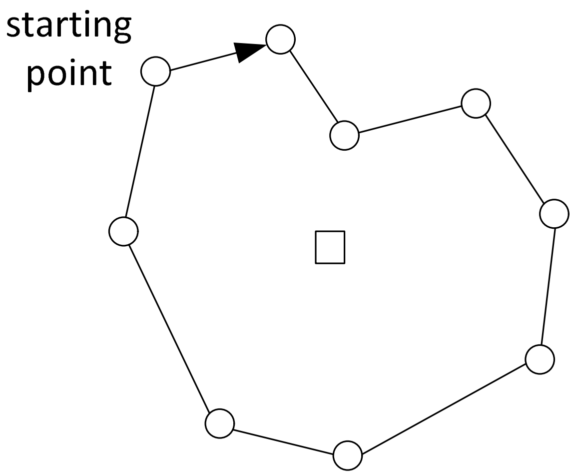
- Altinkemer & Gavish, 1990? CVRP with unit demands, compute an optimal TSP tour then *Split*: $2 - 1/Q$.
- Jansen (1993). Capacitated GRP, 1.5 approximation heuristic for the giant tour, then *Split*: $3.5 - 3/Q$.

Brief history

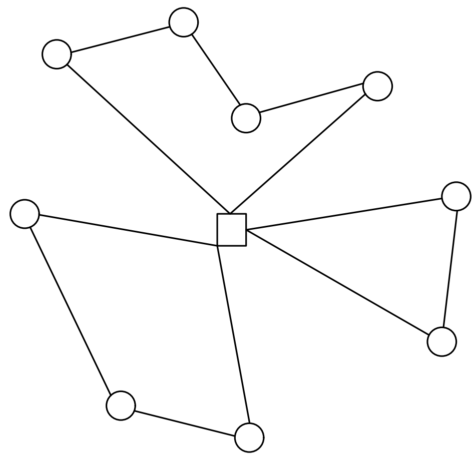
Ryan, Hjorring & Glover (1993) study 1-petals, routes where customers are in ascending or descending order of polar angle relative to the depot. Optimal 1-petals can be computed by splitting a giant tour.



Instance



Giant tour (polar angles)



1-petals after splitting

Brief history

Prins (2001), Lacomme, Prins & Ramdane-Chérif, 2001): memetic algorithms (hybrid GAs) for the CVRP and the CARP, chromosomes encoded as giant tours and decoded by *Split*. First GAs competing with tabu search methods.

2001-today. Split procedures designed for various VRPs and metaheuristics (GA, ILS, ACO…). Best metaheuristics are GA and ILS based on giant tours, and ALNS.

Long time after metaheuristics, Wøhlk (2008) and Prins, Labadie, Reghioui (2009) evaluate route-first cluster-second constructive heuristics for CVRP and CARP.

Brief history

"GAs: the return" (Vidal, Crainic, Gendreau, Prins, 2014). The best metaheuristic for 26 VRP variants becomes a hybrid GA with chromosomes encoded as giant tours and a generic split procedure, plus other tricks.

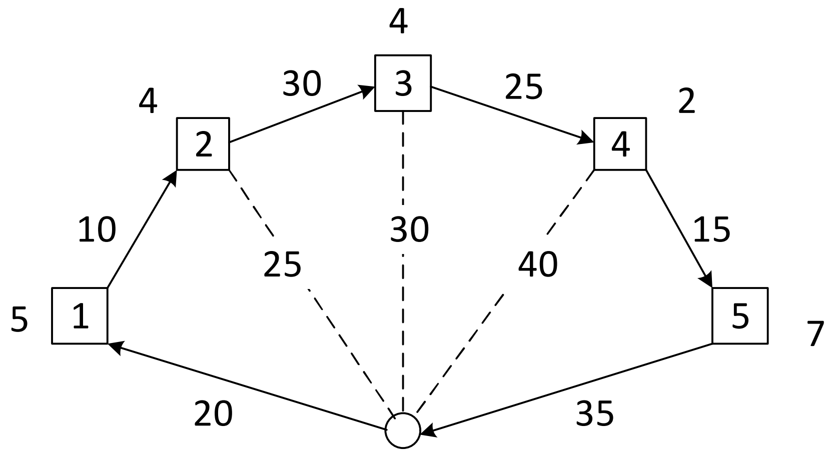
Prins, Lacomme and Prodhon (2014): a review in *Transportation Research Part C* found 74 articles on order-first split-second algorithms!

Note: nice paper for MSc and PhD students because algorithms and numerical examples are also provided.

