

*Citation for published version:*

Lai, M, Battarra, M, Di Francesco, M & Zuddas, P 2015, 'An adaptive guidance meta-heuristic for the vehicle routing problem with splits and clustered backhauls', *Journal of the Operational Research Society*, vol. 66, no. 7, pp. 1222-1235. <https://doi.org/10.1057/jors.2014.123>

*DOI:*

[10.1057/jors.2014.123](https://doi.org/10.1057/jors.2014.123)

*Publication date:*

2015

*Document Version*

Early version, also known as pre-print

[Link to publication](#)

This is a post-peer-review, pre-copyedit version of an article published in Journal of the Operational Research Society. The definitive publisher-authenticated version Lai, M., Battarra, M., Di Francesco, M., & Zuddas, P. (2015). An adaptive guidance meta-heuristic for the vehicle routing problem with splits and clustered backhauls. *Journal of the Operational Research Society*, 66(7), 1222-1235. 10.1057/jors.2014.123 is available online at: <http://dx.doi.org/10.1057/jors.2014.123>.

## University of Bath

**General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

**Take down policy**

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

# An adaptive guidance meta-heuristic for the Vehicle Routing Problem with Splits and Clustered Backhauls

Michela Lai<sup>a</sup>, Maria Battarra<sup>b</sup>, Massimo Di Francesco<sup>a</sup>, Paola Zuddas<sup>a</sup>

<sup>a</sup>*Department of Mathematics and Computer Science, University of Cagliari, Italy*

<sup>b</sup>*School of Mathematics, University of Southampton, UK*

---

## Abstract

This paper presents a vehicle routing problem, where trucks deliver container loads from a port to import customers and collect container loads from export customers to the same port. In each route, import customers must be serviced before export customers and each customer can be visited more than once. We model the problem using an Integer Linear Programming formulation and propose an Adaptive Guidance metaheuristic. Our extensive computational experiments show that the adaptive guidance algorithm is capable of solving average-sized instances within limited computing time.

*Keywords:* Vehicle Routing Problem with Splits, Backhauls, Drayage, Adaptive Guidance, Meta-heuristics

---

## 1. Introduction

2 This paper addresses a vehicle routing problem motivated by the case study  
3 of the Italian carrier Grendi Trasporti Marittimi, which provides *door-to-door*  
4 freight transportation services. The carrier manages a homogeneous fleet of  
5 trucks and containers based at the port of Vado Ligure (Italy). Trucks move  
6 container loads from the port to import customers and from export customers  
7 to the port.

8 It is important to note that in this problem containers are not picked up  
9 or delivered. They are brought to the customers, where they are packed or  
10 unpacked and moved away by the same trucks. Therefore, while containers  
11 are emptied at importer locations, drivers supervise the unloading operations  
12 and wait for empty containers to be returned. Similarly, trucks move empty  
13 containers to export customers, drivers supervise packing operations and wait  
14 for loaded containers to be returned. The truck and the containers are coupled  
15 in the sense that the truck carries the same set of containers throughout the  
16 route.

17 From the customer's point of view, this practice is perceived as a high quality  
18 service, because the loading and unloading operations are closely supervised and  
19 the integrity of the cargo is monitored. From the carrier's point of view, this

20 policy improves container safety and integrity, because containers are never left  
21 unsupervised at customer locations.

22 More important, the carrier is aware of the fact that leaving containers at  
23 customer locations would save drivers the time to supervise loading and unload-  
24 ing operations and they could move to other customers in the meanwhile (Che-  
25 ung et al., 2008). The profitability of this alternative policy depends on the  
26 availability of inland depots close to the customers, but inland depots are not  
27 often financially feasible for small carriers.

28 In this case-study, the container loads of export customers are typically  
29 not ready before the afternoon, thus the carrier serves import customers before  
30 exporters. Moreover, the containers emptied at importers can be filled at export  
31 customers, hence a potential routing cost saving can be obtained.

32 Since the number of containers loads to be delivered to importers and picked  
33 from exporters is possibly different, trucks may be required to leave and enter  
34 the port carrying some empty containers. More precisely, if the number of  
35 container loads to be delivered is larger than the number of container loads  
36 to be picked up, trucks return empty containers back to the port. Otherwise,  
37 trucks leave the port carrying empty containers to accommodate the requests  
38 of all export customers.

39 Importers and exporters often demand a number of container loads larger  
40 than the truck’s capacity. Hence, splitting customer demand may be compul-  
41 sory and each customer may be visited more than once. Moreover, customer  
42 demands can be split among several trucks, even if the demand is lower than  
43 the capacity. The objective is to determine a set of routes in which routing  
44 costs are minimized, all customers are serviced, importers are visited before  
45 exporters, and the capacities of trucks are never exceeded.

