Transportation Research Part C 40 (2014) 179–200

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Overview Paper

Order-first split-second methods for vehicle routing problems: A review

Christian Prins [a,*], Philippe Lacomme [b], Caroline Prodhon [a]

[a] Institut Charles Delaunay/LOSI – UMR CNRS 6281, Université de Technologie de Troyes, CS 42060, 12 rue Marie Curie, 10004 Troyes, France
[b] LIMOS – UMR CNRS 6158, Université Blaise Pascal, Campus des Cézeaux, 63177 Aubière Cedex, France

A R T I C L E I N F O

Article history:
Received 23 January 2013
Received in revised form 14 January 2014
Accepted 15 January 2014

Keywords:
Vehicle routing
Metaheuristic
Route-first cluster-second heuristic
Order-first split-second heuristic
Splitting procedure

A B S T R A C T

Cluster-first route-second methods like the sweep heuristic (Gillett and Miller, 1974) are well known in vehicle routing. They determine clusters of customers compatible with vehicle capacity and solve a traveling salesman problem for each cluster. The opposite approach, called route-first cluster-second, builds a giant tour covering all customers and splits it into feasible trips. Cited as a curiosity for a long time but lacking numerical evaluation, this technique has nevertheless led to successful metaheuristics for various vehicle routing problems in the last decade. As many implementations consider an ordering of customers instead of building a giant tour, we propose in this paper the more general name of ordering-first split-second methods. This article shows how this approach can be declined for different vehicle routing problems and reviews the associated literature, with more than 70 references.

1. Introduction

Vehicle routing problems are widespread in various activities like distribution, waste management, city logistics, meter reading, and inspection of power lines. Their study constitutes a very active research domain in which significant advances have been realized in the two last decades. A considerable number of variants have been studied to cope with various features and constraints, like hard time windows (Repoussis and Tarantilis, 2010), soft time windows (Figliozzi, 2010), dynamic allocation of swap containers (Huth and Mattfeld, 2009), configurable vehicle capacities (Qu and Bard, 2013), or stochastic demands (Bertazzi et al., 2013).

The solution techniques are also quite diversified, including for instance branch-and-cut algorithms (Bettinelli et al., 2011), metaheuristics like tabu search (Badeau et al., 1997) and even simulation (Juan et al., 2011). As most vehicle routing problems are computationally intractable, the current limit of exact algorithms is around one hundred customers and heuristic approaches are thus required for solving the much larger instances met in many industries.

Solving vehicle routing problems involves two kinds of decisions: partitioning the customers into clusters compatible with vehicle capacity and sequencing the customers in each cluster to get a route. A classical approach for constructive heuristics dedicated to the *capacitated vehicle routing problem* (CVRP) is based on the *cluster-first route-second* principle, in which the partition is determined first. A traveling salesman problem (TSP) is then solved for each cluster, exactly or heuristically. Two good examples are the sweep algorithm, commonly attributed to Gillett and Miller (1974), and the Fisher and Jaikumar (1981), where clusters are obtained solving a generalized assignment problem.

* Corresponding author. Tel.: +33 611051120.
E-mail addresses: christian.prins@utt.fr (C. Prins), placomme@isima.fr (P. Lacomme), caroline.prodhon@utt.fr (C. Prodhon).

http://dx.doi.org/10.1016/j.trc.2014.01.011

In the same vein, Beasley (1983) introduced *route-first cluster-second* heuristics, in which the two phases are inverted: vehicle capacity is temporarily relaxed to compute a TSP tour covering all customers, often called *giant tour*, which is then decomposed into feasible vehicle routes. Fig. 1, in which a circle symbolizes a depot while a square indicates a customer to visit, illustrates both the cluster-first route-second and the route-first cluster-second concepts for a vehicle routing problem.

Some advantages stand out in the second approach. Some users may prefer the computation of a giant tour to a clustering algorithm in the first phase. Moreover, Beasley (1983) observed that the second phase can be solved exactly as a shortest path problem in an auxiliary graph, but without reporting numerical experiments. Despite this interesting property, almost twenty years later, Laporte and Semet (2002, p. 121) wrote in a survey on CVRP heuristics: "We are not aware of any computational experience showing that route-first, cluster-second heuristics are competitive with other approaches".

Since 2002, comments have changed. In fact, the route-first cluster-second approach has led in the last decade to successful constructive heuristics and metaheuristics for node routing problems like the CVRP, but also for arc routing problems like the *capacitated arc routing problem* (CARP), where a subset of arcs or edges must be serviced. The main reasons for this growing success are a smaller solution space for metaheuristics (they search the set of giant tours instead of the much larger set of CVRP solutions), flexibility (many additional constraints can be handled) and efficiency (state of the art metaheuristics based on this approach are now available for many vehicle routing problems).

The purpose of this paper is to recall the basic route-first cluster-second approach, to show how it can be implemented efficiently, to see how it can be used in constructive heuristics and metaheuristics, and to review the literature on the numerous extensions published after Beasley's seminal article, with more than 70 papers. As we shall see, the giant tour determined in the first phase is seldom used and some algorithms rather consider an ordering of customers or a priority list before building the routes. This is why we prefer to speak about *order-first split-second methods* in the title and in the sequel. Moreover, as the ordering can be obtained in a variety of ways like heuristics, crossovers and mutation operators, the article mainly focuses on the splitting phase.

The paper is organized as follows. Section 2 describes a general frame for using splitting procedures, beyond a simple usage in constructive heuristics, underlines its advantages and introduces a classification of related published papers. Sections 3–6 are dedicated to each proposed class, with known utilizations in constructive heuristics and metaheuristics. Section 3 defines some notations, recalls the two classical capacitated routing problems (CVRP and CARP) and provides an efficient implementation of the basic splitting procedure (called *Split*) for these problems and some variants. Section 4 is devoted to simple extensions of the basic *Split*, where the construction of the auxiliary graph is affected but not the shortest path computation. More complicated versions with a different shortest path algorithm must be applied are exposed in Section 5. Some cases requiring a more general auxiliary graph are described in Section 6. The main advantages and current limitations of the approach are recapitulated in Section 7, before concluding remarks in Section 8.

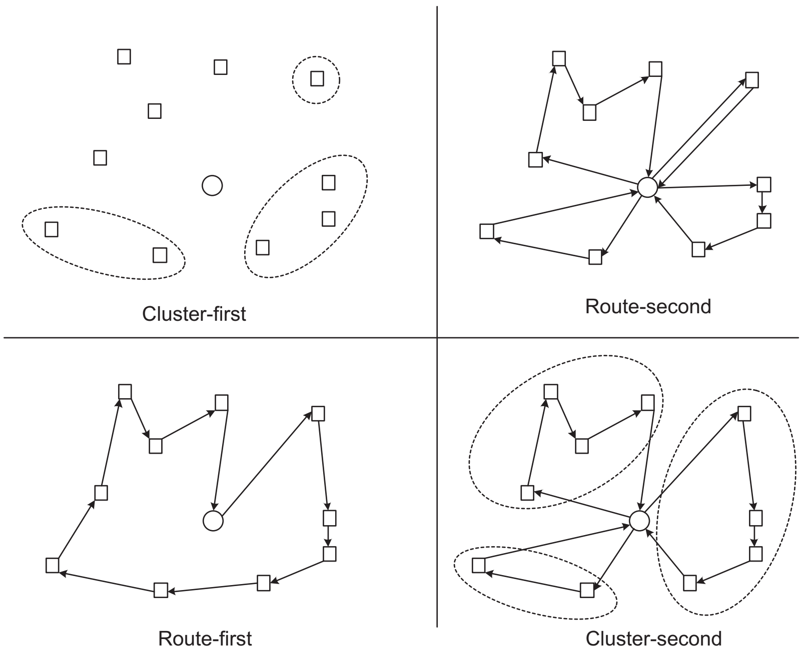


Fig. 1. Examples of two-phase procedures in vehicle routing.

2. Utilization context of splitting procedures and proposed classification

This section describes a general framework for using splitting procedures, explains its advantages and introduces a classification of the literature, based on different extensions of the basic case.

2.1. A general framework for splitting procedures in vehicle routing

In the context of vehicle routing problems, apart from a few works studying approximation results or aiming at recycling TSP algorithms to solve the CVRP, most order-first split-second methods generate in a first phase an *indirect solution representation* (ISR), often called *giant tour* or *task ordering*. Feasible routes are deduced from this ISR in a second phase. This approach offers the following advantages: (i) any solution of the routing problem has an indirect representation; (ii) using a splitting procedure, each ISR can be decoded into a solution to the original problem, and this splitting can be made optimally (see Section 3.2); and (iii) there exists at least one "optimal" ISR, i.e., one that gives an optimal solution to the original problem after splitting.

To check the latter property, consider an optimal solution S to the vehicle routing problem to be studied and concatenate the customers of its successive trips to get a giant tour T. This transformation is called $Split^{-1}$ in the sequel. The optimal solution S will be found again if the splitting procedure is applied to T.

Several effective metaheuristics, based on this approach depicted in Fig. 2, have been published. As an optimal solution (subject to the sequence) can be deduced from each giant tour and as optimal tours exist, such metaheuristics can search the smaller space of ISRs without losing information.

More generally, an iterative search based on ISRs can involve a local search procedure for intensification, in general applied to a complete solution S. For vehicle routing problems, the resulting solution S' can be converted into a new giant tour using the procedure $Split^{-1}$ explained before. Several published works include also a perturbation or mutation mechanism to avoid premature convergence to low-quality local optima. This can be done in the ISR space since a small modification in a giant tour may have a stronger impact on the decoded solution. To take full advantage of the indirect representation of solutions, the search can even alternate cyclically between the two search spaces (giant tours and complete solutions). Such a method, depicted in Fig. 3, is used in the most successful metaheuristics involving splitting procedures.

2.2. Proposed classification of literature

The seminal works on order-first split-second methods have been applied to the CVRP and the CARP at the beginning of the 1980s. In a first step, an ordered sequence of *tasks* (required nodes or edges) is built by relaxing vehicle capacities. The resulting sequence is partitioned into feasible routes in a second step. As we shall see in Section 3, the second step is equivalent to computing a least-cost path in an auxiliary graph in which each arc models one possible trip (subsequence of tasks in the giant tour). The path can be computed using Bellman's algorithm for directed acyclic graphs. The approach has been

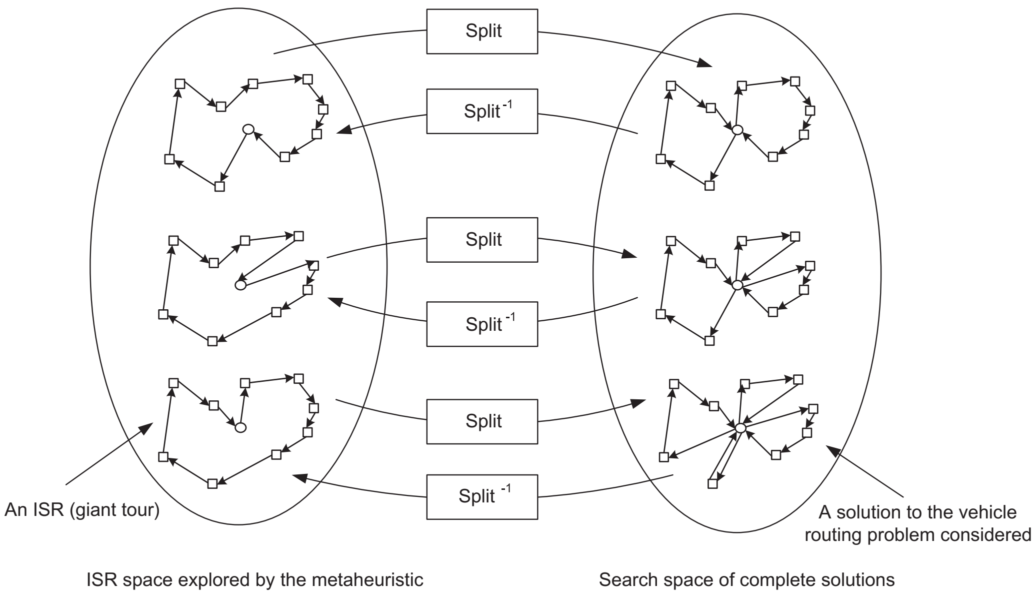


Fig. 2. Metaheuristic operating on the smaller search space of ISRs.

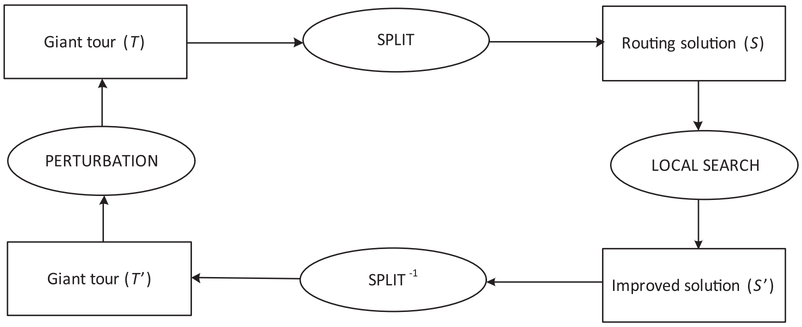


Fig. 3. Alternation between the set of giant tours and the set of complete solutions in metaheuristics.

extended to vehicle routing problems involving more complex networks, extra constraints, and additional decisions like depot location or vehicle selection.