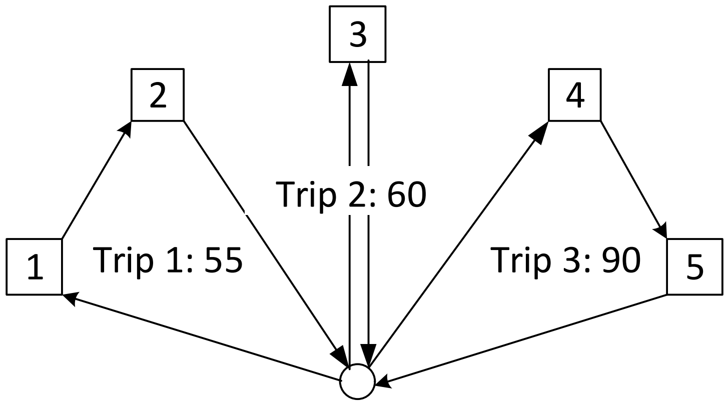
Part 3

Basic splitting procedure

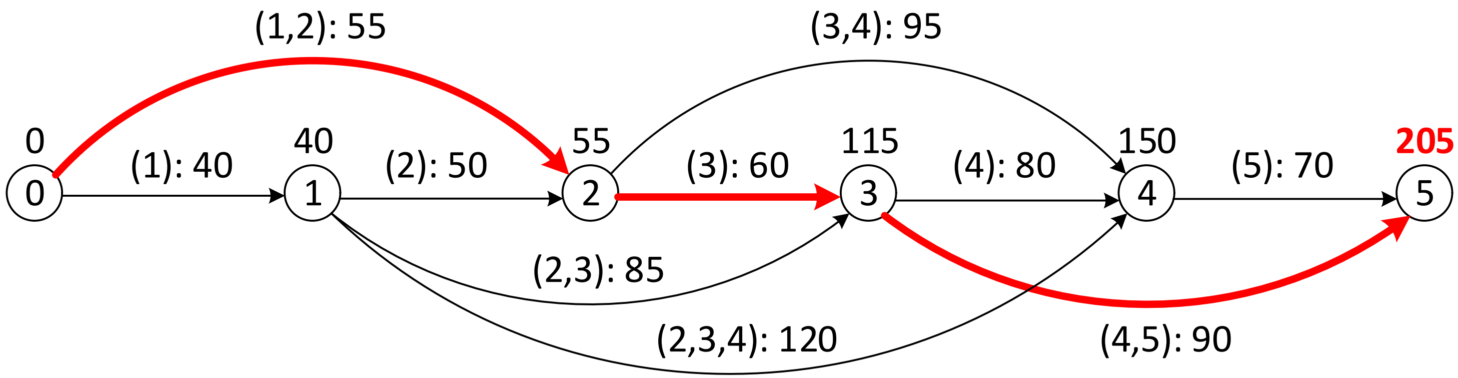
Basic splitting procedure (*Split*)



1. Giant tour T = (1, 2, 3, 4, 5) with demands

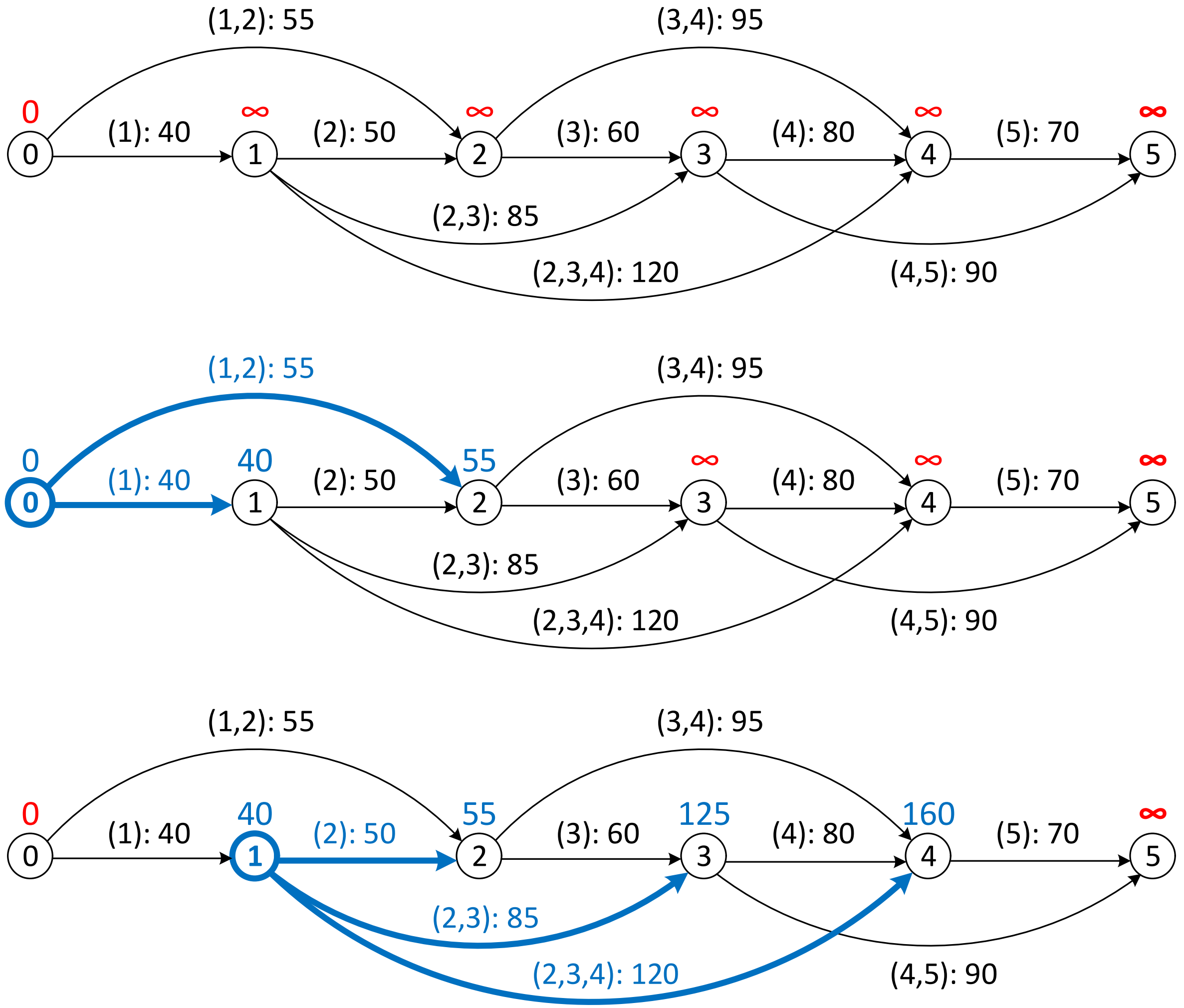


3. Optimal splitting, cost 205



2. Auxiliary graph H of possible trips for Q = 10 – Shortest path in bold

Shortest path – Bellman algorithm



Implementation

Auxiliary graph $H = (X, A, Z)$ with $n + 1$ nodes numbered rom 0.
A feasible route $(T_{i+1}, \dots, T_j)$ is modelled by arc $(i - 1, j)$.

Bellman's algorithm for directed acyclic graphs (DAGs).
Compact form with implicit auxiliary graph (Prins, 2004):

```
set V_0 to 0 and other labels V_i to ∞ (cost of path to node i)
for i ← 1 to n do
   for j ← i to n while subsequence/route (T_i, T_{i+1}, ..., T_j) feasible
      compute route cost, i.e., cost z_{i-1,j} of arc (i − 1, j)
      if V_{i-1} + z_{i-1,j} < V_j then
         V_j ← V_{i-1} + z_{i-1,j}
      endif
    endfor
endfor
```

Remarks

The giant tour T can be built using any TSP algorithm.

Optimal TSP tours do not necessarily lead to optimal CVRP solutions after splitting, good tours are enough.

However, *Split* is optimal, subject to the ordering of T.