46 According to the problem classification in Parragh et al., 2008, this problem  
47 belongs to the class of Vehicle Routing Problems with Clustered Backhauls  
48 (VRPCB), because in each route all deliveries must be performed before all  
49 pickups. However, in classical VRPCB, each customer must be visited only  
50 once, whereas in this problem multiple visits at each customer are allowed. Our  
51 problem also belongs to the class of the so-called *one-to-many-to-one* pickup  
52 and delivery problems, because all delivery demands are initially located at the  
53 port and all pickup demands are destined to the same port (Berbeglia et al.,  
54 2007).

55 This problem is called hereafter Split Vehicle Routing Problem with Clus-  
56 tered Backhauls (SVRPCB) and, as far as we are aware, it has not been ad-  
57 dressed in its current form in the literature before. In this paper, linehaul  
58 customers are referred as import customers, delivery customers or importers.  
59 In the same way, backhaul customers are also called export customers, pickup  
60 customers or exporters. Similarly, let importer routes and exporter routes be  
61 the routes serving only importers or exporters, respectively.

62 An Integer Linear Programming (ILP) model is presented to address small-  
63 sized problems. In order to solve larger instances, we propose a meta-heuristic  
64 which exploits existing algorithms for simpler SVRPCB subproblems and guides  
65 them toward the construction of good SVRPCB solutions. More precisely, the

66 meta-heuristic constructs a feasible SVRPCB solution by first decomposing the  
67 SVRPCB into two Split Vehicle Routing Problems (SVRP), where the first sub-  
68 problem involves only importers and the second only exporters. These problems  
69 are solved by the Tabu Search (TS) of Archetti et al., 2006. Next, importer and  
70 exporter routes are paired and merged by solving an assignment problem. This  
71 two-stage constructive heuristic is the building block for the proposed meta-  
72 heuristic.

73 However, the importer routes and exporter routes by the TS could not result  
74 in good SVRPCB solutions. Therefore, at each iteration of the proposed algo-  
75 rithm, critical properties of the current SVRPCB solution are detected. Some  
76 guidance mechanisms are implemented by perturbing the data of the two SVRP,  
77 in order to discourage the TS in creating routes having undesired characteristics.

78 This paper not only proposes a meta-heuristic algorithm for the SVRPCB,  
79 but also aims at investigating the effect of the growth in transportation capac-  
80 ities on the carrier’s service. The possibility of employing trucks with larger  
81 capacities than a single container is considered. This allows the carrier to esti-  
82 mate the savings in adopting larger vehicles.

83 The rest of the paper is organized as follows. In Section 2, we review the  
84 related literature and in Section 3 we present the ILP formulation. In Section 4,  
85 the meta-heuristic based on Adaptive Guidance mechanisms is proposed. In  
86 Section 5, the results of our extensive computational experience are presented  
87 and a comparison between the performances of a state-of-art solver and the  
88 meta-heuristic algorithms is reported. Finally, conclusions and further research  
89 directions are summarized in Section 6.

## 90 2. Literature Review

91 Several papers address the VRPCB, where all linehauls are visited before  
92 backhauls and each customer must be visited exactly once. Exact methods  
93 for the VRPCB are proposed by Mingozzi et al., 1999 and Toth and Vigo,  
94 1997. Heuristics have been developed by Anily, 1996, Goetschalckx and Jacobs-  
95 Blecha, 1989, Toth and Vigo, 1999, Osman and Wassan, 2002, Brandão, 2006,  
96 Ropke and Pisinger, 2006 and Zachariadis and Kiranoudis, 2012. Recently, the  
97 unified hybrid genetic search algorithm of Vidal et al., 2012 provided the most  
98 competitive results for the VRPCB. We refer to the surveys of Gribkovskaia  
99 and Laporte, 2008 and Toth and Vigo, 2002 for the single-vehicle and multiple-  
100 vehicle problems, respectively.

101 What makes the SVRPCB different from the VRPCB is the possibility to  
102 serve customers more than once. A recent review on SVRP was presented by  
103 Archetti and Speranza, 2012.

104 Some attributes of the SVRPCB can be found in Mitra, 2005 and Mitra,  
105 2008. These papers consider a homogeneous fleet of vehicles located at a depot  
106 to serve delivery and pickup demands of a set of customers. Although splitting  
107 is allowed, unlike in the SVRPCB, importers and exporters can be visited in  
108 any order. Mitra, 2005 developed a Mixed Integer Linear Programming (MILP)

109 formulation for the problem and presented a route construction heuristic, which  
110 improved the best known solutions obtained by the MILP formulation. Mitra,  
111 2008 further investigated this problem designing a parallel clustering technique  
112 and route construction heuristic.