An analysis of literature allowed us to detect more than 70 articles involving splitting procedures. They are gathered in Table 1, with a proposed classification based on four classes:

- *Basic Split*: This class relies on the basic *Split* procedure presented in Section 3. It includes the seminal works on the CVRP, the CARP and closely related problems (two objectives, coexistence of required nodes and required edges) but without additional constraints. Two improved versions called *Split with Flips* and *Split with Shifts* are also described.
- *Simple extensions*: This class surveyed in Section 4 includes many problems with additional constraints. These constraints eliminate some arcs in the auxiliary graph, because the corresponding routes are infeasible, and/or change the way in which route costs are computed. Nevertheless, once the graph is ready, Bellman's algorithm can still be used.
- *Shared resources*: In this class reviewed in Section 5, the routes compete for limited resources like capacitated depots or a fixed heterogeneous fleet of vehicles. Apart from a few polynomial cases, the resource consumptions associated with the arcs in the auxiliary graph raise hard shortest path problems. More complex label-based shortest path algorithms must be used.
- *Special graphs*: This last category presented in Section 6 concerns approaches which can still be considered as order-first split second methods, but involving special auxiliary graphs.

3. Basic *Split*

This section first recalls the CVRP and the CARP, the two basic NP-hard problems with capacitated vehicles in node and arc routing, and introduces some notation. Then it presents the principle of the basic splitting procedure to decompose a giant tour, provides a compact algorithm and analyses its complexity. Finally, the main applications of the basic *Split* are presented.

3.1. CVRP and CARP: the first vehicle routing problems handled by splitting procedures

The CVRP met for instance in distribution is defined on a complete undirected graph $G = (X, E)$. The node-set X contains one depot (0) and n customers indexed from 1 to n. A fleet of identical vehicles with capacity Q is based at the depot. In general the fleet size is not imposed: the number of vehicles used is a decision variable. Each customer i has a non-negative demand q_i. For any two nodes i and j, a $(n + 1) \times (n + 1)$ matrix C gives the cost c_{ij} (distance or duration) of a shortest path from i to j in the real network. Some instances involve service times s_i and a maximum trip length or working time L. The goal is to determine a least-cost set of trips, such that each trip begins and ends at the depot, each customer is visited exactly once and the total amount delivered by a vehicle respects its capacity.

The CARP, raised by applications like municipal refuse collection, is also defined on an undirected graph $G = (X, E)$ but this graph is often sparse. The edge-set E includes a subset E_R of n required edges, to be serviced by a vehicle. A deadheading cost d_e is incurred each time edge e is traversed by a vehicle without being serviced. Each required edge $e \in E_R$ has in addition a non-negative demand q_e and a service cost s_e, counted when the edge is traversed (in any direction) by a vehicle to be serviced. Like in the CVRP, a virtually unlimited fleet of identical vehicles with capacity Q is based at a depot node, the load of a vehicle may not exceed its capacity, and split services are prohibited. The aim is to find a minimum cost set of routes to service all required edges.

The following CARP encoding leads to similar splitting procedures for the two problems. It is based on a list A of $2n + 1$ arcs, indexed from 0 onward, containing one loop on the depot (index 0) and two opposite arcs (i, j) and (j, i) for each required edge $[i, j]$. These two arcs represent the two possible service directions of the edge. Each arc $u \neq 0$ is defined by

Table 1
Classification of published papers involving order-first split-second methods.

References in temporal order	Problem studied	Basic Split Section 3	Simple extensions Section 4	Shared resources Section 5	Special graphs Section 6
Part 1					
CARP and extensions					
Ulusoy (1985)	Fleet size and mix CARP		X		
Lacomme et al. (2001)	CARP	X			
Fleury et al. (2004)	Stochastic CARP	X			
Lacomme et al. (2004)	CARP	X		X	
Fleury et al. (2005)	Stochastic CARP	X			
Lacomme et al. (2005)	Periodic CARP			X	
Belenguer et al. (2006)	Mixed CARP	X			
Chu et al. (2006)	Periodic CARP			X	
Lacomme et al. (2006)	Bi-objective CARP	X			
Reghioui et al. (2007)	CARP with time-windows		X		
Fleury et al. (2008)	Bi-objective stochastic CARP	X			
Labadi et al. (2008a)	CARP with time-windows		X		
Labadi et al. (2008c)	Split delivery CARP	X			
Wøhlk (2008)	CARP	X			
Mourão et al. (2009)	Sectoring arc routing problem	X			
Prins et al. (2009)	CARP	X	X		
Tang et al. (2009)	CARP	X			
Belenguer et al. (2010)	CARP	X			
Kansou and Yassine (2010)	Multi-depot CARP		X		
Liu et al. (2010a)	CARP		X		
Santos et al. (2010)	CARP	X			
Xing et al. (2010)	Multi-depot CARP		X		
Mei et al. (2011)	Periodic CARP			X	
VRP and extensions					
Beasley (1983)	VRP	X			
Golden et al. (1984)	Vehicle fleet mix problem (VFMP)		X		
Haimovich and Rinnooy Kan (1985)	Unit-demand VRP	X			
Altinkemer and Gavish (1990)	Unit-demand VRP	X			
Ryan et al. (1993)	VRP				X
Renaud et al. (1996)	VRP				X
Salhi and Sari (1997)	Multi-depot VFMP		X		
Mosheiov (1998)	Unit-demand PDP	X			
Prins (2004)	VRP	X			
Sörensen (2006)	Bi-objective VRP	X			
Boudia et al. (2007)	Split delivery VRP	X			
Chang and Chen (2007)	VRP with time-windows		X		
Jozefowiez et al. (2007)	VRP with route balancing	X			
El Fallahi et al. (2008)	Multi-compartment VRP		X		
Labadi et al. (2008b)	VRP with time-windows		X		
Imran et al. (2009)	VFMP		X		
Jozefowiez et al. (2009)	VRP with route balancing	X			
Liu et al. (2009)	VFMP		X		
Prins (2009a)	VFMP and HVRP	X		X	
Prins (2009b)	VRP	X			
Prins et al. (2009)	VRP	X	X		

(*continued on next page*)

Table 1 (*continued*)

References in temporal order	Problem studied	Basic Split Section 3	Simple extensions Section 4	Shared resources Section 5	Special graphs Section 6
Velasco et al. (2009)	PDP		X		
Bontoux et al. (2010)	Generalized TSP				X
Bouly et al. (2010)	Team orienteering problem				X
Hamdi-Dhaoui (2011)	VRP		X		
Liu et al. (2010b)	VRP with task selection	X			
Mendoza et al. (2010)	Multi-compartment SVRP		X		
Ngueveu et al. (2010)	Cumulative VRP			X	
Duhamel et al. (2011a)	2-dimensional loading CVRP		X		
Duhamel et al. (2011b)	HVRP			X	
Ursani et al. (2011)	VRP with time-windows		X		
Duhamel et al. (2012)	HVRP			X	
Reiter and Gutjahr (2012)	Bi-objective VRP	X			
Velasco et al. (2012)	Bi-objective PDP		X		
Vidal et al. (2012)	Multi-depot periodic VRP			X	
Khebbache-Hadji et al. (2013)	2L-CVRP with time-windows		X		
Lacomme et al. (2013)	3-dimensional loading CVRP		X		
Vidal et al. (2013a)	VRP with time-windows		X	X	
Vidal et al. (2013b)	26 VRP variants				
Afsar et al. (2014)	Generalized VRP (GVRP)				X
Vidal et al. (2014)	Multi-depot VFMP	X	X		
Part 2					
LRP and extensions					
Nagy and Salhi (1996)	LRP	X			
Prins et al. (2006)	LRP	X			
Prodhon and Prins (2008)	Periodic LRP	X			
Duhamel et al. (2010)	LRP			X	
Villegas et al. (2010)	Truck and trailer routing pb				X
Duhamel et al. (2011b)	LRP			X	
Villegas et al. (2011)	Truck and trailer routing pb				X
Nguyen et al. (2012a)	Two-echelon LRP		X		
Nguyen et al. (2012b)	Two-echelon LRP		X		
Other routing problems					
Jansen (1993)	Capacitated general routing pb	X			
Prins and Bouchenoua (2004)	Node, edge and arc routing pb	X			

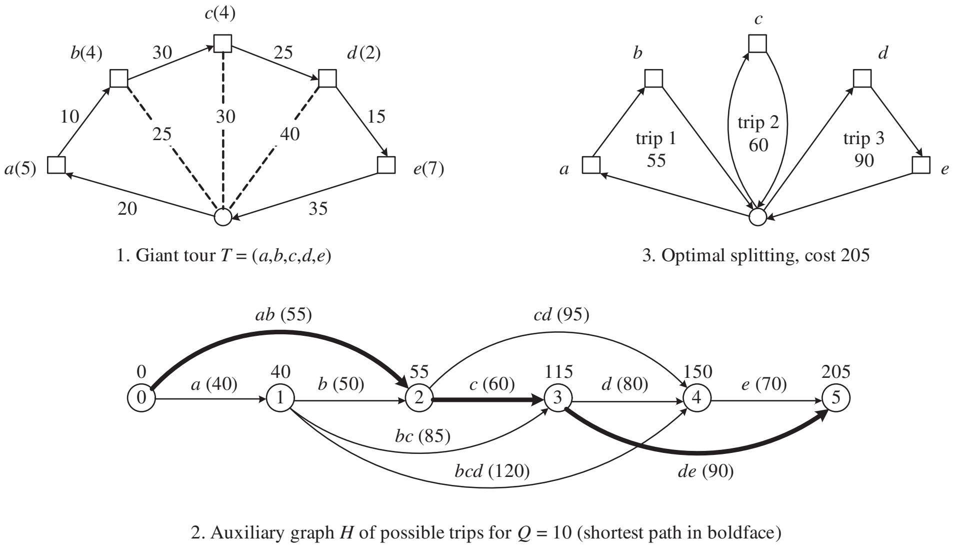


Fig. 4. Example of splitting procedure.

a demand q_u, a service cost s_u (inherited from the edge of origin) and the index $inv(u)$ of the opposite arc. The encoding is completed by a pre-computed $(2n+1) \times (2n+1)$ matrix C, indexed by arc numbers, in which c_{uv} is the cost of a shortest path (in terms of deadheading cost) from node e_u to node b_v in the original network.

Using this notation, a CVRP trip can be defined as a list of distinct customers $T = (T_1, T_2, \dots, T_k)$, visited in this order, with a total cost $c(0,T_1) + \sum_{i=1}^{k-1}(s(T_i) + c(T_i, T_{i+1})) + s(T_k) + c(T_k, 0)$. A giant tour T corresponds to the particular case $k = n$ since it must cover the n customers. For the CARP, a trip can be stored as a list $(T_1, T_2, \dots, T_k)$ of arc numbers, each arc representing one of the two directions of a required edge, but its cost can be computed using the same formula as above. A giant tour T for the CARP contains the n required edges, each one being represented by one of its two arcs. In the sequel, the word *task* will refer indifferently to a customer or a required edge.

3.2. Basic splitting process

Consider the CVRP to fix ideas and a given ordering $T = (T_1, T_2, \dots, T_n)$ of the n customers. As shown by Beasley (1983), an optimal splitting (subject to the sequence) can be obtained by computing a shortest path in an auxiliary graph $H = (Y, U)$. Y contains $n + 1$ nodes numbered from 0 to n. Each subsequence $(T_i, T_{i+1}, \dots, T_j)$ corresponding to a feasible trip is modeled in U by one arc $(i-1, j)$, weighted by the trip cost $cost(i,j) = c(0,T_i) + \sum_{k=i}^{j-1}(s(T_k) + c(T_k, T_{k+1})) + s(T_j) + c(T_j, 0)$. The optimal splitting corresponds to a shortest path from node 0 to node n in H.

Fig. 4 depicts a small example with five customers. The given ordering $T = (a, b, c, d, e)$ is materialized on the left as a giant tour, to show traveling costs between customers and between customers and the depot. The values in brackets are the demands. The auxiliary graph in the lower part is built assuming a vehicle capacity $Q = 10$. For instance, arc *ab* models a trip visiting customers a and b, with total cost 55, while arc *abc* is not present due to an excessive demand (13). The shortest path (thick arcs) has three arcs and cost 205. The resulting CVRP solution is given on the right.

The shortest path can be computed using Bellman's algorithm for directed acyclic graphs. Algorithm 1 is a compact version where the auxiliary graph is not generated explicitly (Prins, 2004). A version for the CARP can be found in Lacomme et al. (2004). Roughly speaking, two nested loops indexed by i and j inspect each subsequence $(T_i, T_{i+1}, \dots, T_j)$ and compute its total demand (*load*) and the trip cost (*cost*). The subsequence is ignored if it violates vehicle capacity. Otherwise, it corresponds to arc $(i-1, j)$ in the implicit auxiliary graph and label V_j of node j is updated when improved, i.e., if $V_{i-1} + cost < V_j$. This label (value on top of each node in the auxiliary graph of Fig. 4) is the cost of a shortest path from node 0 to node j. The cost of the CVRP solution can be found at the end in V_n.

The role of P_j in Algorithm 1 is to store the predecessor of node j on the shortest path leading to j. Algorithm 2 shows how to extract the CVRP solution S using these predecessors. S is coded as a list of trips, each trip being a list of customers.