$O(n^2)$ routes $(T_i, \ldots, T_j)$ are tested. Capacity and cost can be checked in $O(1)$ for each route: *Split* runs in $O(n^2)$.

More precisely, if b nodes per route on average, there are b outgoing arcs per node and *Split* runs in $O(nb)$.

For the CARP, T is a list of required edges with chosen directions, connected implicitly by shortest paths.

Part 4

Applications to constructive heuristics

Applications to constructive heuristics

Metaheuristics involving *Split* are known since 2001.
But evaluation on constructive heuristics is more recent.

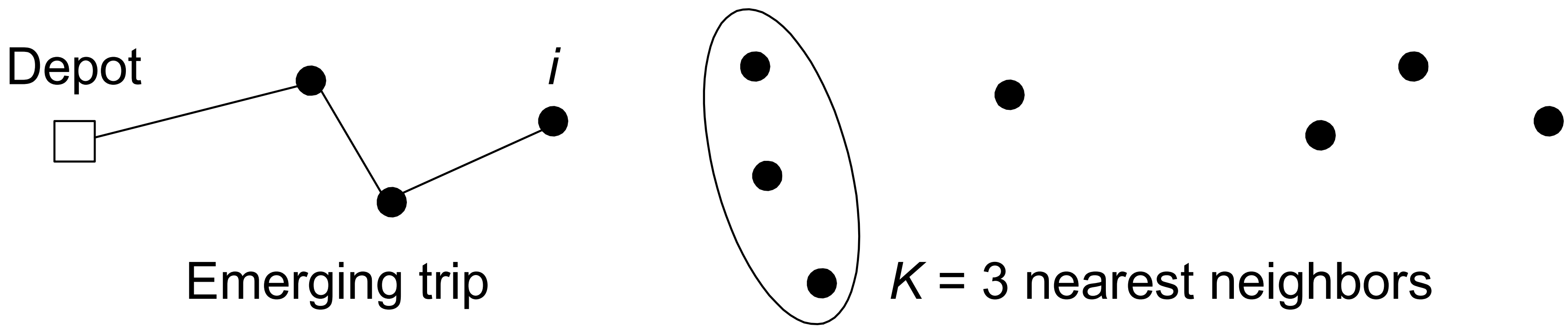
Prins, Labadie, Reghioui, Tour splitting algorithms for vehicle routing problems, *International Journal of Production Research*, 2009.

Comparison of splitting heuristics, randomized or not, with classical heuristics for CVRP and CARP:

- very simple heuristics to build giant tours
- randomized versions to generate several tours
- each tour is cut using different splitting procedures
- the best solution is returned at the end.

Randomized giant tours

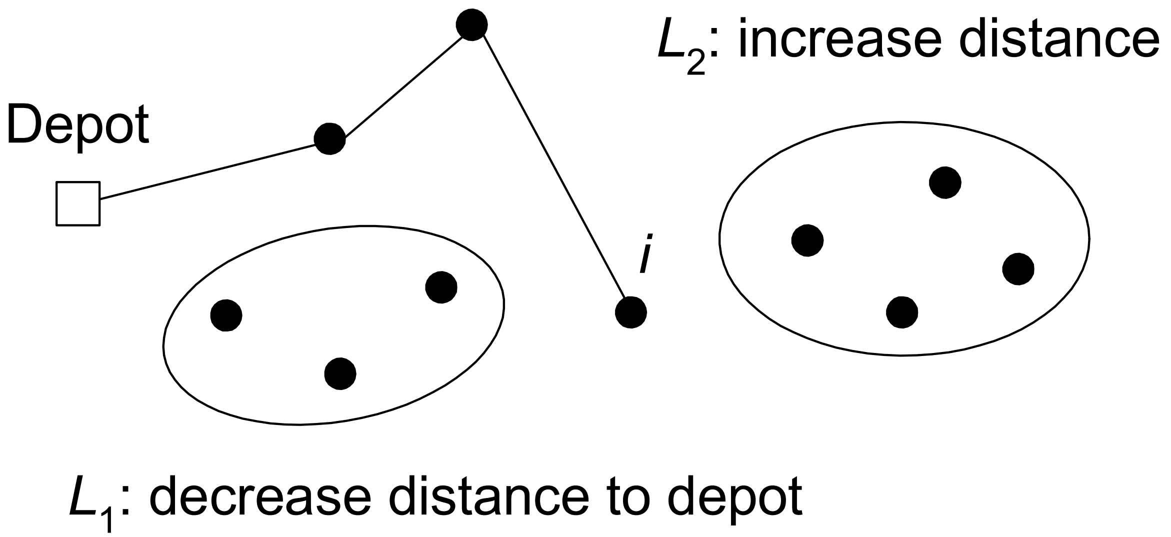
Nearest Neighbor heuristic (NN), well known for the TSP.
Randomized version:

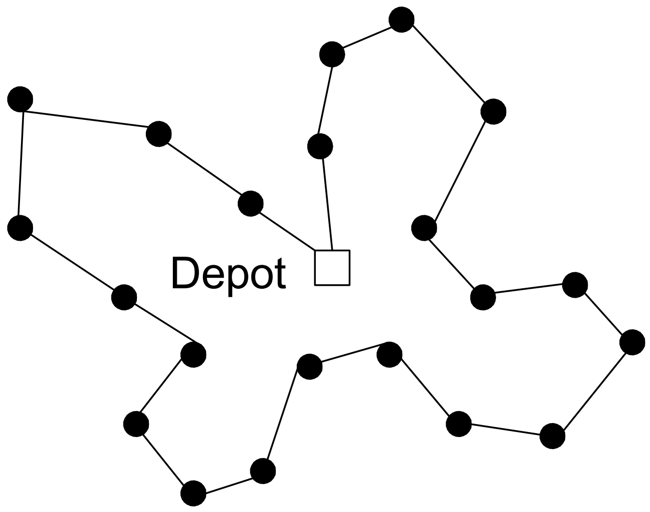


Draw the next client *j* among the *K* nearest ones.