113 In the field of intermodal freight transportation, the distribution of con-  
114 tainers by trucks between customers and intermodal terminals is known as  
115 “drayage”. According to Macharis and Bontekoning, 2004, drayage involves  
116 the distribution of a full container from an intermodal terminal to a receiver  
117 and the subsequent collection of an empty container, or the provision of an  
118 empty container to a shipper for the subsequent transportation of a full con-  
119 tainer. This definition accounts for both policies where trucks and containers  
120 are separated or coupled, as in the SVRPCB.

121 The separation of trucks and containers has been investigated by Jula et al.,  
122 2005, Chung et al., 2007, Zhang et al., 2011, Zhang et al., 2010, Vidovic et al.,  
123 2011, Braekers et al., 2013 and Nossack and Pesch, 2013. The variant where  
124 trucks and containers are coupled received less attention, in fact it has been  
125 investigated only in papers motivated by specific technical restrictions (i.e., Imai  
126 et al., 2007) or regulation policies (Cheung et al., 2008).

127 From a methodological point of view, the latter variant was investigated  
128 by Imai et al., 2007, who formulated their problem as the optimal assignment  
129 of trucks to a set of delivery and pickup pairs. They developed a subgradient  
130 heuristic based on Lagrangian Relaxation. However, trucks cannot visit more  
131 than one importer or one exporter in a single trip, because they can carry one  
132 container only. Caris and Janssens, 2009 modeled the container drayage prob-  
133 lem as a full truckload pickup and delivery problem with time windows. They  
134 constructed an initial solution by a two-phase insertion heuristic and improved  
135 it using a local search heuristic based on three neighborhoods. Yet, in their  
136 problem setting, each truck carries one container only. Lai et al., 2013 investi-  
137 gated how to deliver and collect container loads by trucks carrying one or two  
138 containers. A feasible solution was built using an adaptation of the Clarke and  
139 Wright, 1964 algorithm and it was improved using two neighborhoods. Hence,  
140 this algorithm cannot be used for trucks carrying more than two containers.

141 To conclude, a frequent characteristic of papers on drayage is the assump-  
142 tion that trucks carry at most one container (Jula et al., 2005, Namboothiri  
143 and Erera, 2008, Zhang et al., 2011, Zhang et al., 2010 and Sterzik and Kopfer,  
144 2013). However, if the weight of the containers is under a set value, the capacity  
145 of trucks could be higher than one container. Carrying two or more containers  
146 per truck is allowed in many countries (Nagl, 2007). Since larger capacities can  
147 increase the efficiency of the distribution, this paper investigates this opportu-  
148 nity and aims at quantifying its benefits. However, it is important to note that  
149 this opportunity substantially increases the difficulty of SVRPCB, because the  
150 underlying packing problem becomes more difficult to solve.

151 **3. Formulation**

152 This section introduces the notation and presents an ILP model for the  
 153 SVRPCB. Let  $p$  be the port,  $I$  the set of importers,  $E$  the set of exporters and  
 154  $K$  the set of trucks, each with capacity  $Q$ -containers. Let  $d_i$  be the number of  
 155 containers used to serve customer  $i \in I \cup E$ . If  $i \in I$ ,  $d_i$  represents the number of  
 156 containers used to deliver container loads to import customer  $i \in I$ . If  $i \in E$ ,  $d_i$   
 157 represents the number of containers used to pick up container loads from export  
 158 customer  $i \in I$ .

159 Given a direct graph  $G = (N, A)$ , the set  $N$  is defined as  $N = \{p \cup I \cup E\}$ .  
 160 Since trucks are not allowed to move from exporters to importers, the set  $A$   
 161 of arcs is defined as  $A = A_1 \cup A_2$ , where  $A_1 = \{(i, j) | i \in p \cup I, j \in N, i \neq j\}$   
 162  $A_2 = \{(i, j) | i \in E, j \in p \cup E, i \neq j\}$ . Three sets of variables are defined:

- 163  $x_{ij}^k$ : Routing selection variables taking value 1 if arc  $(i, j) \in A$  is traversed by  
 164 truck  $k \in K$ , 0 otherwise; let  $c_{ij} \geq 0$  be the cost of traversing arc  $(i, j)$ ;  
 165  $y_{ij}^k$ : Number of loaded containers carried along arc  $(i, j) \in A$  by truck  $k \in K$ ;  
 166  $z_{ij}^k$ : Number of empty containers carried along arc  $(i, j) \in A$  by truck  $k \in K$ .