Algorithm 1. A compact implementation of *Split* for the CVRP

```
V_0 ← 0
P_0 ← 0
for i ← 1 to n do V_i = ∞ endfor
for i ← 1 to n do
  j ← i
  load ← 0
  repeat
    load ← load + q(T_j)
    if i = j then
      cost ← c(0, T_i) + s(T_i) + c(T_i, 0)
    else
      cost ← cost − c(T_{j−1}, 0) + c(T_{j−1}, T_j) + s(T_j) + c(T_j, 0)
    endif
    if (load ⩽ Q) and (V_{i−1} + cost < V_j) then
      V_j ← V_{i−1} + cost
      P_j ← i − 1
    endif
    j ← j + 1
  until (j > n) or (load > Q)
endfor
```

Algorithm 2. Extraction of CVRP solution after *Split*

```
S = ∅
j ← n
repeat
  trip ← ∅
  for k ← P_j + 1 to j do
    add customer T_k at the end of trip
  endfor
  add trip at the beginning of S
  j ← P_j
until j = 0
```

The key-observation to establish the complexity of Algorithm 1 is that the load and cost for subsequence $(T_i, T_{i+1}, \ldots, T_{j+1})$ are deduced in $O(1)$ from the ones for $(T_i, T_{i+1}, \ldots, T_j)$, instead of browsing the whole subsequence. So, each subsequence is treated in $O(1)$ and the complexity of *Split* is proportional to the number $O(n^2)$ of feasible subsequences, which is also the number of arcs in the auxiliary graph H. If β denotes the average length (number of customers) of feasible subsequences, H has $n\beta$ arcs and we get a more precise complexity in $O(n\beta)$. In particular, the running time decreases in practice when the quotient q/Q increases, q denoting the average demand.

3.3. Application of basic Split to CVRP and CARP

3.3.1. Constructive heuristics and local search procedures

The *Split* method as described in Section 3.2 was originally introduced by Beasley in 1983 for the CVRP. This author proposed to relax vehicle capacity to solve a TSP, exactly or heuristically, and then to apply the *Split* procedure to the resulting tour. The goal of the paper was to show that any TSP algorithm can be recycled to solve the CVRP, not to provide computational results.

Some authors investigated greedy versions of *Split*: starting from one node, the giant tour is cut whenever the next node does not fit residual vehicle capacity. The result is no longer optimal. Haimovich and Rinnooy Kan (1985) described a tour partitioning (TP) heuristic for the VRP with unit demands: a TSP tour is cut into successive trips with Q customers. They also studied an iterated TP (ITP), in which TP is applied to each possible starting node to return the best solution. Altinkemer and Gavish (1990) showed that this ITP has a worst case error ratio of $2 - 1/Q$ if the TSP tour is optimal. Mosheiov (1998) derived similar heuristics for a pickup and delivery problem with unit demands.

Other interesting theoretical results exist for the optimal *Split* procedure. Jansen (1993) analyzed a two-phase heuristic called Shortest Optimal Tour Partitioning (SOTP) for the *capacitated general routing problem* (CGRP), a problem with required nodes and edges. A giant tour covering all tasks is determined using a 3/2 approximation algorithm and then split into feasible routes. Jansen shows that the approximation factor of SOTP is $7/2 - 3/Q$, provided all demands and Q are integral and $Q \geqslant 3$.

The basic *Split* respects the edge traversal directions specified by the giant tour. This paragraph and the next concern improved versions for the CARP, able to select the best traversal direction of each edge. Wøhlk (2008) showed that Jansen's approximation factor (1993) concerns the CARP too, since required nodes in the CGRP can be transformed into demand edges. She proposed a new heuristic A-ALG for the CARP, with the same approximation factor, but always at least as good as SOTP for the same giant tour. The tour construction remains identical but the partition is computed via a dynamic programming procedure which selects the best traversal direction for each edge in feasible trips. In fact, this procedure gives the same results as *Split with Flips*, described in the next paragraph. A numerical evaluation shows that A-ALG outperforms classical constructive heuristics for the CARP.

In a guided local search for the CARP, Beullens et al. (2003) described a move which relocates a subsequence of required edges in a trip, while determining the best direction of each edge. The same technique can be applied in *Split*, giving a version for the CARP called *Split with Flips* (Prins et al., 2009). Fig. 5 gives one example for a subsequence (T_2, T_3, T_4). As said in Section 3.1, each required edge is coded as two arcs linked by a pointer *inv*. The upper layer contains the directions defined by the giant tour while the lower layer contains the opposite arcs. Two copies of the depot are added as source and destination. Thin lines represent shortest paths in the real network and their costs. The best edge directions are given by a shortest path between the two depot copies (labels correspond to framed numbers). The best path $(inv(T_2), T_3, T_4)$ has a cost 64 instead of 80 for the original directions. It is possible to embed this computation in Algorithm 1 while preserving its $O(n\beta)$ complexity: the trick is to reuse the two labels computed for T_j and $inv(T_j)$ when testing subsequence $(T_i, T_{i+1}, \ldots, T_{j+1})$.

In the basic *Split*, the route associated to a subsequence of customers begins at the depot, visits the customers in the specified order and returns to the depot. Prins et al. (2009) proposed an improvement called *Split with Shifts* which considers each subsequence as a circular list and determines the best insertion of the depot. Fig. 6 depicts this process on a subsequence (T_2, T_3, T_4) of three customers, assuming $c(T_4, T_2) = 20$. The costs if the trip starts with the first, second or third customer are respectively 120, 100 and 120. In *Split with Shifts*, the cost of arc (1, 4) which models the subsequence in the auxiliary graph will correspond to the best insertion (100) instead of 120 in the basic *Split*.

Recall that Algorithm 1 tests each subsequence $(T_i, T_{i+1}, \ldots, T_j)$ of the giant tour. In *Split with Shifts*, the best depot position must be recorded for each subsequence, to reorder the trip correctly when the CVRP solution is extracted using Algorithm 2. When $j = i$, the depot must be before T_i. Consider a subsequence $(T_i, T_{i+1}, \ldots, T_j)$ for which the best depot position is before customer T_k, $i \leqslant k \leqslant j$. For the next subsequence $(T_i, T_{i+1}, \ldots, T_{j+1})$, it is easy to see that the best depot position is still before T_k, before T_{j+1}, or after T_{j+1}. So, the best position for the depot can be updated in $O(1)$ instead of browsing $(T_i, T_{i+1}, \ldots, T_{j+1})$ and, finally, *Split with Shifts* has the same $O(n\beta)$ complexity as the basic *Split*. Experiments with two-phase constructive heuristics for the CVRP and the CARP show that versions with *Split with Shifts* outperform the ones based on the basic *Split* (Prins et al., 2009).

In the basic *Split*, the only filter to model or not a subsequence $(T_i, T_{i+1}, \ldots, T_j)$ as one arc in H is to check if vehicle capacity is respected, i.e., if $\sum_{k=i}^{j} q(T_k) \leqslant Q$. Other limitations like a maximum volume, distance traveled or working time are easily tackled. For instance, most benchmarks for the CVRP include some instances with a maximum distance L: any subsequence such that $cost(i, j) > L$ can be ignored. This test can be added to the *if* statement line 14 in Algorithm 1, like in Prins (2004).

As an optimal TSP or RPP tour does not necessarily lead to an optimal CVRP or CARP solution after splitting, the giant tour can be computed in practice by any heuristic. An easy way to improve the final result consists in splitting several giant tours, generated via a randomized heuristic. Prins et al. (2009) showed that this approach competes with classical heuristics for the CVRP and the CARP.

Prins et al. (2009) also describe iterative improvement procedures. An initial giant tour T is split, the resulting solution S undergoes a local search procedure to get a solution S' and the trips of S' are concatenated by $Split^{-1}$ to get a new giant tour T'. As the local search moves some customers, T' may differ significantly from T and a solution at least as good as S' is obtained if the same process is repeated on T'. In this method, the local search explores rather small neighborhoods, involving one or two trips, while *Split* behaves like a large neighborhood move, able to change simultaneously all trip limits. Tests on the CARP indicate that this approach is very effective but rather time-consuming.

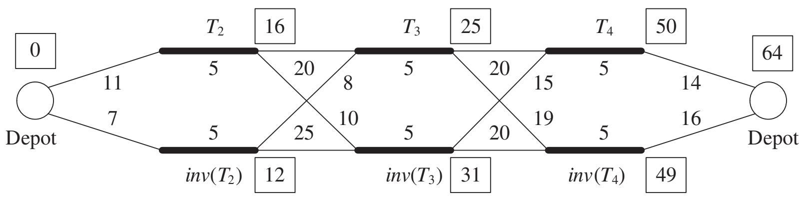


Fig. 5. *Split with Flips* for a subsequence of three tasks.

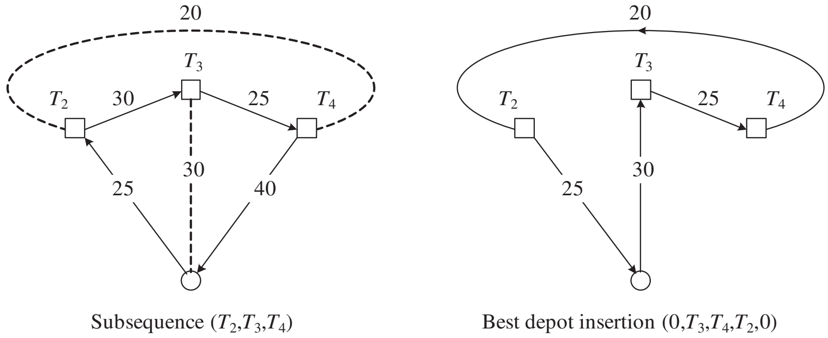


Fig. 6. *Split with Shifts* for a subsequence of three tasks.

3.3.2. Metaheuristics

A more advanced application of *Split* is to evaluate chromosomes encoded as an ordering of the tasks in population-based metaheuristics. This idea is based on the iterative approach presented in Section 2.1 (Fig. 3). Lacomme et al. (2001) were the first to develop a memetic algorithm (MA, a genetic algorithm hybridized with a local search) based on this principle for the CARP and designed in 2004 a more efficient version. Prins (2004) proposed for the CVRP a similar method which was the first genetic algorithm able to compete with the best methods available at that time, tabu search heuristics. In all these algorithms, new chromosomes are generated using a crossover operator. In the ant colony optimization algorithm proposed by Santos et al. (2010), *Split* is used again to evaluate giant tours, but these tours are built by the ants on the basis of pheromone deposits.

Prins (2009b) proposed for the CVRP another iterative approach called multi-start evolutionary local search (MS-ELS). This method constructs an initial CVRP solution S using a randomized heuristic and concatenates its trips to get a giant tour T, which gives a first pair (S, T). Then, each iteration produces a fixed number of children-solutions as follows: it applies a mutation operator to a copy T' of T, splits T' to get a CVRP solution S' and improves S' using a local search. The trips of the resulting best child are finally concatenated to give the pair (S, T) for the next iteration. MS-ELS outperforms the previous MA, both in terms of solution quality and running time. More recent MS-ELS methods implementing the alternation of Fig. 3 are cited in Sections 4 and 5 for other routing problems.

Split was used as a large neighborhood move by Tang et al. (2009), in a MA for the CARP. The local search of this MA browses classical moves like 2-OPT and ends with a new move called merge-split (MS). MS selects a subset of routes at random, merges the customers into an unordered list and applies a classical CARP heuristic (Path-Scanning) to get five solutions. The trips of each solution are concatenated to get a giant tour and the basic *Split* is applied. The best resulting solution is returned. MS is quite effective to generate new solutions significantly different from the current solution.

The basic *Split* has been used in bi-objective genetic algorithms with chromosomes encoded as giant tours. Note that in all references cited below the *Split* procedure acts upon the total distance only: the other objective is handled by other components of the metaheuristic.

Lacomme et al. (2006) minimize the total length and the length of the longest route in the CARP, using the non-dominated sorting genetic algorithm NSGA-II (Deb, 2001). Reiter and Gutjahr (2012) consider the same objectives for the CVRP. An adaptive ε-constraint method is used to derive all Pareto-optimal solutions from a set of non-dominated solutions pre-computed by NSGA-II. A branch-and-cut algorithm is called to ensure that the solution of each single-objective sub-problem is optimal.

Jozefowiez et al. (2007) consider another balance criterion for the CVRP, the difference between maximum and minimum route lengths. They develop a new multi-objective heuristic, called Target Aiming Pareto Search, in which each solution of the Pareto front P computed via NSGA-II undergoes a local search restricted to a vertical stripe between this solution and the previous one on P. This strategy ensures that the entire area that dominates P is explored, while avoiding that two local searches examine the same area. In 2009, the same authors developed for the same problem a cooperative MA based on the island model and implemented on eight processors.

Sörensen (2006) designed a MA to compute good but not too similar CVRP solutions. The chromosomes are encoded as giant tours and decoded by *Split*. The dissimilarity is evaluated by a distance measure in solution space. If $D_P(C)$ is the smallest distance of a child C to any solution in population P, child C is accepted if $D_p(C) \geqslant \Delta$, where Δ is a diversity parameter. This acceptance rule leads to a final population which provides the decision maker with good and diverse solutions.

In the *stochastic CARP*, demands are random variables like the amount of garbage of a street in waste collection. When a trip planned with average demands is executed, two cases are possible, depending on the real demands met on the field: if the total demand does not exceed vehicle capacity, the planned distance is respected, otherwise the trip must be interrupted to go and unload at the depot, which incurs an additional distance. Fleury et al. (2005) run the CARP MA (Lacomme et al., 2004) with average demands and with or without a spare capacity in each vehicle and then evaluate the robustness of solutions through simulation.

Using mild assumptions, Fleury et al. (2004) derive analytical expressions for the mathematical expectation of the cost of each route. The basic *Split* is still used to convert each giant tour into a planned solution, using deterministic costs, but the associated expected cost on the field is immediately deduced using the analytical formulas. Fleury et al. (2008) extended this approach by adding a second objective to balance the routes, the duration of the longest trip.

3.4. Other applications of basic Split

Several vehicle routing problems closely related to the CVRP or the CARP can be handled by changing the composition of giant tours, without affecting the auxiliary graph construction and the shortest path algorithm. For example, a giant tour may combine required nodes and edges in waste collection problems, or contain several occurrences of the same customer in split delivery problems.