Randomized giant tour

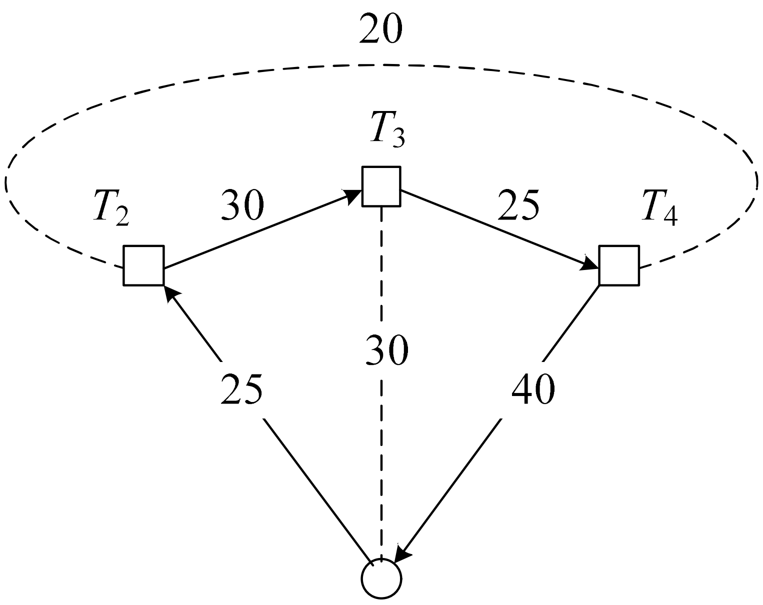
Nearest Neighbor randomized, "flower" version (NNF):



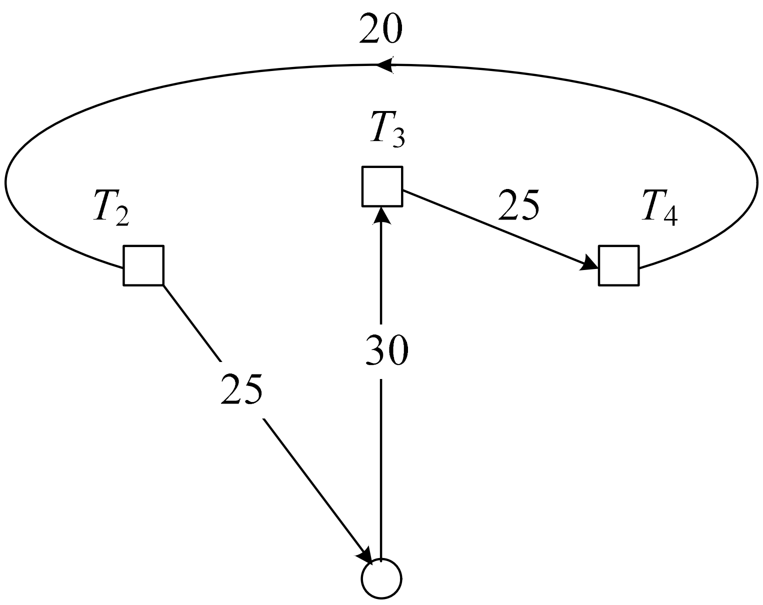


If load in $[kQ, (k+0.5)Q]$ for some k, draw next client j in L_2 else in L_1: higher probability to cut the tour when it is close to the depot!

Split with shifts (rotations)



Subsequence (T_2,T_3,T_4)

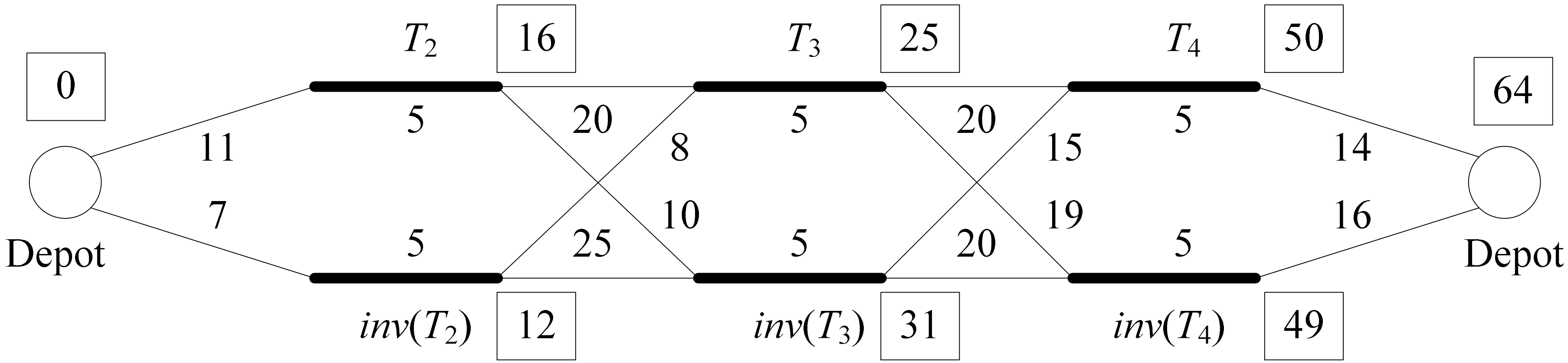


Best depot insertion (0,T_3,T_4,T_2,0)

The idea is to allow circular shifts of subsequences.
This is equivalent to a best insertion of the depot in the trip.
Trip (0,T_2,T_3,T_4): 120. (0,T_3,T_4,T_2,0): 100. (0,T_4,T_2,T_3,0): 120.

Split with flips for the CARP

Two directions per edge. How to select the best for each subsequence? Example for subsequence/route (T_2, T_3, T_4):



$inv(T_k)$ denotes the inverse (other direction) of edge T_k.
(T_2, T_3, T_4): cost 80, $(inv(T_2), T_3, inv(T_4))$: cost 64.
Note: it is also possible to combine shifts and flips!