167 The problem can be formulated as follows:

$$\min \sum_{k \in K} \sum_{(i,j) \in A} c_{ij} x_{ij}^k \quad (1)$$

s.t.

$$\sum_{k \in K} \sum_{l \in N} y_{il}^k = \sum_{k \in K} \sum_{j \in p \cup I} y_{ji}^k - d_i \quad \forall i \in I \quad (2)$$

$$\sum_{k \in K} \sum_{l \in N} z_{il}^k = \sum_{k \in K} \sum_{j \in p \cup I} z_{ji}^k + d_i \quad \forall i \in I \quad (3)$$

$$\sum_{l \in N} y_{il}^k \leq \sum_{j \in p \cup I} y_{ji}^k \quad \forall i \in I, \forall k \in K \quad (4)$$

$$\sum_{l \in N} z_{il}^k \geq \sum_{j \in p \cup I} z_{ji}^k \quad \forall i \in I, \forall k \in K \quad (5)$$

$$\sum_{k \in K} \sum_{l \in p \cup E} y_{il}^k = \sum_{k \in K} \sum_{j \in N} y_{ji}^k + d_i \quad \forall i \in E \quad (6)$$

$$\sum_{k \in K} \sum_{l \in p \cup E} z_{il}^k = \sum_{k \in K} \sum_{j \in N} z_{ji}^k - d_i \quad \forall i \in E \quad (7)$$

$$\sum_{l \in p \cup E} y_{il}^k \geq \sum_{j \in N} y_{ji}^k \quad \forall i \in E, \forall k \in K \quad (8)$$

$$\sum_{l \in p \cup E} z_{il}^k \leq \sum_{j \in N} z_{ji}^k \quad \forall i \in E, \forall k \in K \quad (9)$$

$$\sum_{(ji) \in A} (y_{ji}^k + z_{ji}^k) = \sum_{(il) \in A} (y_{il}^k + z_{il}^k) \quad \forall i \in I \cup E, \forall k \in K \quad (10)$$

$$y_{ij}^k + z_{ij}^k \leq Q x_{ij}^k \quad \forall (i, j) \in A, \forall k \in K \quad (11)$$

$$\sum_{j \in N} x_{ji}^k - \sum_{l \in N} x_{il}^k = 0 \quad \forall i \in N, \forall k \in K \quad (12)$$

$$\sum_{j \in N} x_{ij}^k \leq 1 \quad \forall i \in N, \forall k \in K \quad (13)$$

$$\sum_{k \in K} \sum_{i \in I \cup E} z_{ip}^k - \sum_{k \in K} \sum_{i \in I \cup E} z_{pi}^k = \sum_{i \in I} d_i - \sum_{i \in E} d_i \quad (14)$$

$$x_{ij}^k \in \{0, 1\} \quad \forall (i, j) \in A, \forall k \in K \quad (15)$$

$$y_{ij}^k \in \{0, 1, \dots, Q\} \quad \forall (i, j) \in A, \forall k \in K \quad (16)$$

$$z_{ij}^k \in \{0, 1, \dots, Q\} \quad \forall (i, j) \in A, \forall k \in K \quad (17)$$

168 Routing costs are minimized in the objective function (1).

169 Constraints (2)-(5) concern the distribution of containers to importers. Con-  
 170 straints (2) and (3) are the flow conservation constraints of loaded and empty  
 171 containers, respectively, at each importer node. Constraints (4) enforce that  
 172 the number of loaded containers cannot increase after servicing any importer,  
 173 whereas constraints (5) guarantee that the number of empty containers does  
 174 not decrease.

175 Similarly, constraints (6)-(9) concern the distribution of containers to ex-  
 176 porters. Constraints (6) and (7) are the flow conservation constraints of loaded  
 177 and empty containers, respectively, for each exporter. Constraints (8) and (9)  
 178 enforce that the number of loaded containers cannot decrease after visiting an  
 179 exporter, whereas the number of empty containers cannot increase.

180 Constraints (10) guarantee that the number of containers carried by each  
 181 truck does not change after visiting a customer. Constraints (11) impose that  
 182 the number of containers on each truck does not exceed the capacity  $Q$ .

183 Constraints (12) represent the flow conservation constraints for each truck  
 184 at each node. Constraints (13) enforces that each truck can reach only one node  
 185 from the current node. It is important to note that constraints (12) and (13)  
 186 enforce that the degree of each node must be at most 2. This forces a vehicle to  
 187 visit the same customer at most once in a route. Moreover, if there is a successor  
 188 for a node  $i$  visited in the route of truck  $k$ , Constraints (12) impose that there  
 189 is also a predecessor for the same node and the same truck. Constraints (13)  
 190 also guarantee that trucks are not used more than once.

191 Constraints (14) represent the flow conservation of empty containers at the  
 192 port  $p$ . Finally, Constraints (15), (16) and (17) define the domain of the decision  
 193 variables.