The *mixed CARP* (MCARP), inspired by municipal waste collection, combines required edges and required arcs with given demands. A one-way street is modeled by one arc. A two-way street with two sides collected separately gives two opposite arcs. A required edge models a two-way street whose both sides can be collected in parallel and in any direction. Belenguer et al. (2006) adapted to the MCARP the CARP MA designed by Lacomme et al. (2004). Compared with the encoding of Section 3.1, a required arc u is such that $inv_u = 0$ and giant tours contain all required arcs and one of the two arcs that represent required edges. Lacomme et al. (2004) described how to adapt their CARP MA to what they call the *Extended CARP* (ECARP). This ECARP corresponds to a MCARP with forbidden turns, turn penalties and parallel links, useful to model service roads for instance.

The *node, edge and arc routing problem* (NEARP) or *mixed and capacitated general routing problem* (MCGRP) can be viewed as a MCARP to which required nodes are added. In a waste management context, these nodes correspond to punctual accumulations of waste like hospitals or supermarkets. A memetic algorithm for the NEARP was designed by Prins and Bouchenoua (2004). Giant tours include all required arcs, all required nodes, and one arc (traversal direction) for each required edge.

Apart from the giant tour composition, the splitting procedure remains identical in these memetic algorithms for the MCARP and the NEARP but the local search procedure is more involved. For instance, a 2-OPT move consisting in inverting a subsequence of tasks is always possible in the CVRP and the CARP, but it becomes infeasible for the MCARP if the subsequence contains required arcs.

Mourão et al. (2009) investigated the *sectoring-arc routing problem* (SARP) raised by municipal refuse collection. The goal is to partition the mixed graph of the streets in a given number of sectors (each sector being assigned to a vehicle and its crew), and to solve the MCARP in each sector to minimize the total duration of trips. Three heuristics are described. One of them builds balanced sectors and solves the mixed CARP for each sector by constructing a mixed rural postman tour which is then converted into a set of feasible trips via the *Split* procedure.

Liu et al. (2010b) investigate *the task selection and routing problem*. A carrier with a private fleet considers two sets of transportation requests defined by an origin i, a destination j and a distance. The first set contains tasks given by shippers, which can be executed by private vehicles or sub-contracted with a penalty g_{ij}. The second set contains tasks subcontracted by other carriers, which can be refused or processed with a compensative payment e_{ij}. The goal is to select the tasks and to calculate a set of routes to minimize a total cost including the distance travelled, the fixed costs of vehicles used and the penalties for sub-contracting demands, decreased by the compensative payments for accepting demands from other carriers. Each route performs one request at a time and is limited by a maximum distance. Liu et al. designed a MA where each chromosome is an ordering T of the n requests.

The goal of the splitting procedure is to determine a position π such that demands T_1 to T_π are served by private vehicles while $T_{\pi+1}$ to T_n are left to external carriers. The basic splitting procedure is first applied with ordinary costs (without penalties and compensations), which gives the labels V_j of Algorithm 1. For a given π, the solution cost is then $V_\pi - \sum_{k=1}^{\pi} e(T_k) + \sum_{k=\pi+1}^{n} g(T_k)$. The best solution for a given ordering is finally obtained by testing each value of π.

Vehicle routing problems with split deliveries have raised a growing interest since a paper by Archetti et al. (2006), showing that the optimal cost for a CVRP instance can be halved in the best case if split deliveries are allowed. Labadi et al. (2008c) introduce a memetic algorithm with a distance measure in solution space for the *split-delivery CARP* (SDCARP). The crossover operator generates giant tours in which each edge may occur several times, with the amount served for each visit. For a trip with multiple visits to the same edge, it is easy to see that the cost does not increase if all but one copies of the edge are deleted, if the triangle inequality holds. As finding the best deletions can be time-consuming if the trip contains many interleaved split edges, the splitting procedure keeps only, in each subsequence, the first visit to each split edge to assign a cost to the trip. Belenguer et al. (2010) reuses the same encoding and splitting method, but in a more efficient multi-start evolutionary local search. Boudia et al. (2007) adapted the MA of Labadi et al. (2008c) to the *split-delivery VRP* (SDVRP).

Multi-depot problems can be solved by specific splitting methods able to select the best depot for each trip, see Sections 4.2 and 5. However, a few authors handled the *location-routing problem* (LRP) using one giant tour per depot. Nagy and Salhi (1996) developed a nested heuristic with route length estimation, in which a local search improves progressively a subset of open depots. To avoid solving completely a multi-depot VRP to evaluate each move, they assign customers to depots and build one giant tour for each depot. Instead of splitting the tour, the cost of the routes is derived from the tour length using a regression formula. Prins et al. (2006) describe a memetic algorithm for the location-routing problem, where each chromosome has two parts: one indicates which depots are open and to which depot each customer is assigned, the other contains one customer ordering for each open depot. To define the routes and get a complete LRP solution, *Split* is applied to each ordering. Prodhon and Prins (2008) generalize this MA to the *periodic LRP*, using the same encoding in each period.

4. Simple extensions impacting arc feasibility and/or arc costs

Up to now, the routes have been deduced from any task ordering (even mixing required nodes, arcs and edges) by the basic version of *Split*. Several CVRP and CARP extensions have been treated using simple modifications, like giant tours with multiple visits to customers in the split-delivery case or the application of the basic *Split* to sub-problems restricted to one depot or sector. This section covers cases for which the criteria for adding an arc in the auxiliary graph H and/or computing its cost are affected, but without changing the shortest path algorithm once the auxiliary graph is ready.

4.1. Extra constraints on tasks

In practice, many tasks have a *time window* $[b_i, e_i]$ to begin service, giving a *vehicle routing problem with time windows* (VRPTW). Late arrivals are forbidden but a vehicle may arrive before b_i and wait. Most published heuristics try to minimize the fleet size f_1 and then the total distance f_2: time windows are only used to check route feasibility. Labadi et al. (2008b) designed a MA with chromosomes encoded as giant tours. In *Split*, subsequences of customers violating time windows are detected using the dynamic programming techniques introduced by Kindervater and Savelsbergh (1997) and discarded. Using an objective function $Mf_1 + Nf_2$, this MA can give priority either to the number of vehicles used (M large and $N = 1$) or to the total distance ($M = 1$ and N large). Chang and Chen (2007) designed independently a similar memetic algorithm to minimize the total distance traveled.

Ursani et al. (2011) considered the VRPTW with distance minimization as the sole objective and study a localized genetic algorithm, based on overlapping sub-problems. Starting from one random ordering of customers, evaluated by *Split*, the following cycle is applied: (a) the overlapping sub-problems, corresponding to all possible pairs of routes, are optimized individually using a fast GA; (b) a non-bipartite matching problem is solved to select a subset of route-disjoint pairs maximizing the total saving; and (c) the trips of the resulting VRPTW are concatenated to yield a new giant tour which is perturbed using random exchanges of customers.

Labadi et al. (2008a) applied a GRASP with path relinking to the *CARP with time windows* (CARPTW). In each GRASP iteration, a path is generated from the incumbent solution S to one target solution S' randomly selected in a pool of elite solutions. In fact, the giant tour obtained by $Split^{-1}(S)$ is progressively converted into the giant tour corresponding to S'. The intermediate tours are decoded by *Split* and improved using the local search of the GRASP. Reghioui et al. (2007) improved this algorithm using better greedy randomized heuristics, a path relinking between S and the most distant solution in the pool, and *Split with Shifts* (see Section 3.2) to evaluate giant tours.

Pairing constraints are frequent in pickup and delivery problems (PDP) met for instance in on-demand transportation. Each request comprises a pickup node and a delivery node. Each time a vehicle route stops at a node, passengers may be taken or dropped but a route transporting a passenger must contain both his/her pickup and delivery nodes. In *Split*, subsequences violating these pairing constraints must be ignored. Velasco et al. (2009) analyze a memetic algorithm based on this principle for a PDP raised by the transportation of personnel to oil fields by helicopter. The same authors designed in 2012 a bi-objective version with the total duration of the routes and a service level measure (a weighted sum of arrival times).

4.2. Impact on route costs

4.2.1. Unlimited heterogeneous fleet

The *vehicle fleet mix problem* (VFMP) or *fleet size and mix problem* considers a virtually unlimited fleet of p vehicle types. Each vehicle type k has a capacity Q_k, a purchase cost F_k and a cost per distance unit Z_k: hence, the cost of a trip of length λ is $F_k + \lambda Z_k$. The goal is to determine the fleet of vehicles and their trips to minimize the total cost. *Split* can tackle the VFMP by allocating to each subsequence of customers the cheapest vehicle type with enough capacity.

Golden et al. (1984) tested a two-phase VFMP heuristic in which a number of 2-optimal TSP tours are partitioned in this way into feasible routes. Ulusoy (1985) tackled a fleet size and mix CARP, solving first a rural postman problem (RPP) to get a giant tour. To initialize a variable neighborhood search for the VFMP, Imran et al. (2009) compute one giant tour using a sweep algorithm and improve it by 2-opt moves before splitting. Salhi and Sari (1997) introduce a three-phase heuristic for a multi-depot VFMP. The first phase assigns customers to depots, builds one giant tour for each depot used and its customers, and splits each giant tour. The second phase is a local search based on simple moves while the third one is devoted to more involved moves that may change the depot assigned to each route.

The splitting procedure has been applied in memetic algorithms for the VFMP to evaluate chromosomes encoded as giant tours, see Liu et al. (2009) and Prins (2009a). The latter reference describes two versions tackling both the VFMP and the heterogeneous fixed fleet VRP described in Section 5. The second version uses a distance measure in solution space to control diversity.

In *open routing problems*, the vehicles stay at the last customer at the end of their trip instead of going back to the depot. This possibility is offered by some truck rental contracts. The trip cost in *Split* simply becomes

$cost(i,j) = c(0, T_i) + \sum_{k=i}^{j-1}(s(T_k) + c(T_k, T_{k+1}))$. Liu et al. (2010a) implement such a splitting procedure in a memetic algorithm for the open capacitated arc routing problem.

In *multi-depot problems*, the departure and end nodes for each trip must be selected among a set D of depots. Uncapacitated depots are easily handled by selecting for each subsequence $(T_i, T_{i+1}, \ldots, T_j)$ the best depot $dep(i,j) = \arg\min\{c(k, T_i) + c(T_j, k) | k \in D\}$ and computing the trip cost as $cost(i,j) = c(dep(i,j), T_i) + \sum_{k=i}^{j-1}(s(T_k) + c(T_k, T_{k+1})) + s(T_j) + c(T_j, dep(i,j))$. The best depot for all pairs (i,j) of customers can be pre-computed in $O(n^2)$. Kansou and Yassine (2010) apply this technique in an ant colony algorithm for the *multi-depot CARP* (MDCARP): each ant builds a rural postman tour, then *Split* determines the trips with their best depot. Xing et al. (2010) focus on the ECARP defined in Section 3.3 but add multiple depots. Their solution method is a memetic algorithm inspired by the CARP MA of Lacomme et al. (2004) and the MCARP MA of Belenguer et al. (2006), but involving edge frequencies to choose break-points in the crossover and mutation operators.

Recently, Vidal et al. (2014) extended *Split with Shifts*, described in Section 3.2, in an iterated local search and a hybrid genetic algorithm for the *multi-depot VRP* (MDVRP). For each subsequence of the giant tour, the splitting procedure determines both the best depot and its optimal insertion. It is even possible to add an unlimited heterogeneous fleet of vehicles, giving a *multi-depot vehicle fleet mix problem* (MDVFMP): in that case, *Split* selects additionally the cheapest vehicle type compatible with the trip load. All these extensions are implemented with the same $O(n\beta)$ complexity as the basic *Split*. Moreover, the hybrid GA outperforms state of the art metaheuristics on the two problems.

The *2-echelon location-routing problem* (LRP-2E) is another multi-depot problem where two levels of routes must be built: from a main depot to potential satellite depots, with opening costs and limited capacities, and from the selected satellites to a set of customers. Nguyen et al. (2012a) designed for this problem a GRASP reinforced by a learning process and path relinking. One of the randomized heuristics called in this GRASP builds a set of subtours covering all customers, each subtour being compatible with the capacity of a first-level vehicle. Satellite capacities are then relaxed and the splitting process explained before for the MDCARP is used to partition each subtour into second-level trips while selecting the best satellite. Possible satellite capacity violations are repaired by transferring trips. Finally, a nearest neighbor heuristic constructs first-level trips to serve the open satellites and obtain a complete LRP-2E solution.

The same authors (2012b) developed a multi-start iterated local search (MS-ILS) implementing the alternation of Fig. 3 between giant tours and complete solutions. The splitting procedure is identical to the one used in the GRASP, except that the giant tour covers all customers and a subset of satellites to open is pre-selected, with enough capacity to serve the customers.

The memetic algorithm proposed by Vidal et al. (2012) for the periodic and multi-depot VRP with limited fleet (see Section 5.1) was extended to time windows in Vidal et al. (2013a). This MA accepts infeasible but penalized solutions with respect to route constraints: capacity, duration and time windows. A sophisticated local search evaluates moves in amortized constant time while problem decompositions are developed to cope efficiently with large instances. The algorithm outperforms all published methods for the VRPTW and the PVRP and MDVRP with time windows.

4.3. Constraints on vehicle loading

Regulations on hazardous materials like chemicals forbid the transportation of some products in a same vehicle. Hamdi-Dhaoui (2011) studies a hybrid GA for a VRP where each customer requires a set of products and a list of incompatibilities (product pairs) is given. In the splitting procedure, subsequences containing incompatible products do not correspond to valid trips and can be discarded.