Results for the CARP

Comparison with two classical constructive heuristics:

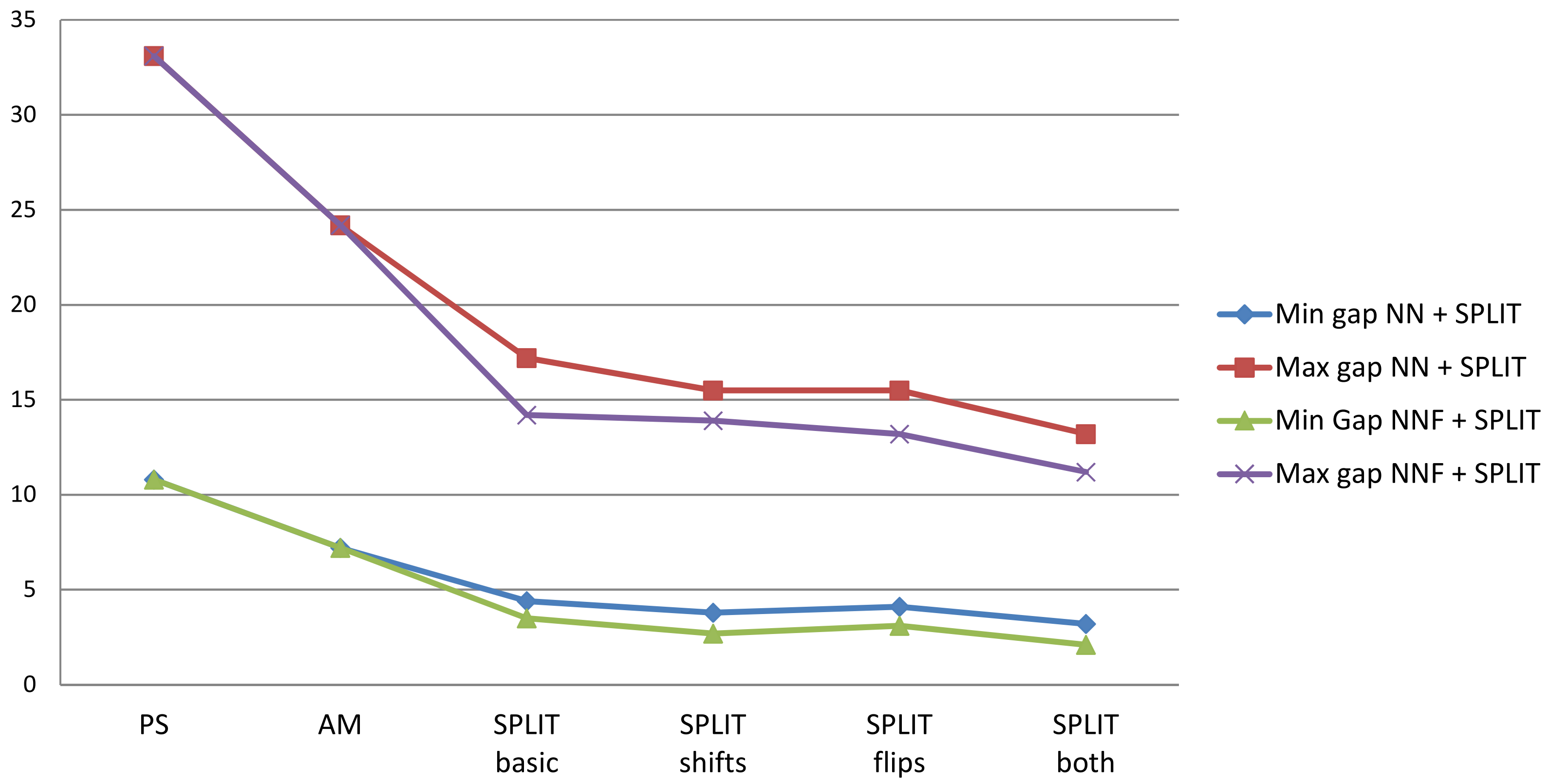
- Path-Scanning (PS) from Golden et al. (1983)
- Augment-Merge (AM) from Golden & Wong (1981)
- Using the 23 "gdb" instances ($n = 7\text{-}27$, $m = 11\text{-}55$)
- Optimal solutions are known for these instances.

Order-first split-second methods tested:

- build 20 giant tours using NN or NNF (randomized)
- apply *Split* (basic, with shifts, with flips, with both).

Running times are negligible: < 10 ms on a 3 GHz PC.

Average gap to the optimum in %



Conclusion: average and worst gaps better than PS and AM!
Best version NNF+Split with shifts & flips: 8 optima out of 23.
Using 10,000 giant tours, average 0.88%, 16 optima, 0.15s!

Part 5

Applications to metaheuristics

Split in metaheuristics

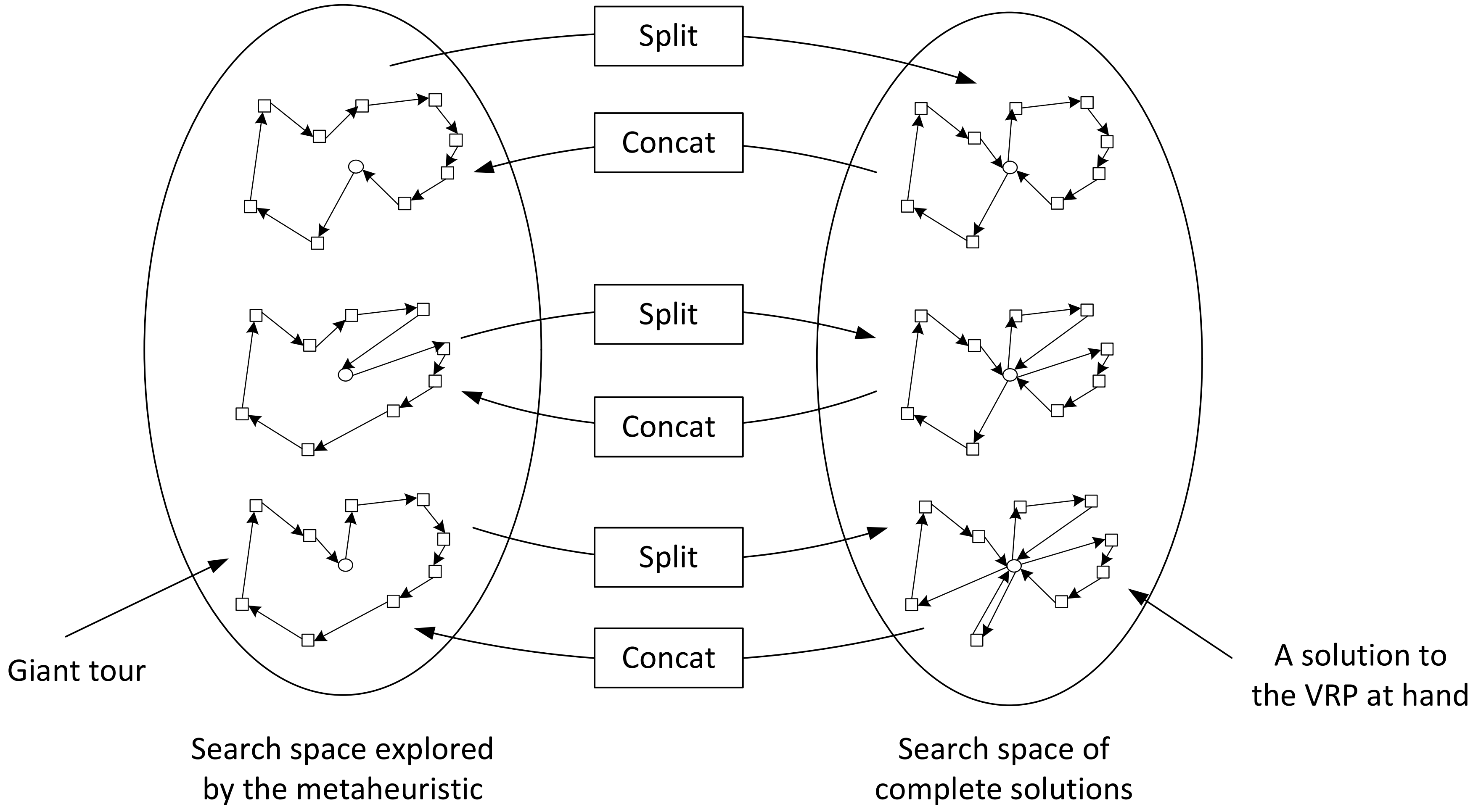
General principles:

- search the space of giant tours
- split current tour to get a solution for the VRP
- apply a local search to improve solution
- concatenate routes of solution to get a new giant tour.

Giant tours can be created by:

- constructive heuristics (at the beginning)
- crossover operators in genetic algorithms
- pheromone trails in ant colony optimization
- mutation/perturbation in GA and iterated local search
- concatenation of the routes of one VRP solution.

Split in metaheuristics



Advantages & drawbacks

No loss of information:

- *Split* cuts each tour optimally (subject to the sequence)
- and there exists at least one "optimal" giant tour.

Simplicity and efficiency:

- a smaller space (giant tours) is explored
- in GA, classical TSP crossovers can be reused
- no capacity violation, so no repair procedures
- *Split* + *Concat* act like a large neighborhood move!

Drawbacks:

- some problems may need tricky *Split* procedures
- additional running time of *Split* (but small in general).

Multi-start ILS (MS-ILS) for CVRP

ILS = Iterated Local Search
We describe a variant from Prins (2009) for the CVRP.

Basic ILS – See Lourenço et al. (2010) for a survey.

```
Heur(S)
Improve(S)
for iter = 1 to max_iter
   S' = Shake(S)
   Improve(S')
   if cost(S') < cost(S) then
      S = S'
   endif
endfor
return S
```

Generate a sequence of local optima with decreasing costs, using perturbation and local search.

Only 3 components:

- greedy heuristic: *Heur*
- perturbation procedure: *Shake*
- local search : *Improve*

MS-ILS for CVRP

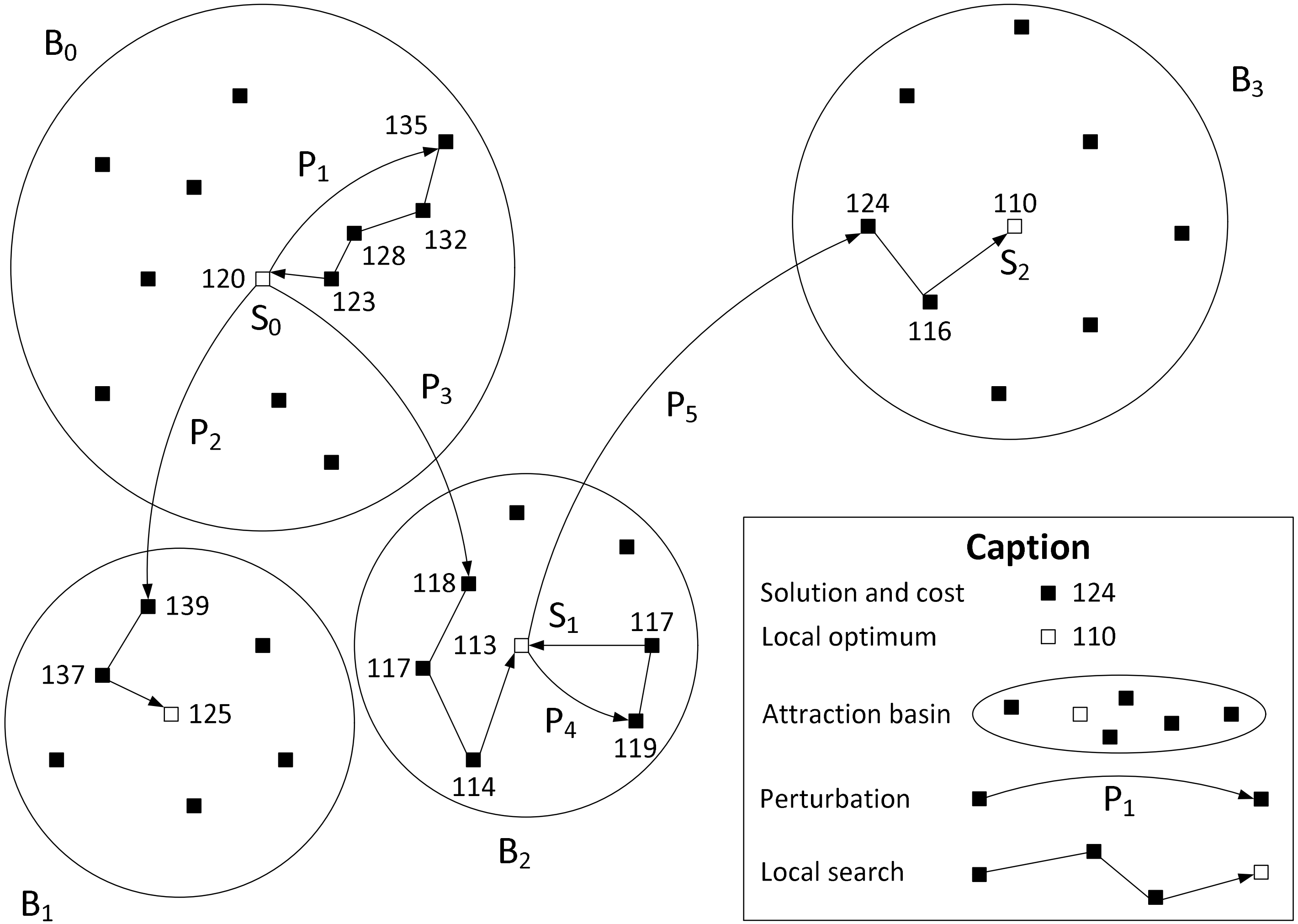
Why does it work? Proximality principle (Glover): local optima are often close to each other and they are grouped in clusters.