194 The model has been implemented using IBM ILOG CPLEX Optimization  
 195 Studio 12.5 and solved by ILOG CPLEX 12.2 solver. Since exact methods may  
 196 not be able to solve realistic-size instances of SVRPCB with high truck capacity,  
 197 we present a meta-heuristic, which is described in the following section.

198 **4. Meta-heuristic algorithm**

199 The proposed meta-heuristic is based on *Adaptive Guidance* (AG) mecha-  
200 nisms, which are simple rules applied to check the quality of the current solution  
201 and detect possibly improvements. Then, the input parameters of simpler sub-  
202 problems are perturbed so as to achieve the desired diversification in the complex  
203 problem at hand. Examples of successful implementations of adaptive guidance  
204 algorithms are presented in Battarra et al., 2009, Bai et al., 2007, Kramer, 2008  
205 and Olivera and Viera, 2007. Moreover, Hart, 2005 presented a large class of  
206 simple rules of behavior, called adaptive heuristics.

207 Our overall meta-heuristic consists of three phases:

208 (i) SVRP phase decomposes the SVRPCB into two SVRPs, one for im-  
209 porters and one for exporters, each solved by the TS (Glover and Laguna, 1998)  
210 proposed by Archetti et al., 2006.

211 (ii) Merging phase merges importer routes and exporter routes determined  
212 in SVRP phase by an ILP model based on the saving concept;

213 (iii) AG phase analyses the current solution, detects areas of improvement  
214 and adjusts the input parameters of the SVRP phase.

215 The three phases are repeated sequentially until a stop criterion is satisfied  
216 and the best solution found is returned.

217 Table 1 illustrates the pseudo-code of the meta-heuristic algorithm, in which  
218 the following notation is adopted:

219 **tExe** Execution time;

220 **it** Number of consecutive iterations performed during the whole execution;

221 **notImpIt** Number of consecutive iterations performed since an improving so-  
222 lution was found;

223 **S\*** Best solution found;

224 **MAXTIME** Maximum execution time;

225 **MAXIT** Maximum number of consecutive iterations allowed during the whole  
226 execution;

227 **SolImp** Set of importer routes determined in the SVRP phase by the TS solving  
228 the SVRP on the set  $I$  of importers.

229 **SolExp** Set of exporter routes determined in the SVRP phase by the TS solving  
230 the SVRP on the set  $E$  of exporters.

231 **Sol** Current solution of the meta-heuristic;

232 **SMatrix** Matrix of all savings that can be obtained by merging importer routes  
233 and exporter routes;



234 **Merge**( $Sol$ ,  $SolImp$ ,  $SolExp$ ,  $SMatrix$ ) Function merging routes determined  
235 in the SVRP phase by an ILP model. The input parameters are the cur-  
236 rent solution  $Sol$ , the set of importer routes  $SolImp$  and exporter routes  
237  $SolExp$ , and the saving matrix  $SMatrix$ . The output is the new current  
238 solution  $Sol$ ;

239 **AdaptiveGuidance**( $Sol$ ,  $SolImp$ ,  $SolExp$ ,  $it$ ) Function analyzing the cur-  
240 rent solution  $Sol$  according to different criteria (or guidance mechanisms)  
241 and perturbing the costs in the SVRP phase. Since it is not compulsory  
242 to perform all mechanisms at each iteration, this function depends on the  
243 current number of iterations  $it$ .

---

```

procedure MAIN
  Start  $tExe$ 
   $it = 0$ 
   $notImpIt = 0$ 
   $S^* \leftarrow \emptyset$ 
  while  $tExe \leq MAXTIME$  &  $notImpIt \leq MAXIT$  do
     $it = it + 1$ 
     $notImpIt = notImpIt + 1$ 
     $SolImp \leftarrow TS(I)$ ; ▷ SVRP phase Section 4.1
     $SolExp \leftarrow TS(E)$ ;
    Create the savings matrix  $SMatrix$ 
     $Sol \leftarrow \emptyset$ 
     $Sol \leftarrow MERGE(Sol, SolImp, SolExp, SMatrix)$  ▷ Merging phase
  Section 4.2
  if  $Sol \leq S^*$  ||  $S^* == \emptyset$  then
     $S^* \leftarrow \emptyset$ 
     $S^* \leftarrow Sol$ 
     $notImpIt \leftarrow 0$ 
  end if
  ADAPTIVEGUIDANCE( $Sol, SolImp, SolExp, it$ ) ▷ AG phase Section 4.3
end while
return  $S^*$ 
end procedure

```

---

Table 1: The structure of the meta-heuristic.