Even when products are compatible, they must be loaded in the vehicles. In the *two-dimensional loading VRP* (2L-VRP), each customer orders rectangular items. A feasible packing in the rectangular bay of the vehicle must be determined for the set of items delivered in each route. The problem is extremely hard because it combines two NP-hard problems: the CVRP and a two-dimensional bin packing problem. *Split* must check that a feasible packing exists for each subsequence of customers.

Duhamel et al. (2011a) develop a multi-start evolutionary local search for the 2L-VRP, with the same alternation between giant tours and complete solutions as in Prins (2009b). The NP-complete packing problem to decide if the items assigned to a vehicle can be loaded is relaxed as a resource-constrained scheduling problem with one resource (vehicle area) and solved by a fast heuristic. The subsequence is accepted if this heuristic finds a feasible solution to the relaxed problem. A real packing is determined at the end of the MS-ELS for the best solutions recorded during the search. This technique is very efficient: the tests show that this packing can be found in more than 90% of cases if the relaxed problem has a solution. The approach has been extended to three-dimensional loading constraints by Lacomme et al. (2013). Khebbache-Hadji et al. (2013) study a giant tour memetic algorithm for the 2L-VRP with time windows, in which feasibility is checked using a packing heuristic.

El Fallahi et al. (2008) consider a *multi-compartment vehicle routing problem* (MC-VRP) where each vehicle can carry ρ kinds of products (e.g., frozen goods, refrigerated products and non-perishable foodstuffs), using ρ dedicated compartments. Knowing the demands of customers for each product, the goal is to build a least-cost set of trips in which each product ordered by a customer is delivered by one vehicle. However, a customer may receive its different products from several vehicles. In fact, this MC-VRP can be viewed as a restricted split-delivery VRP, in which the products for a customer may be delivered separately, but not the amount specified for one product. The authors describe a hybrid GA with path relinking: like in the SDCARP papers (Labadi et al., 2008c; Belenguer et al., 2010) cited in Section 3.3, a customer may occur several times in giant tours (but here no more than ρ times) and *Split* regroups all deliveries to a customer on the first visit when computing the cost of a trip.

Mendoza et al. (2010) also consider a CVRP with multiple compartments, but this time the ρ products ordered by a customer must be brought by the same vehicle and the demand for each product is stochastic. Vehicle routing problems with stochastic demands are cited in Section 3.1, but they have no compartments and the giant tours are partitioned using deterministic trip costs by the basic *Split*. Mendoza et al. compute the probability of trip interruption caused by one product, the probability of interruption if all products are considered simultaneously and the resulting mathematical expectation of trip cost. These costs are used as arc weights in the auxiliary graph.

5. Split taking into account limitations on shared resources

Considerations that played a role in the modifications of the splitting procedure studied until now have no influence on the computation of the shortest path to determine the routes, once the auxiliary graph is ready. In this section, Bellman's algorithm for directed acyclic graphs can no longer be used in *Split*, due to special objective functions or limited resources to be shared by the different trips. More involved shortest paths algorithms, sometimes not fully polynomial, must be used.

5.1. Homogeneous fixed fleet

Bellman's algorithm for directed acyclic graphs, used in Algorithm 1, has no control on the number of vehicles (number of arcs on the shortest path). For a limited fleet of b vehicles, the general form of Bellman's algorithm can be used. At the beginning, it initializes labels $V_0^0 = 0$ and $V_j^0 = \infty$ for each node $j \neq 0$. At each iteration $k > 0$, it computes $V_j^k = \min\{V_i^{k-1} + cost(i,j) | i \textit{ predecessor of } j\}$. It is easy to show recursively that labels at iteration k define the costs of shortest paths with at most k arcs. Like in Section 3.2, let β be the average number of customers per feasible subsequence. Each iteration costs $O(n\beta)$ and the best splitting with at most b vehicles can be obtained in $O(bn\beta)$, by stopping the algorithm at the end of iteration b.

This method has been used in population-based methods with giant tours for *periodic problems*. These problems are defined over a multi-period horizon and can be solved by splitting the list of customers visited in each period. The *periodic CARP* (PCARP) is a good example raised by household refuse collection. Each edge (street segment) e has a frequency f_e and a daily waste production q_e. To satisfy implicitly spacing constraints between services, a set of day combinations or patterns is associated with each street, for instance {(Monday, Thursday), (Tuesday, Friday)} for $f_e = 2$. For each pattern and each of its days, a demand can be deduced from the q_e. Indeed, the amount of waste found in a street by a vehicle is the total amount of waste produced since the previous day of the combination. The goal is to select one day combination for each visit and to solve a CARP in each period, to minimize the total cost of the routes over the planning horizon. The fleet is always limited, otherwise optimal solutions concentrate the activity in a few periods, leading to excessive workload variations.

Lacomme et al. (2005) design for the PCARP a memetic algorithm in which each chromosome, divided into ordered sub-lists (one per day), contains f_e copies compatible with a day combination for each edge e, e.g., the index of one edge e with frequency two and day combination (Monday, Thursday) appears in sub-lists 1 and 4. New chromosomes satisfying also frequencies and day combinations are generated using a periodic LOX crossover (PLOX) which extends the classical linear order crossover (LOX) for the TSP. The hierarchical objective function, to be minimized, gives priority to the required fleet size and then to the total length of the routes. *Split* is first applied to each sub-list, with a unit cost on each arc of the auxiliary graph H. The costs of resulting shortest paths correspond to the minimum number of vehicles required in each day, and the fleet size b is the maximum of these values. *Split* is finally executed for each day, with the real trip cost on each arc of H and an additional constraint: the shortest path may use up to b arcs, i.e., up to b vehicles.

Chu et al. (2006) develop a scatter search based on the same splitting procedure. Although *Split* is optimal for the sequence defined by a chromosome, the crossover may generate children with an excessive fleet size. Mei et al. (2011) improve on average the results of the two previous metaheuristics by adding a procedure which reduces fleet size.

Vidal et al. (2012) elaborate a powerful memetic algorithm for the PVRP, based on the chromosome encoding used in Lacomme et al. (2005) for the PCARP. However, the fleet size b is imposed in the data, a different crossover called PIX (periodic insertion crossover) is used, the algorithm accepts solutions that violate vehicle capacity, and chromosomes are replaced according to a rule which takes both solution quality and diversity into account. The general form of Bellman's algorithm is used to split the sub-chromosome associated with each period into at most b routes. To avoid too many arcs in the auxiliary graph and limit capacity violations, the load of each subsequence modeled as one arc in the auxiliary graph is limited to $2Q$. The proposed MA can solve the PVRP, the multi-depot VRP (each depot being assimilated to a period) and the PVRP with multiple depots.

Ngueveu et al. (2010) address the *cumulative VRP* with a limited fleet, where the objective consists in minimizing the sum of arrival times at serviced nodes. This problem is raised by disaster logistics, in which relief goods must be quickly dispatched. The cost of an arc in the auxiliary graph is the sum of arrival times at the nodes of the corresponding subsequence in the giant tour. Here again, the general form of Bellman's algorithm is executed to extract the routes.

Lacomme et al. (2004) consider a version of the CARP raised by urban refuse collection, where the goal is to minimize the length of the longest route, subject to a limited fleet. This can be achieved using a min–max version of Bellman's algorithm, in which the label of node j at iteration k is recursively computed as $V_j^k = \min\{\max(V_i^{k-1}, cost(i,j)) : i \textit{ predecessor of } j\}$.

5.2. Multiple resources

Many routing problems encompass in practice several categories of resources like vehicles and depots. Each resource in a category can be characterized by a set of attributes. For a depot k, the attributes can be for instance its capacity Ω_k and opening cost O_k. For a vehicle type k, one can define the number of vehicles available a_k, a capacity Q_k, a fixed cost F_k and a cost per distance unit Z_k.

The consumption of resources by a route of length λ and total demand *load* depends on the category. For vehicles, we can choose one in any type k such that $load \leqslant Q_k$, with a route cost $F_k + \lambda Z_k$. Regarding depots, we can take any depot k with enough residual capacity and the route will consume in this depot *load* units of storage for the goods to be shipped. However, the opening cost O_k will be charged only once, even if several routes are based at the same depot. To simplify the exposé, we consider in the sequel the *heterogeneous fixed fleet VRP* (HVRP) with p vehicle types having the attributes defined above. The discussion can be easily transposed to depots or other shared resources.

In the basic *Split*, the label V_j on a node j of the auxiliary graph H is the minimum cost of the paths connecting the dummy node 0 to node j. For the HFVRP, we have to select one compatible vehicle for each arc in H, in such a way that the resource consumptions $r_1, r_2, \ldots, r_p$ of each path (number of vehicles consumed for each type) do not exceed $a_1, a_2, \ldots, a_p$, respectively. The search for a least-cost path in these conditions becomes a *resource-constrained shortest path problem* (RCSPP). Such problems are in general NP-hard, except in some particular cases like $p = 1$ in Section 5.1.

Fortunately, the RCSPP for the HVRP can be solved quickly enough in practice, by adapting a multi-label extension of Bellman's algorithm (Desrochers, 1988). The label for a path becomes a vector $V = (\phi | r_1, r_2, \ldots, r_p)$ with the path cost ϕ and resource consumptions r_k. Indeed, we have to remember these consumptions to know if a partial path can be extended. Theoretically, we should store at node j the labels of all incoming paths because, contrary to the basic *Split*, it is not certain that the cheapest path can be extended to reach the final node. In practice, many labels can be discarded using a *dominance rule*. Let $V.f$ denote field f of label V: label V *dominates* (weakly) label W if $V \cdot \phi \leqslant W \cdot \phi$ and $\forall k : V \cdot r_k \leqslant W \cdot r_k$. If V does not dominate W and W does not dominate V, the two labels are *incomparable*. A set of labels is said *non-dominated* if its labels are pairwise incomparable. A stronger domination rule where at least one of the inequalities must be strict could be used, like in multi-objective optimization, but a weak domination is enough here: if two labels corresponding to distinct paths have the same cost and resource consumptions, we can keep only one of them.

For example, consider one HVRP instance with $p = 3$ vehicle types, $a = (4, 3, 10)$, and one label $V = (1254|2, 1, 3)$. This means that the path associated with this label has a cost 1254 and consumes 2, 1 and 3 vehicles of each respective type. V dominates $(1254|2, 2, 3)$ which has the same cost but requires more vehicles of type 2. V dominates also $(1259|2, 1, 3)$: it requires the same vehicles while being cheaper, and $(1259|3, 1, 4)$ because it is both cheaper and less vehicle-consuming. Finally, V and $(1259|2, 0, 1)$ are not comparable: V is cheaper but needs more vehicles for types 2 and 3.

The splitting problem for one HVRP giant tour T is sketched in Algorithm 3. Like in Algorithm 1, all feasible subsequences $(T_i, T_{i+1}, \ldots, T_j)$ of the giant tour are enumerated (by the *for* loop line 3 and the *repeat* loop line 6) and the auxiliary graph H is not built explicitly. Only non-dominated labels are stored on each node and $\Lambda(i)$ denotes the set of labels for node i. Service times are here ignored because the cost of each route is deduced from its length and vehicle type, using the formulas presented before.

At the beginning, only node 0 has a label, corresponding to an empty path. The incumbent subsequence $(T_i, T_{i+1}, \ldots, T_j)$ has a total demand *load* (line 7) and a length λ (lines 8–12). Like in the basic *Split*, recall that this subsequence is modeled by one arc $(i - 1, j)$ in the auxiliary graph. This is why we try to propagate each label W of node $i - 1$ (*for* loop lines 14–24). For this, we test each vehicle type k (*for* loop lines 15–23). If the type has enough capacity and is still available, the cost of the route is deduced from its length and a new label W is prepared for node j (line 18). If W is not dominated (line 19), all labels it dominates are erased (lines 20–21) and the label is saved (line 21). The basic *Split* stops scanning the routes which begin at customer T_i when *load* exceeds the unique vehicle capacity Q. Here, a flag *stop* indicates if new labels have been generated for $(T_i, T_{i+1}, \ldots, T_j)$: if this not the case, it is useless to increment j and the algorithm may proceed with the next i.

In general, several non-dominated labels are obtained at node n and the returned solution corresponds to the least-cost one. Some giant tours may be infeasible, even if total demand does not exceed fleet capacity. For instance, consider $p = 2$ vehicle types with availabilities $a = (1, 1)$ and capacities $Q = (10, 5)$, and $n = 3$ customers with demands 7, 3, 5: $T = (1, 2, 3)$ is feasible but not $T' = (1, 3, 2)$. This can be detected line 14 if $\Lambda(i - 1) = \emptyset$. In metaheuristics, infeasible tours are simply discarded.