Perturbation: similar to mutations of genetic algorithms. Example: swap 2 customers at random current solution.

Some tuning is required:

- if the perturbation is too weak, the search stays in the attraction basin of S^*.
- if perturbation is too strong, S is too different from S^* and iterations become independent, like in a GRASP.
- the perturbation must not use a move of the local search, otherwise the local search can repair its effects!

MS-ILS for CVRP



MS-ILS for CVRP

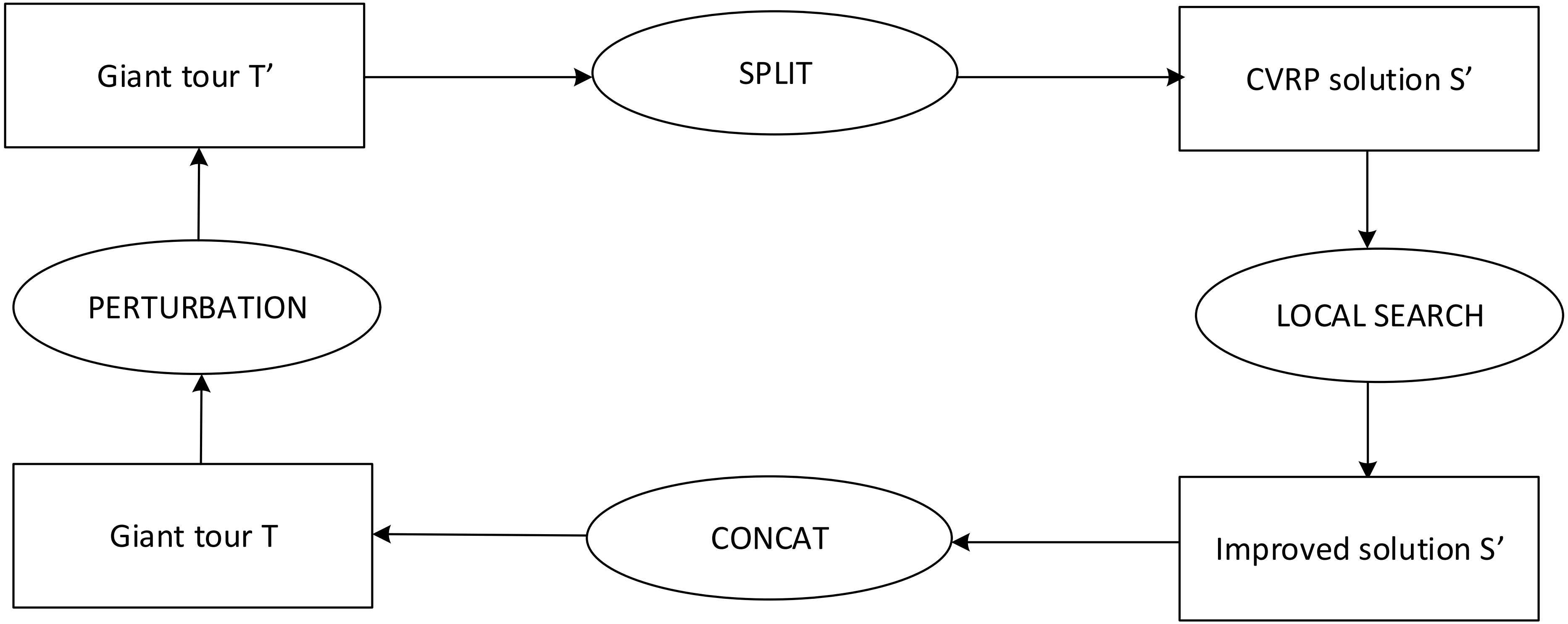
MS-ILS:

```
cost(S*) = ∞  //Global best
for start = 1 to nb_starts
   Randomized_Heur(S)
   Improve(S)
   for iter = 1 to max_iter
      S'= Shake(S)
      Improve(S')
      if cost(S') < cost(S)
      then S = S' endif
   endfor
   if cost(S) < cost(S*)
   then S* = S endif
end for
return S*
```

MS-ILS with giant tours:

```
cost(S*) = ∞  //Global best
for start = 1 to nb_starts
   Randomized_Heur(S*)
   Improve (S); Concat (S,T)
   for iter = 1 to max_iter
      T'= Shake(T); Split (T',S')
      Improve(S')
      if cost(S') < cost(S) then
         S = S'; Concat (S,T)
      endif
   endfor
   if cost(S) < cost(S*)
   then S* = S end if
end for
```

MS-ILS for CVRP



Cyclic alternation giant tours ↔ CVRP solutions

MS-ILS for CVRP

The components are relatively simple:

1. Heuristic to provide the initial solutions for each start:

- Randomized Clarke & Wright heuristic.
- Mergers are inspected in decreasing order of savings.
- Current merger is executed with probability $\alpha = 0.85$.

2. Perturbation with adaptive strength:

- k random exchanges of customers in the giant tour, at the beginning $k = k_{min} = 1$.
- After local search, if current solution S is not improved, k is incremented but without exceeding $k_{max} = 4$.
- W=Each time S is improved, k is reset to k_{min}.

MS-ILS for CVRP

3. Local search. Standard moves:

- Replace 2 edges by 2 others (2-opt moves)
- Relocate a string of up to 3 customers (Or-opt moves)
- Exchange two strings of up to 3 customers (λ-interchanges)

Implementation of moves:

- Moves applied to one or two routes
- Moved strings can be inverted when reinserted
- At each iteration, first improving move found is executed
- Efficient speed-up technique (Irnich et al., 2006).