244 In the following, the three phases of the algorithm are described in detail.

#### 245 4.1. SplitVRP phase

246 The SVRP phase consists of solving two SVRPs: the first involves importers  
247 only, whilst the second exporters only. As stated previously, the TS by Archetti  
248 et al., 2006 is employed to solve this NP-hard problem. The algorithm consists  
249 of three phases: (i) the first phase determines the initial feasible solution con-  
250 structing a giant tour by the GENIUS algorithm (Gendreau et al., 1992) and  
251 imposing trucks to return to the depot whenever their load equals the capac-  
252 ity; (ii) the second phase consists of a TS based on relocation moves, where

253 a customer is either relocated into another route or copied into an alternative  
 254 route. In the latter case, its original demand is split between the two routes;  
 255 (iii) the third phase improves the solution found by removing t-split cycles and  
 256 by re-optimizing each route using the GENIUS algorithm.

#### 257 4.2. Merging phase

258 Routes determined in the SVRP phase are merged in the Merging phase  
 259 according to an ILP model, which is inspired by the Clarke and Wright savings  
 260 algorithm (Clarke and Wright, 1964). In this algorithm savings are obtained by  
 261 merging a route servicing importers with a route servicing exporters, instead of  
 262 leaving them separate. It is important to note that, four possible routes can be  
 263 generated by merging a selected pair of routes, because the first and the last  
 264 importer may be linked to the first or the last exporter. To clarify, consider  
 265 for instance  $n$  importers, serviced by route  $r_i = \{p, i_1, \dots, i_n, p\}$ , and  $m$   
 266 exporters serviced by route  $r_j = \{p, e_1, \dots, e_m, p\}$ . Moreover, let  $c(i_n, e_1)$  be  
 267 the cost of arc  $(i_n, e_1) \in A$ , and so on. When the merging of routes  $r_i$  and  $r_j$   
 268 is evaluated, the algorithm computes four different savings based on the extra  
 269 mileage evaluation:

- 270 •  $s_{ij}^1 = c(i_n, p) + c(p, e_1) - c(i_n, e_1)$ , where routes  $r_i$  and  $r_j$  keep their original  
 271 direction in the final route, i.e. importers are visited from  $i_1$  to  $i_n$  and  
 272 exporters from  $e_1$  to  $e_m$ ;
- 273 •  $s_{ij}^2 = c(i_n, p) + c(p, e_1) + c(e_m, p) - c(i_n, e_m) - c(e_1, p)$ , where in the final  
 274 route  $r_i$  has the original direction and  $r_j$  the opposite one, i.e. importers  
 275 are visited from  $i_1$  to  $i_n$  and exporters from  $e_m$  to  $e_1$ ;
- 276 •  $s_{ij}^3 = c(p, i_1) + c(i_n, p) + c(p, e_1) - c(p, i_n) - c(i_1, e_1)$ , where in the final route  
 277  $r_i$  has the opposite direction and  $r_j$  the original one, i.e. importers are  
 278 visited from  $i_n$  to  $i_1$  and exporters from  $e_1$  to  $e_m$ ;
- 279 •  $s_{ij}^4 = c(p, i_1) + c(i_n, p) + c(p, e_1) + c(e_m, p) - c(p, i_n) - c(i_1, e_m) - c(e_1, p)$ ,  
 280 where routes  $r_i$  and  $r_j$  have the opposite direction in the final route, i.e.  
 281 importers are visited from  $i_n$  to  $i_1$  and exporters from  $e_m$  to  $e_1$ ;

282 Each pair of routes is supposed to be merged according to the maximum  
 283 saving. Therefore, the saving generated by merging routes  $r_i$  and  $r_j$  is  $s_{ij} =$   
 284  $\max\{s_{ij}^1, s_{ij}^2, s_{ij}^3, s_{ij}^4\}$ . Maximum savings are recorded in a matrix, in which the  
 285 number of rows is equal to  $|SolImp|$  and the number of columns is equal to  
 286  $|SolExp|$ .

287 Routes determined in SVRP phase are merged in the Merging phase by the  
 288 following assignment problem. For all  $i \in SolImp$  and  $j \in SolExp$ , let  $w_{ij}$  be a  
 289 binary variable, which takes value 1 if routes  $r_i$  and  $r_j$  are merged, 0 otherwise.  
 290 The assignment problem can be formulated by the following ILP model:

$$\max \sum_{i \in SolImp} \sum_{j \in SolExp} s_{ij} w_{ij} \quad (18)$$

s.t.

$$\sum_{j \in SolExp} w_{ij} \leq 1 \quad \forall i \in SolImp \quad (19)$$

$$\sum_{i \in SolImp} w_{ij} \leq 1 \quad \forall j \in SolExp \quad (20)$$

$$w_{ij} \in \{0, 1\} \quad \forall i \in SolImp, j \in SolExp \quad (21)$$

291 The overall gain is maximized in the objective function (18), where  $s_{ij}$  rep-  
 292 represents the maximum saving obtained by merging routes  $i$  and  $j$ , as described  
 293 above.