Algorithm 3. Implementation of *Split* for the HVRP

```
Λ(0) ← {(0|0, ..., 0)}
for i ← 1 to n do Λ(i) = ∅ endfor
for i ← 1 to n do
  j ← i
  load ← 0
  repeat
    load ← load + q(T_j)
    if i = j then
      λ ← c(0, T_i) + c(T_i, 0)
    else
      λ ← λ − c(T_{j−1}, 0) + c(T_{j−1}, T_j) + c(T_j, 0)
    endif
    stop ← false
    for each label V = (φ|r_1, r_2, ..., r_p) in Λ(i − 1) do
      for each vehicle type k such that (load ⩽ Q_k) and (r_k < a_k) do
        stop ← true
        cost ← F_k + λU_k
        W ← (φ + cost|r_1, ..., r_k + 1, ..., r_p)
        if no label in Λ(j) dominates W then
          delete in Λ(j) all labels dominated by W
          Λ(j) ← Λ(j) ∪ {W}
        endif
      endfor
    endfor
    j ← j + 1
  until (j > n) or (stop = false)
endfor
```

Fig. 7 gives one example of label propagation for the three first customers T_1, T_2, T_3 of a giant tour. The figure provides the locations of customers in the Euclidean plane and their demands in brackets. The fleet has $p = 2$ vehicle such that $a = (3, 1)$, $Q = (10, 9)$, $F = (2, 1)$ and $Z = (0.2, 0.6)$. Hence, the cost for a trip of length λ is $2 + 0.2\lambda$ for type 1 and $1 + 0.6\lambda$ for type 2. The

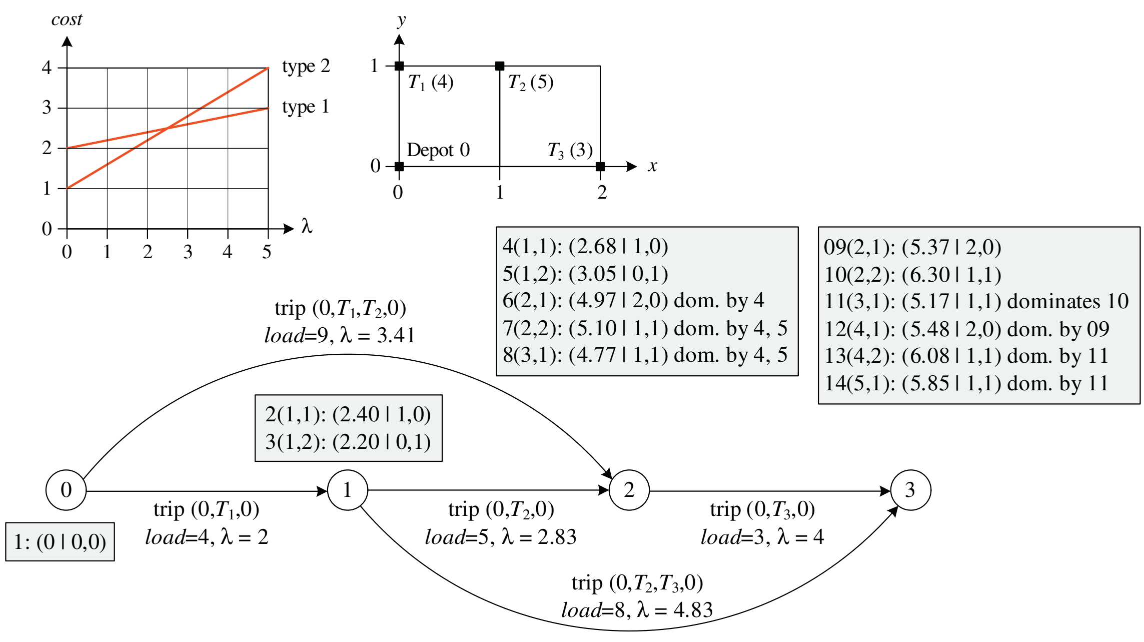


Fig. 7. Example of label propagation in *Split* for the HVRP.

list of generated labels is given for each node. In the header "$h(p,k)$" before each label, h is a counter indicating in which order the labels are generated, p is the index of the label used as predecessor and k the type of vehicle used. Thanks to dominance, only two labels are obtained at node 3: (5.17|1, 1) and (5.37|2, 0).

In addition to dominance, various knacks can be used to discard the new label W generated line 14. For example, let $Q_{res} = \sum_{k=1}^{p}(a_k - r_k)$ be the residual fleet capacity and $D_{res} = \sum_{k=j+1}^{n} q(T_k)$ the total demand not yet satisfied: W can be ignored if $D_{res} > Q_{res}$. Another way is to compute an upper bound UB, using a greedy heuristic, and a lower bound $LB(j)$ for the cost of a shortest path from node j to node n in H, relaxing resource constraints. The label can be discarded if $W \cdot \phi + LB(j) \geqslant UB$.

Suboptimal but faster versions of *Split* can be used in metaheuristics to accelerate the search on large instances. The simplest way is to limit the number of labels for each node (only the cheapest ones are kept) and/or the total number of labels generated. As Algorithm 3 examines the outgoing arcs for node $0, 1, \ldots, n$, many labels can be generated before getting a first label at node n. A depth-first-search (DFS) of the auxiliary graph can be used to reach node n more rapidly. This method called *DFS Split* requires a stack of labels which stores a partial path rooted at node 0. This stack is initialized with an empty path label, like in line 1 of Algorithm 3. When a path is extended, the corresponding label is pushed on the stack. A backtrack occurs when the final node is reached or when the incumbent path cannot be extended due to lack of resources. The search stops when the stack is empty.

Prins (2009a) implemented Algorithm 3 in a memetic algorithm for the HVRP, including the label elimination based on remaining demands and vehicles. The splitting procedure can be implemented in $O(mpn^p)$, where m is the number of subsequences such that $load \leqslant \max(Q_k | k = 1, 2, \ldots, p)$. This pseudo-polynomial complexity becomes fully polynomial when the number of vehicle types p is fixed, which is often true in practice since most companies use the same types during a few years. In practice, the algorithm is fast enough for $p \leqslant 5$.

Duhamel et al. (2010) adapted this algorithm to the LRP, where resources correspond to depots with limited capacity. Their version named *greedy split* is strengthened by a lower bound, a limited number of labels per node and a limit on the total number of labels. Called in a multi-start evolutionary local search, it improves several best-known solutions for classical LRP instances. The same authors (2011b) compared the *greedy split* with *DFS Split* when included in MS-ELS for the HVRP and the LRP. The results indicate that limiting the number of labels has only a marginal impact on solution quality and that the DFS version leads to better results on average. A further study on the HVRP confirmed the superiority of *DFS Split* in a hybrid evolutionary local search (Duhamel et al., 2012).

6. Split dedicated to special auxiliary graphs

This section surveys algorithms which are still based on the ordering-first split-second principle but whose auxiliary graph H differs from the one used in the splitting procedures presented up to now.

Ryan et al. (1993) study a CVRP heuristic where customers are sorted in ascending polar angle, taking the depot as origin, to give a circular list $T = (T_1, T_2, \ldots, T_n)$. The solution of a TSP for the customers of a subsequence of T is called a *1-petal*. The best partition into 1-petals requires a special auxiliary graph H. Its node-set is a ring whose the nodes correspond to T_1, $T_2, \ldots, T_n$, respectively. Any feasible subsequence $(T_i, T_{i+1}, \ldots, T_j)$ (circularly) is modeled in H as one arc $(i, (j+1) \bmod n)$: e.g., a trip visiting T_n only gives arc $(n, 1)$. The associated 1-petal and its cost are computed applying an Or-Opt local search to the trip $(0, T_i, T_{i+1}, \ldots, T_j, 0)$. The best partition with k as first customer is given by a shortest circuit containing k, computable as follows: (a) take a copy H' of H, (b) discard all arcs bypassing k, (c) split node k into one origin k' and one destination k'', k' initiating the outgoing arcs of k and k'' receiving its incoming arcs, and (d) compute a shortest path from k' to k'' in H'. This process can be repeated for $k = 1, 2, \ldots, n$ to get the overall best partition. In fact, if $(T_i, T_{i+1}, \ldots, T_j)$ denotes the smallest feasible subsequence (in terms of customers), the authors show that the optimal partition can be found by using only $T_i, T_{i+1}, \ldots, T_j, T_{j+1}$ as first customer k.

Renaud et al. (1996) extends the previous heuristic by adding 2-petals to the auxiliary graph. A 2-petal is a pair of routes built from the customers of a subsequence of T, with a total load greater than Q but not greater than $2Q$. It is initialized by two back-and-forth routes connecting the two most distant customers and the depot. The remaining customers are added using successive cheapest insertions, interleaved with a reoptimization of the two emerging routes via 4-Opt* (double bridge) moves.

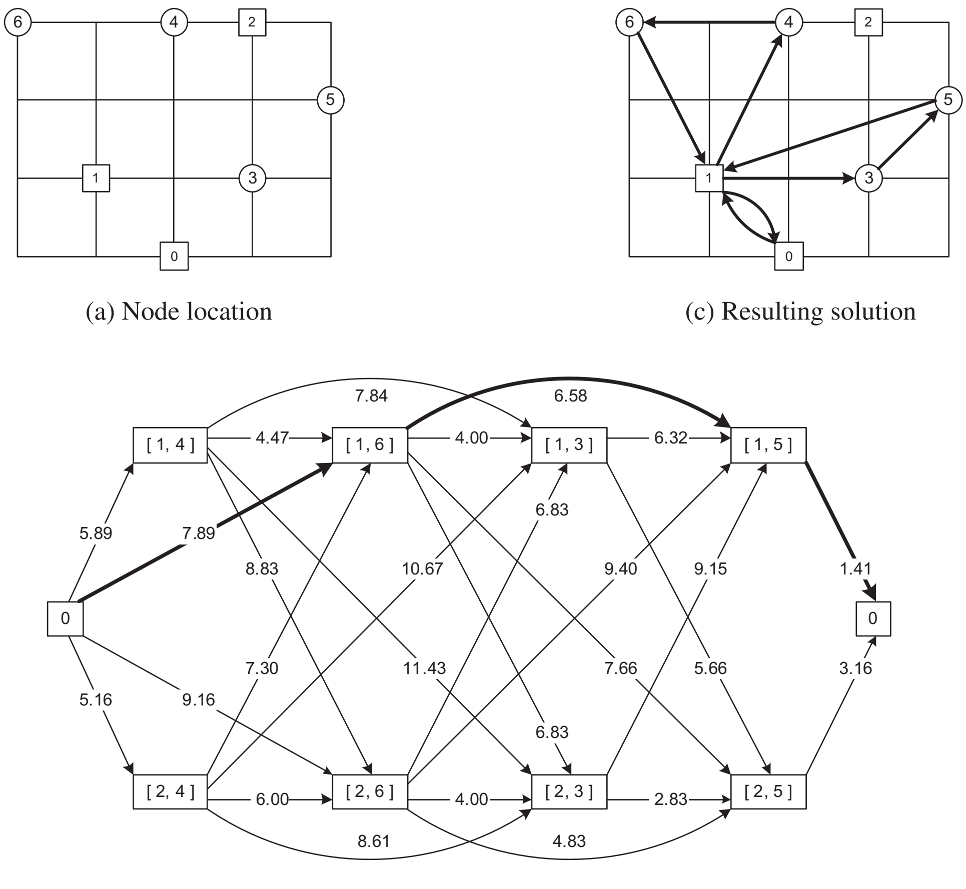
In the *generalized traveling salesman problem* (GTSP), the n customers are partitioned into p clusters and the goal is to determine a minimum-cost cycle, starting and ending at the depot, and visiting a single node in each cluster. Bontoux et al. (2010) devised a memetic algorithm in which each chromosome is an ordering $T = (T_1, T_2, \ldots, T_p)$ of the clusters. The evaluation is based on an acyclic auxiliary graph with $p + 2$ layers of nodes. Layers 1 and $p + 2$ contain only the depot while layers 1 to p contain the nodes of clusters $T_1, T_2, \ldots, T_p$, respectively. The arc-set contains all arcs (i, j) such that $i \in T_k$ and $j \in T_{k+1}$, for any cluster $k = 1, 2, \ldots, p - 1$. The optimal GTSP solution subject to this ordering corresponds to a shortest path between the two copies of the depot.

The *generalized VRP* (GVRP) is the capacitated version of the GTSP. A multi-start iterated local search has been developed recently for this problem by Afsar et al. (2014). It alternates between giant tours and complete solutions, according to the principle depicted in Fig. 3. A giant tour is in fact an ordering $T = (T_1, T_2, \ldots, T_p)$ of the p clusters. The decoding involves two nested splitting procedures. The basic *Split* is employed as usual to compute a shortest path in an auxiliary graph where each arc $(i-1, j)$ represents a route servicing clusters T_1 to T_j in this order. The method of Bontoux et al. (2010) is called inside to determine the best route for each subsequence. The cost of this route is used to weight the corresponding arc in the auxiliary graph of the basic *Split*.

Bouly et al. (2010) also use a special auxiliary graph for the *team orienteering problem* (TOP), in which a fleet of m identical vehicles with limited working time L must visit customers with known profits. Due to the time limit, it is impossible to serve all customers and the problem consists in selecting customers and building routes to maximize the total profit. Bouly et al. analyze a memetic algorithm in which chromosomes are giant tours containing all customers. Their interpretation of the splitting is to select subsequences (feasible routes) to maximize profit, but not necessarily consecutive: the customers left between two selected routes are not visited. They call *saturated route* a feasible route $(T_i, T_{i+1}, \ldots, T_j)$ such that $j = n$ (last route selected) or the duration of $(T_i, T_{i+1}, \ldots, T_j, T_{j+1})$ exceeds L. They show that there exists an optimal selection in which all routes are saturated.

The auxiliary graph contains four layers of nodes. Layer 1 is reduced to the depot (node 0), connected by one arc to each node in layer 2, which contains nodes 1 to n modeling customers T_1 to T_n. Layer 3 is made of nodes $n+1$ to $2n$ which are copies of nodes 1 to n. One arc links each of these nodes to a depot copy (node $2n+1$) in layer 4. One arc $(i, n+i)$ from layer 2 to layer 3 models the saturated route beginning at customer T_i, it is weighted by the associated profit. One arc $(n+i, j)$ from layer 3 to layer 2, with a null cost, indicates that the saturated tour beginning at T_j may start after the one starting at T_i, maybe some unvisited customers in between. The optimal splitting corresponds to a shortest path with at most $2m+1$ arcs in the resulting graph.