MS-ILS for CVRP

14 "CMT" instances with 50-199 customers.

In Cordeau et al., "New heuristics for the VRP" (2005):
12 metaheuristics for the VRP are compared.

Four methods < 0.3% to best-known solutions using one run:

- AGES (Active Guided ES) Mester & Bräysy (2007).
- Bone Route, Tarantilis and Kiranoudis (2002).
- SEPAS (Solutions Elite PArts Search), Tarantilis (2005).
- Memetic algorithm, Prins (2004).

MS-ILS for CVRP

Method	AGES	ILS	Bone Route	SEPAS	MA
Dev. BKS%	0.027	0.071	0.183	0.196	0.236
BKS found	13	10	11	9	8
Time (s)	163	16	62	67	154

BKS = best-known solutions. Times scaled for a 2.8 GHz PC.

The ILS outperforms the other methods except AGES, while being simpler (less components) and much faster.

Since this MS-ILS of 2009, my PhD student Thibaut Vidal published in 2014 a generalization of my MA, which is now the best metaheuristic for the CVRP.

Other examples

Problem	Method	Reference
CVRP	Hybrid GA (MA)	Prins (2004)
CARP	Hybrid GA (MA)	Lacomme et al. (2004)
Mixed CARP	Hybrid GA (MA)	Belenguer et al. (2006)
Periodic CARP	Scatter Search	Chu et al. (2006)
Split delivery CVRP	Hybrid GA (MA)	Boudia et al. (2007)
CVRP	Multi-start ILS	Prins (2009)
Heterogeneous fleet VRP	Hybrid GA (MA)	Prins (2009)
Cumulative VRP	Hybrid GA (MA)	Ngueveu et al. (2010)
CARP	ACO	Santos et al. (2010)
CVRP with 2D-loading	Multi-start ILS	Duhamel et al. (2011)
Truck & trailer routing pb	Evolutionary PR	Villegas et al. (2011)
2-echelon LRP	GRASP+PR	Nguyen et al. (2012)
Multi-depot periodic VRP	Hybrid GA (MA)	Vidal et al. (2012)
26 VRP variants	Hybrid GA (MA)	Vidal et al. (2014)

Part 6

Extensions to other vehicle routing problems

Simple extensions

The steps in red are easily adapted to various VRPs.
In general, the low $O(nb)$ complexity can be kept.

set V_0 to 0 and all other labels V_i to ∞
for $i \leftarrow 1$ **to** n **do**
  for $j \leftarrow i$ **to** n **while** route $(T_i, T_{i+1}, \ldots, T_j)$ is feasible
    compute route cost (cost $z_{i-1,j}$ of arc $(i-1, j)$)
    if $V_{i-1} + z_{i-1,j} < V_j$ **then**
      $V_j \leftarrow V_{i-1} + z_{i-1,j}$
    end if
  end for
end for

Examples of simple extensions

CVRP. Feasibility test: discard trips with loads > Q.

Maximum trip length or duration$L$.
Feasibility test: discard trips of length > L.

Multi-depot VRP (MDVRP), D set of uncapacitated depots.
Route cost: to begin and end each route $(T_i, T_{i+1}, \dots, T_j)$, use the depot $d = \arg\min \left\{ c_{k,T(i)} + c_{T(j),k} \,\middle|\, k \in D \right\}$.

Examples (continued)

VRP with Time Windows (VRPTW):

- Feasibility test: discard routes violating time windows.
- Route cost: add waiting times if the goal is total time.

Vehicle Fleet Mix Problem (VFMP):

- p vehicle types, type t has capacity Q_t and fixed cost F_t
- Feasibility test: discard trips with loads > $\max\{Q_t\}$
- Arc cost: add F_k, k cheapest type with enough capacity

Relaxation of feasibility constraints

Some authors relax some feasibility constraints, but partially to avoid too many arcs in the auxiliary graph:

- Vidal et al. (2012) accept route loads up to $2Q$ (with a penalty) in a hybrid GA for the CVRP and MDVRP.
- Mendoza et al. (2010) do the same in a hybrid GA for a multi-compartment VRP with stochastic demands.

In these examples, penalties are reduced using a local search, called after *Split*.

Cases requiring another algorithm

For some VRPs, the auxiliary graph is identical but requires a different shortest path algorithm:

- Balanced trips. Update label of node j if $\max\{V_{i-1}, z_{i-1,j}\} < V_j$ (min-max shortest path).
- Limited fleet size *K*. Compute a shortest path with at most *K* arcs (general form of Bellman algorithm).

However, these algorithms are still fast:

- $O(nb)$ in the first case, like the basic *Split*
- $O(nbK)$ in the second case.

(b average nb of clients per feasible subsequence).

The limit: NP-hard cases for *Split*

Split can be hard when routes share limited resources.

Heterogeneous fixed fleet VRP (HFFVRP): p vehicle types, type t has only a_t vehicles of capacity Q_t and fixed cost F_t. *Split* must assign one vehicle to each arc (route) but the paths must use at most a_t vehicles for each type t.

NP-hard resource-constrained shortest path problem!
Fortunately, pseudo-polynomial algorithms are possible, using multiple labels per node.

E.g., *Split* for the HFFVRP runs in $O(mn^p)$ (Prins, 2009).

Other cases: MDVRP with capacitated depots etc.

Concluding remarks

The order-first split-second principle is general & flexible.

It can be used to design efficient constructive heuristics and metaheuristics: 74 references in (Prins et al., 2014).

Some theoretical results (performance guarantees) exist, see for instance (Jansen, 1993) and (Wøhlk, 2008).

The current best metaheuristic for 26 VRP variants is a hybrid GA based on this principle (Vidal et al., 2014).

Concluding remarks

Splitting a giant tour can be done exactly (subject to the sequence) and in most cases in polynomial time.

Current limit when the underlying shortest path problem is no longer polynomial (Heterogeneous Fixed Fleet VRP).

Two interesting research directions:

1. Faster splitting for hard cases. Use recent advances on large resource-constrained shortest path problems?
2. Tour splitting algorithms use two search spaces. A better exploitation of this feature is probably possible. For instance, when to switch to the other space?