294 Constraints (19) and (20) enforce that each route in  $SolImp$  can be merged  
 295 at most with a route in  $SolExp$  and vice-versa. We do not consider merging  
 296 operations involving more than an importer route and an exporter route, because  
 297 it is quite unlikely that feasible SVRPCB solutions would be obtained due to  
 298 the violation of the capacity constraint.

#### 299 4.3. Adaptive guidance phase

300 The AG phase analyses the incumbent SVRPCB solution according to pre-  
 301 defined criteria, each of which gives rise to a guidance mechanism. If drawbacks  
 302 are detected in the solution, the input data of the TS are suitably perturbed by  
 303 guidance mechanisms, which are implemented by penalizing costs in the SVRP  
 304 phase. In this section we illustrate how to identify drawbacks in the incumbent  
 305 solution, define quantitative measures for their evaluation and design suitable  
 306 penalization mechanisms that would result in the desired diversification effect,  
 307 without corrupting the original SVRPCB input data.

308 Our meta-heuristic is guided by the following guidance mechanisms:

##### 309 (i) A.G.M.1 - Avoiding too many Splits

310 Since the TS tends to generate routes where load splitting is allowed, the  
 311 resulting SVRPCB solutions may be likely poor when the number of visits to  
 312 customers is unnecessarily high. This guidance mechanism is aimed at correcting  
 313 this drawback. Given a customer  $i$ , let  $minTrip_i = \lceil d_i/Q \rceil$  be the minimum  
 314 number of visits required to satisfy its demand, let  $visit_i$  be the number of visits  
 315 to customer  $i$  in the current solution and let  $exceed_i$  be the difference between  
 316  $visit_i$  and  $minTrip_i$ . This guidance mechanism selects the importer and the  
 317 exporter with the largest positive values of  $exceed_i$ , if any. A penalization  
 318 is introduced for all arcs entering and leaving these customers in the next  $\gamma$   
 319 iterations, in order to guide the TS toward a lower use of arcs connecting these  
 320 customers and, hence, split the load into a lower number of routes.

321 (ii) *A.G.M.2 - Promising extreme importers*

322 The first importer and the last one play a crucial role in the SVRPCB,  
323 in fact, if they are close to export customers, the Merging phase is far more  
324 effective in connecting the importer route to an exporter route. However, the  
325 set *SolImp* of import routes determined in the SVRP phase ignores the location  
326 of exporters and, hence, the resulting SVRPCB solutions may be likely poor.  
327 This guidance mechanism aims to guide the TS, so that importers with close  
328 exporters are forced to be the first ones or the last ones in the new solution of  
329 the SVRP phase. In what follows, we refer to extreme importers instead of the  
330 first and the last importer in a route.

331 Given a importer  $i \in I$ , we denote with  $\alpha_i$  the number of times in which  
332  $i \in I$  is visited as an extreme node in the incumbent SVRPCB solution minus  
333 the number of times in which  $i \in I$  is visited as an internal node. In order  
334 to diversify the current solution, we are interested in the negative values of  $\alpha_i$ ,  
335 because they indicate customers which are frequently visited as internal nodes.  
336 Moreover, let  $\sigma_i$  be the sum of all distances between the selected importer  $i \in I$   
337 and all exporters. Since low values of  $\sigma_i$  indicate the high proximity of many  
338 exporters to the selected importer, this guidance mechanism selects the importer  
339  $i \in I$  having a negative value of  $\alpha_i$  and the minimum value of  $\sigma_i$ , if any. In order  
340 to remove customer  $i$  from its frequent position of internal node, a penalization  
341 is added in the SVRP phase to all arcs entering or leaving importer  $i$  for the  
342 next  $\gamma$  iterations.

343 (iii) *A.G.M.3 - Promising extreme exporters*

344 This mechanism works as *A.G.M.2*, but it considers extreme exporters in-  
345 stead of importers.

346 (iv) *A.G.M.4 - Avoiding expensive arcs*

347 This mechanism aims to avoid the use of the most costly arcs in the incum-  
348 bent SVRPCB solution. This guidance mechanism selects the most expensive  
349 arcs connecting pairs of importers and pairs of exporters and penalize them in  
350 the SVRP phase for the next  $\gamma$  iterations.

351 *Remarks*

352 In the proposed meta-heuristic the execution of a single guidance mechanism  
353 is iteration-dependent. As a result, at any iteration one may run a guidance  
354 mechanism with all other mechanisms, with some of them or one at a time.  
355 Hence, it is important to properly calibrate parameters controlling when each  
356 guidance mechanism should be performed during the overall execution of the  
357 algorithm.