Villegas et al. (2010) develop a MS-ELS for the *single truck and trailer routing problem* (STTRP), where a truck with a trailer must serve n customers inaccessible with the trailer. The complete vehicle makes one primary trip linking parking locations, where the trailer is detached to do secondary trips on customers. A specific auxiliary graph is used to split a giant tour $T = (T_1, T_2, \ldots, T_n)$ for h parkings. It contains one row of n nodes per parking, between two depot copies. The node $[k, i]$ in row k, column i models a partial splitting whose last trip visits T_i before returning to parking k. An arc $([k, i], [l, j])$ extends the partial splitting $[k, i]$ as follows: (i) if $k \neq l$ then the truck takes its trailer to move from parking k to parking l and (ii) then



(a) Node location

(c) Resulting solution

(b) Auxiliary graph

Fig. 8. Auxiliary graph for the single truck and trailer routing problem (Villegas et al. 2010).

the truck drops the trailer at l, visits customers $T_{i+1}, T_{i+2}, \ldots, T_j$ and returns to l. An arc $(0, [k, i])$ means that the vehicle leaves the depot, goes to parking k, detaches the trailer, visits T_1 to T_i and returns to k. Finally, an arc $([k, i], 0)$ completes the partial splitting $[k, i]$ by a return to the depot. All arcs are weighted by the length of the associated move.

An optimal splitting corresponds to a least-cost path in this graph, computable in $O(n\beta h^2)$, where β is the average number of customers per secondary trip. Fig. 8 (Villegas et al., 2010) shows a small instance with trailer capacity 3, two parkings (nodes 1, 2) and four clients 3, 4, 5, 6 with respective demands 2, 1, 1, 1. The grid (a) is made of 1×1 squares and the distance is Euclidean. Part (b) shows the auxiliary graph and the shortest path in boldface. Part (c) gives the resulting solution.

Villegas et al. (2011) extends this approach to the *truck and trailer routing problem* (TTRP) with several vehicles. Roughly speaking, they nest two polynomial splitting procedures. The giant tour is partitioned via the basic *Split* to get the subset of customers processed by each vehicle. The splitting procedure for the single-vehicle case is called to assign a cost to each feasible subsequence.

7. Achievements, efficiency and limits of tour splitting approaches

Table 1 refers to 74 papers published over 30 years, concerning 38 different vehicle routing problems with ad hoc splitting procedures, and involving more than 70 sets of benchmark instances. Therefore, it is not possible in a limited space to provide tables of comparison with other approaches. It is also very difficult to quantify the contribution of a splitting procedure to the overall performance, when it constitutes one component among others in a metaheuristic. For instance, to compare a GA based on giant tours with one working on complete solutions, the crossover operator and the heuristics to fill the initial population must be also changed. However, we try in this section to recapitulate the main achievements of the ordering-first split-second approach, its efficiency, and its current limits.

7.1. Main achievements

The race for the best metaheuristics in vehicle routing has evolved very quickly in the last 30 years and many algorithms which outperformed their competitors at the time of their publication have been progressively out-distanced by new approaches. Nevertheless, split-based methods have been permanently well placed in this race and this survey provides many examples.

Concerning constructive heuristics, performance guarantees are known for splitting algorithms like SOTP for the capacitated general routing problem (Jansen, 1993) and A-ALG for the CARP (Wøhlk, 2008). Such results which are still rare in vehicle routing must be underlined. Moreover, A-ALG outperforms in practice well-known greedy heuristics for the CARP. Prins et al. (2009) showed that splitting a series of giant tours generated via a randomized heuristic outperforms many classical heuristics for the CARP and the CVRP.

Concerning metaheuristics, the memetic algorithms from Lacomme et al. (2001, 2004) for the CARP and from Prins (2004) for the CVRP were the first genetic methods to outperform state of the art tabu search algorithms. Other successes of split-based evolutionary algorithms include the mixed CARP (Belenguer et al., 2006), the periodic CARP (Chu et al., 2006), the heterogeneous fixed fleet VRP (Prins, 2009a), the multi-compartment VRP with stochastic demands (Mendoza et al., 2010), the location routing problem (Prodhon and Prins, 2008) and the cumulative VRP (Ngueveu et al., 2010).

In the last five years, the giant tour approach has led to better results when integrated in other types of metaheuristics, for instance in iterated local searches for the CVRP (Prins, 2009b), the location-routing problem (Duhamel et al., 2010) and the heterogeneous fixed fleet VRP (Duhamel et al., 2012), in a path relinking algorithm for the truck and trailer routing problem (Villegas et al., 2011), and in an ant colony optimization method for the CARP (Santos et al., 2010).

There is a strong competition today in vehicle routing between such algorithms and adaptive large neighborhood search (ALNS) but recent events indicate that genetic algorithms are now back to the pole position, with the hybrid GAs from Vidal et al. (2012, 2013a,b, 2014). It is true that the efficiency of these GAs not only relies on the giant tour approach, but also on sophisticated local search procedures and population management techniques, but anyway they constitute today the most efficient metaheuristics. For instance, Vidal et al. (2014) describe a hybrid GA with a unified splitting procedure, able to handle most simple extensions of Section 4 via an object-oriented software design. This GA is at least as good as the current best metaheuristics on 26 different problems.

7.2. Considerations about efficiency and limits

Two main reasons can be put forward to explain the efficiency on split-based metaheuristics. As we saw in Section 2.1, the approach allows working on a smaller search space, without losing information since an optimal solution (subject to the sequence) to the problem at hand can be deduced from each giant tour. Moreover, when used in conjunction with a local search, like in Fig. 3, the split procedure can be viewed as a large neighborhood operator which completes and strengthens the local search. The price to pay is the extra time spent in the splitting procedure, but we saw in Section 3.2 that the basic *Split* is quite fast, with an $O(n\beta)$ algorithmic complexity. Most additional constraints discussed in Section 4 can be handled without increasing this low complexity, e.g., time windows, multiple uncapacitated depots, unlimited heterogeneous fleets of vehicles.

Table 2
Distribution of papers in the four categories.

Comments	Basic Split Section 3	Simple extensions Section 4	Shared resources Section 5	Special graphs Section 6
First publication	1983	1984	2004	1993
First use in metaheuristics	2001	2004	2004	2010
Number of publications	34	27	10	7
Routing problems handled	19	13	6	5

An increase in complexity can be observed when restricted resources must be shared by the routes. This situation is not critical when the splitting procedure remains fully polynomial. When b identical vehicles are available, for instance, the complexity becomes $O(bn\beta)$.

The border between easy and hard cases is crossed in problems like the heterogeneous fixed fleet VRP (HVRP) and the location-routing problem with capacitated depots (CLRP), which raise NP-hard resource-constrained shortest path problems. Fortunately, multi-label extensions of Bellman's algorithms with a pseudo-polynomial complexity are often possible. It is fair to say that the running time can become excessive if many resources are involved, for instance more than 5 vehicle types in the HVRP (Prins, 2009a). However, even in such cases, it is possible to resort to heuristic and faster splitting procedures, with a small loss in solution quality, see Duhamel et al. (2010) for the CLRP and Duhamel et al. (2012) for the HVRP.

The difficulty cannot be predicted for splitting procedures based on unusual auxiliary graphs, like in Section 6. The problem must be studied in details to find the ad hoc graph and see if a polynomial implementation can be designed. This is at least the case for the generalized TSP (Bontoux et al., 2010), the team orienteering problem (Bouly et al., 2010), the truck and trailer routing problem (Villegas et al., 2010), and the generalized VRP (Afsar et al., 2014).

According to us, two research directions are promising. The first one consists in accelerating the split procedures in the hard cases like the HVRP, using for instance relaxed but accurate versions based on Lagrangean relaxation. The other direction is to better understand what happens when a metaheuristic navigates between giant tours and complete solutions. For instance, the current split-based heuristics alternate systematically between the two search spaces: it is probably possible to obtain better results by defining criteria to decide when *Split* or $Split^{-1}$ must be called.

8. Conclusion

This article reviews for the first time the main contributions on order-first split-second methods for vehicle routing problems. The eleven publications from 1980 to 2000 define the principles of these approaches and apply them to constructive heuristics or worst-case performance analysis. In most of these papers, the second phase consists in splitting a giant tour by computing a shortest path in an auxiliary graph, in which each arc models a possible trip. A turning point occurs in 2001 with the first publication integrating a splitting procedure in a metaheuristic (Lacomme et al., 2001). The key-feature consists in encoding solutions as giant tours and using a splitting procedure to extract the corresponding solution of the vehicle routing problem. After this work, more than 60 publications have reported very good results with the inclusion of splitting procedures in various metaheuristics.

Table 2 resumes the distribution of published articles according to this classification. The total number of papers is larger than in Table 1 because some papers propose several splitting procedures. The recent paper from Vidal et al. (2014) is not included because it would hide the trends, with its treatment of 26 problems by a single GA. It appears that 43% of publications use the basic version, even to deal with extensions of classical routing problems. Around 35% of articles tackle simple extensions required for instance by time windows or unlimited heterogeneous fleets. More involved adaptations with shared resources or special auxiliary graphs are still scarce, in spite of their numerous potential applications to rich vehicle problems.

We hope that the reader is now convinced that route-first cluster second heuristics compete with other heuristics for vehicle routing, contradicting what Laporte and Semet wrote in 2002 (see introduction).

References

Afsar, H.M., Prins, C., Santos, A.C., 2014. Exact and heuristic algorithms for solving the generalized vehicle routing problem with flexible fleet size. Int. Trans. Oper. Res. 21 (1), 153–175.
Altinkemer, K., Gavish, B., 1990. Heuristics for delivery problems with constant error guarantees. Transport. Sci. 24, 294–297.
Archetti, C., Savelsbergh, M.W.P., Speranza, M.G., 2006. Worst-case analysis for split delivery vehicle routing problems. Transport. Sci. 40, 226–234.
Badeau, P., Guertin, F., Gendreau, M., Potvin, J.-Y., Taillard, E., 1997. A parallel tabu search heuristic for the vehicle routing problem with time windows. Transport. Res. Part C: Emerg. Technol. 5 (2), 109–122.
Beasley, J.E., 1983. Route-first cluster-second methods for vehicle routing. Omega 11, 403–408.
Belenguer, J.M., Benavent, E., Lacomme, P., Prins, C., 2006. Lower and upper bounds for the mixed capacitated arc routing problem. Comput. Oper. Res. 33, 3363–3383.
Belenguer, J.M., Benavent, E., Labadi, N., Prins, C., Reghioui, M., 2010. Split-delivery capacitated arc routing problem: lower bound and metaheuristic. Transport. Sci. 44, 206–220.
Bertazzi, L., Bosco, A., Guerriero, F., Laganà, D., 2013. A stochastic inventory routing problem with stock-out. Transport. Res. Part C: Emerg. Technol. 27, 89–107.

Bettinelli, A., Ceselli, A., Righini, G., 2011. A branch-and-cut-and-price algorithm for the multi-depot heterogeneous vehicle routing problem with time windows. Transport. Res. Part C: Emerg. Technol. 19 (5), 723–740.
Beullens, P., Muyldermans, L., Cattrysse, D., Van Oudheusden, D., 2003. A guided local search heuristic for the capacitated arc routing problem. Eur. J. Oper. Res. 147, 629–643.
Bontoux, D., Artigues, C., Feillet, D., 2010. A memetic algorithm with large neighborhood operator for the generalized traveling salesman problem. Comput. Oper. Res. 37, 1844–1852.
Boudia, M., Prins, C., Reghioui, M., 2007. An effective memetic algorithm with population management for the split-delivery vehicle routing problem. In: Bartz-Beielstein, T. et al. (Eds.), Hybrid Metaheuristics, Lecture Notes in Computer Science, 4771. Springer, Berlin, pp. 16–30.
Bouly, H., Dang, D.C., Moukrim, A., 2010. A memetic algorithm for the team orienteering problem. 4OR – Quart. J. Oper. Res. 8, 49–70.
Chang, Y., Chen, L., 2007. Solve the vehicle routing problem with time windows via a genetic algorithm. Discrete Contin. Dyn. Syst. Suppl., 240–249.
Chu, F., Labadi, N., Prins, C., 2006. A scatter search for the periodic capacitated arc routing problem. Eur. J. Oper. Res. 169, 586–605.
Deb, K., 2001. Multi-objective Optimization using Evolutionary Algorithms. Wiley, Chichester.
Desrochers, M., 1988. An Algorithm for the Shortest Path Problem with Resource Constraints. Technical Report G-88-27, GERAD, Montréal.
Duhamel, C., Lacomme, P., Prins, C., Prodhon, C., 2010. A GRASP $\times$ ELS approach for the capacitated location-routing problem. Comput. Oper. Res. 37, 1912–1923.
Duhamel, C., Lacomme, P., Quilliot, A., Toussaint, H., 2011a. A multi-start evolutionary local search for the two-dimensional loading capacitated vehicle routing problem. Comput. Oper. Res. 38, 617–640.
Duhamel, C., Lacomme, P., Prodhon, C., 2011b. Efficient frameworks for greedy split and new depth first search split procedures for routing problems. Comput. Oper. Res. 38, 723–739.
Duhamel, C., Lacomme, P., Prodhon, C., 2012. A hybrid evolutionary local search with depth first search split procedures for the heterogeneous vehicle routing problem. Eng. Appl. Artif. Intell. 25 (2), 345–358.
El Fallahi, A., Prins, C., Wolfler Calvo, R., 2008. A memetic algorithm and a tabu search for the multi-compartment vehicle routing problem. Comput. Oper. Res. 35, 1725–1741.
Figliozzi, M.A., 2010. An iterative route construction and improvement algorithm for the vehicle routing problem with soft time windows. Transport. Res. Part C: Emerg. Technol. 18 (5), 668–679.
Fisher, M.L., Jaikumar, R., 1981. A generalized assignment heuristic for the vehicle routing problem. Networks 11, 109–124.
Fleury, G., Lacomme, P., Prins, C., 2004. Evolutionary algorithms for stochastic arc routing problems. In: Raidl, G.R. et al. (Eds.), Applications of Evolutionary Computing, Lecture Notes in Computer Science, vol. 3005. Springer, Berlin, pp. 501–512.
Fleury, G., Lacomme, P., Prins, C., Ramdane-Chérif, W., 2005. Improving robustness of solutions to arc routing problems. J. Oper. Res. Soc. 56, 526–538.
Fleury G., Lacomme, P., Prins, C., Sevaux, M., 2008. A Bi-objective Stochastic Approach for the Stochastic CARP. Report LIMOS RR-08-06, Blaise Pascal University, Clermont-Ferrand, France.
Gillett, B.E., Miller, L.R., 1974. A heuristic algorithm for the vehicle-dispatch problem. Oper. Res. 21, 340–349.
Golden, B., Assad, A., Levy, L., Gheysens, F., 1984. The fleet size and mix vehicle routing problem. Comput. Oper. Res. 11, 49–66.
Haimovich, M., Rinnooy Kan, A.H.G., 1985. Bounds and heuristics for capacitated routing problems. Math. Oper. Res. 10, 527–542.
Hamdi-Dhaoui, K., 2011. Optimisation conjointe du transport et du placement avec des contraintes de sécurité (Joint Optimization of Routing and Loading with Safety Constraints). PhD Thesis, Troyes University of Technology, France.
Huth, T., Mattfeld, D.C., 2009. Integration of vehicle routing and resource allocation in a dynamic logistics network. Transport. Res. Part C: Emerg. Technol. 17 (2), 149–162.
Imran, A., Salhi, S., Wassan, N.A., 2009. A variable neighborhood-based heuristic for the heterogeneous fleet vehicle routing problem. Eur. J. Oper. Res. 197 (2), 509–518.
Jansen, K., 1993. Bounds for the general capacitated routing problem. Networks 23, 165–173.
Jozefowiez, N., Semet, F., Talbi, E.G., 2007. Target aiming Pareto search and its application to the vehicle routing problem with route balancing. J. Heuristics 13, 455–469.
Jozefowiez, N., Semet, F., Talbi, E.G., 2009. An evolutionary algorithm for the vehicle routing problem with route balancing. Eur. J. Oper. Res. 195, 761–769.
Juan, A., Faulin, J., Grasman, S., Riera, D., Marull, J., Mendez, C., 2011. Using safety stocks and simulation to solve the vehicle routing problem with stochastic demands. Transport. Res. Part C: Emerg. Technol. 19 (5), 751–765.
Kansou, A., Yassine, A., 2010. New upper bounds for the multi-depot capacitated arc routing problem. Int. J. Metaheurist. 1, 81–95.
Khebbache-Hadji, S., Prins, C., Yalaoui, A., Reghioui, M., 2013. Heuristics and memetic algorithm for the two-dimensional loading capacitated vehicle routing problem with time windows. Central Eur. J. Oper. Res. 21, 307–336.
Kindervater, G.A.P., Savelsbergh, M.W.P., 1997. Vehicle routing: handling edge exchanges. In: Aarts, E., Lenstra, J.K. (Eds.), Local Search in Combinatorial Optimization. Wiley, pp. 337–360.
Labadi, N., Prins, C., Reghioui, M., 2008a. GRASP with path relinking for the capacitated arc routing problem with time windows. In: Fink, A., Rothlauf, F. (Eds.), Advances in Computational Intelligence in Transport, Logistics, and Supply Chain Management, Studies in Computational Intelligence, vol. 144. Springer, Berlin, pp. 111–135.
Labadi, N., Prins, C., Reghioui, M., 2008b. A memetic algorithm for the vehicle routing problem with time windows. RAIRO-Oper. Res. 42, 415–431.
Labadi, N., Prins, C., Reghioui, M., 2008c. An evolutionary algorithm with distance measure for the split delivery capacitated arc routing problem. In: Cotta, C., Van Hemert, J. (Eds.), Recent Advances in Evolutionary Computation for Combinatorial Optimization, Studies in Computational Intelligence, vol. 153. Springer, Berlin, pp. 275–294.
Lacomme, P., Prins, C., Ramdane-Chérif, W., 2001. Competitive genetic algorithms for the capacitated arc routing problem and its extensions. In: Boers, E.J.W. et al. (Eds.), Evo Workshops 2001, Lecture Notes in Computer Science, vol. 2037. Springer, Berlin, pp. 473–483.
Lacomme, P., Prins, C., Ramdane-Chérif, W., 2004. Competitive memetic algorithms for arc routing problems. Ann. Oper. Res. 131, 159–185.
Lacomme, P., Prins, C., Ramdane-Chérif, W., 2005. Evolutionary algorithms for periodic arc routing problems. Eur. J. Oper. Res. 165, 535–553.
Lacomme, P., Prins, C., Sevaux, M., 2006. A genetic algorithm for a bi-objective capacitated arc routing problem. Comput. Oper. Res. 33, 3473–3493.
Lacomme, P., Toussaint, H., Duhamel, C., 2013. A GRASP x ELS for the vehicle routing problem with basic three-dimensional loading constraints. Eng. Appl. Artif. Intell. 26, 1795–1810.
Laporte, G., Semet, F., 2002. Classical and new heuristics for the vehicle routing problem. In: Toth, P., Vigo, D. (Eds.), The Vehicle Routing Problem. SIAM, Philadelphia, pp. 109–128.
Liu, S., Huang, W., Ma, H., 2009. An effective genetic algorithm for the fleet size and mix vehicle routing problems. Transport. Res. Part E 45, 434–445.
Liu, R., Jiang, Z., Geng, N., Liu, T., 2010a. A heuristic approach for the open capacitated arc routing problem. In: Int. Conf. on Supply Chain Management and Information Systems (SCMIS), IEEE, 6p.
Liu, R., Jiang, Z., Liu, X., Chen, F., 2010b. Task selection and routing problems in collaborative truckload transportation. Transport. Res. Part E 46, 1071–1085.
Mei, Y., Tang, K., Yao, X., 2011. A memetic algorithm for the periodic capacitated arc routing problem. IEEE Trans. Syst. Man Cybernet. – Part B 41, 1654–1667.
Mendoza, J.E., Castanier, B., Gueret, C., Medaglia, A.L., Velasco, N., 2010. A memetic algorithm for the multi-compartment vehicle routing problem with stochastic demands. Comput. Oper. Res. 37, 1886–1898.
Mosheiov, G., 1998. Vehicle routing with pick-up and delivery: tour-partitioning heuristics. Comput. Indust. Eng. 34 (3), 669–684.
Mourão, M.C., Nunes, A.C., Prins, C., 2009. Heuristic methods for the sectoring arc routing problem. Eur. J. Oper. Res. 196, 856–868.
Nagy, G., Salhi, S., 1996. A nested location-routing heuristic using route length estimation. Stud. Locational Anal. 10, 109–127.

Ngueveu, S.U., Prins, C., Wolfler Calvo, R., 2010. An effective evolutionary algorithm for the cumulative capacitated vehicle routing problem. Comput. Oper. Res. 37, 1877–1885.
Nguyen, V.-P., Prins, C., Prodhon, C., 2012a. Solving the two-echelon location routing problem by a hybrid GRASP path relinking complemented by a learning process. Eur. J. Oper. Res. 216, 113–126.
Nguyen, V.-P., Prins, C., Prodhon, C., 2012b. A multi-start iterative local search with tabu list and path relinking for the two-echelon location routing problem. Eng. Appl. Artif. Intell. 25, 56–71.
Prins, C., 2004. A simple and effective evolutionary algorithm for the vehicle routing problem. Comput. Oper. Res. 31, 1985–2002.
Prins, C., 2009a. Two memetic algorithms for heterogeneous fleet vehicle routing problems. Eng. Appl. Artif. Intell. 22, 916–928.
Prins, C., 2009b. A GRASP $\times$ evolutionary local search hybrid for the vehicle routing problem. In: Pereira, F.B., Tavares, J. (Eds.), Bio-inspired algorithms for the vehicle routing problem, Studies in Computational Intelligence, vol. 16. Springer, Berlin, pp. 35–53.
Prins, C., Bouchenoua, A., 2004. A memetic algorithm solving the VRP, the CARP, and more general problems with nodes, edges and arcs. In: Hart, W., Krasnogor, N., Smith, J. (Eds.), Recent advances in memetic algorithms and related search technologies. Springer, Berlin, pp. 65–85.
Prins, C., Prodhon, C., Calvo, R. Wolfler, 2006. A memetic algorithm with population management (MA|PM) for the capacitated location-routing problem. In: Gottlieb, J., Raidl, G.R. (Eds.), EvoCOP 2006, Lecture Notes in Computer Science, vol. 3906. Springer, Berlin, pp. 183–194.
Prins, C., Labadi, N., Reghioui, M., 2009. Tour splitting algorithms for vehicle routing problems. Int. J. Prod. Res. 47, 507–535.
Prodhon, C., Prins, C., 2008. A memetic algorithm with population management for the periodic location-routing problem. In: Blesa, M.J. et al. (Eds.), Hybrid Metaheuristics 2008, Lecture Notes in Computer Science, vol. 5296. Springer, Berlin, pp. 43–57.
Qu, Y., Bard, J.F., 2013. The heterogeneous pickup and delivery problem with configurable vehicle capacity. Transport. Res. Part C: Emerg. Technol. 32, 1–20.
Reghioui, M., Prins, C., Labadi, N., 2007. GRASP with path relinking for the capacitated arc routing problem with time windows. In: Giacobini, M. et al. (Eds.), Applications of evolutionary computing, Lecture Notes in Computer Science, vol. 4448. Springer, Berlin, pp. 722–731.
Reiter, P., Gutjahr, W.J., 2012. Exact hybrid algorithms for solving a bi-objective vehicle routing problem. Central Eur. J. Oper. Res. 20, 19–43.
Renaud, J., Boctor, F.F., Laporte, G., 1996. An improved petal heuristic for the vehicle routeing problem. J. Oper. Res. Soc. 47, 329–336.
Repoussis, P.P., Tarantilis, C.D., 2010. Solving the fleet size and mix vehicle routing problem with time windows via adaptive memory programming. Transport. Res. Part C: Emerg. Technol. 18 (5), 695–712.
Ryan, D.M., Hjorring, C., Glover, F., 1993. Extensions of the petal method for the vehicle routing. J. Oper. Res. Soc. 44, 289–296.
Salhi, S., Sari, M., 1997. A multi-level composite heuristic for the multi-depot vehicle fleet mix problem. Eur. J. Oper. Res. 103, 95–112.
Santos, L., Coutinho-Rodrigues, J., Current, J.R., 2010. An improved ant colony optimization based algorithm for the capacitated arc routing problem. Transport. Res. Part B 44, 246–266.
Sörensen, K., 2006. Route stability in vehicle routing decisions: a bi-objective approach using metaheuristics. Central Eur. J. Oper. Res. 14, 193–207.
Tang, K., Mei, Y., Yao, X., 2009. Memetic algorithm with extended neighborhood search for capacitated arc routing problems. IEEE Trans. Evol. Comput. 13, 1151–1166.
Ulusoy, G., 1985. The fleet size and mix problem for capacitated arc routing. Eur. J. Oper. Res. 22, 329–337.
Ursani, Z., Essam, D., Cornforth, D., Stocker, R., 2011. Localized genetic algorithm for vehicle routing problem with time windows. Appl. Soft Comput. 11, 5375–5390.
Velasco, N., Castagliola, P., Dejax, P., Guéret, C., Prins, C., 2009. A memetic algorithm for a pick-up and delivery problem by helicopter. In: Pereira, F.B., Tavares, J. (Eds.), Bio-inspired Algorithms for the Vehicle Routing Problem, Studies in Computational Intelligence, vol. 161. Springer, Berlin, pp. 173–190.
Velasco, N., Dejax, P., Guéret, C., Prins, C., 2012. A non-dominated sorting genetic algorithm for a bi-objective pickup and delivery problem. Eng. Optim. 44, 305–325.
Vidal, T., Crainic, T.G., Gendreau, M., Lahrichi, N., Rei, W., 2012. A hybrid genetic algorithm for multi-depot and periodic vehicle routing problems. Oper. Res. 60 (3), 611–624.
Vidal, T., Crainic, T.G., Gendreau, M., Prins, C., 2013a. A hybrid genetic algorithm with adaptive diversity management for a large class of vehicle routing problems with time windows. Comput. Oper. Res. 40 (1), 475–489.
Vidal, T., Crainic, T.G., Gendreau, M., Prins, C., 2013b. A unified solution framework for multi-attribute vehicle routing problems. Eur. J. Oper. Res., doi 10.1016/j.ejor.2013.09.045.
Vidal, T., Crainic, T.G., Gendreau, M., Prins, C., 2014. Implicit depot assignments and rotations in vehicle routing heuristics. Eur. J. Oper. Res., http://dx.doi.org/10.1016/j.ejor.2013.12.0442014.
Villegas, J.G., Prins, C., Prodhon, C., Medaglia, A.L., Velasco, N., 2010. GRASP/VND and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots. Eng. Appl. Artif. Intell. 23 (5), 780–794.
Villegas, J.G., Prins, C., Prodhon, C., Medaglia, A.L., Velasco, N., 2011. GRASP with evolutionary path relinking for the truck and trailer routing problem. Comput. Oper. Res. 38 (9), 1319–1334.
Wøhlk, S., 2008. An approximation algorithm for the capacitated arc routing problem. Open Oper. Res. J. 2, 8–12.
Xing, L., Rohlfshagen, P., Chen, Y., Yao, X., 2010. An evolutionary approach to the multi-depot capacitated arc routing problem. IEEE Trans. Evol. Comput. 14, 356–374.