358 *4.4. Penalizations*

Once the incumbent SVRPCB solution has been analysed according to a  
guidance mechanism, the selected arc costs are penalized in the SVRP phase

for the subsequent  $\gamma$  iterations. If arc  $(i, j)$  connects two customers, its cost is penalized as

$$c_{ij} = c_{ij} + \text{RandomCoef} \cdot M \quad (22)$$

The value  $M$  is set up as the largest entry of the cost matrix and  $\text{RandomCoef}$  is a coefficient that randomly decreases/increases the penalties during the overall execution of the algorithm, according to the formula:

$$\text{RandomCoef} = (\text{Random}(1, \dots, \alpha) + \beta)/100 \quad (23)$$

359 where  $\beta$  is a self-adapting parameter taking initial value 0 and increasing by  
 360  $\alpha$  after each  $\alpha$  iterations. Whenever a better SVRPCB solution is found,  $\beta$  is  
 361 set to 0, in order to refresh penalties.

A larger penalty is added to the cost of arcs connecting customers to the port, in order to minimize the number of trucks in the solution. More formally, if arc  $(i, j)$  connects a customer to the port, its cost is penalized as

$$c_{ij} = c_{ij} + \text{RandomCoef} \cdot M + (|N| - 1) \cdot M \quad (24)$$

362 where  $M$  is the largest entry of the cost matrix and  $N$  the set of nodes.  
 363 Moreover, whenever an improving solution is found, penalties are set to zero for  
 364 arcs linking the port to the set of importers or exporters serviced by a lower  
 365 number of routes. This allows for a lower number of routes and, hence, lower  
 366 routing costs.

367 Three different methods are proposed for the introduction of penalties. The  
 368 three methods are:

369 **(i) Unchecked penalties** Penalties are added to an arc cost, even if a penal-  
 370 ization is already applied. To clarify, if a penalty on arc  $(i, j) \in A$   
 371 is added from iteration  $it$  to  $it + \gamma$  and the arc is selected to be penalized at  
 372 iteration  $it + \delta$  (with  $\delta \leq \gamma$ ), the penalty is applied twice;

373 **(ii) Unique penalties** Penalties are applied in the next  $\gamma$  iterations on an arc  
 374 only if it is not penalized at the moment. To clarify, if a penalty on arc  
 375  $(i, j) \in A$  is applied from iteration  $it$  to  $it + \gamma$  and the arc is selected to  
 376 be penalized at iteration  $it + \delta$  (with  $\delta \leq \gamma$ ), the penalty is rejected and  
 377 the adaptive guidance mechanism is executed again, until an arc not yet  
 378 penalized is detected or no more penalties become available;

379 **(iii) Incremental unique penalties** It implements both previous penaliza-  
 380 tion strategies. To clarify, if a penalty is applied on arc  $(i, j) \in A$  from  
 381 iteration  $it$  to  $it + \gamma$  and the arc is selected to be penalized at iteration  
 382  $it + \delta$  (with  $\delta \leq \gamma$ ), the penalty is accepted and the adaptive guidance  
 383 mechanism is executed again, until an arc not yet penalized is detected  
 384 or no more penalties become available. Therefore, if a penalty on an arc  
 385  $(i, j) \in A$  is found at iterations  $it$  and  $it + \delta$  (with  $\delta \leq \gamma$ ), the penalty is  
 386 inserted twice and the adaptive guidance mechanism is executed again to  
 387 look for additional penalties.

## 388 5. Computational results

389 This section aims to analyze the performance of the proposed meta-heuristic.  
390 Our test set consists of 140 uniformly generated instances with 10, 20, 30, 40, and  
391 50 customers. Since large-sized instances are the most challenging, we generate  
392 a larger number of instances of large size (12 instances with 10 customers, 20  
393 with 20 customers, 28 with 30 customers, 36 with 40 customers and 44 with 50  
394 customers).

395 Instances with the same number of customers have the same customer lo-  
396 cations, which are integers uniformly generated between  $-1000$  and  $+1000$  and  
397 the same demands, which are integers generated according to a random uniform  
398 distribution in the range 1 to 5.

399 The ratio between the number of importers and exporters is generated as  
400 follows. Denoting by  $n$  the number of customers, we generate  $n/5 + 1$  instances.  
401 The number of importers in instance  $k \neq \{0, n/5\}$  is  $5k$ , consequently the num-  
402 ber of exporters is  $n - 5k$ . However, in order to have at least two importers and  
403 two exporters in the instance  $k = 0$  and  $k = n/5$ , the number of import and  
404 export customers for such instances is forced to be 2, respectively.

405 The number of trucks in each instance is fixed and is equal to the minimum  
406 number of trucks needed to service all container loads. It is computed as the  
407 the bin packing lower bound  $\lceil \max\{\sum_{i \in I} d_i, \sum_{i \in E} d_i\} / Q \rceil$ .

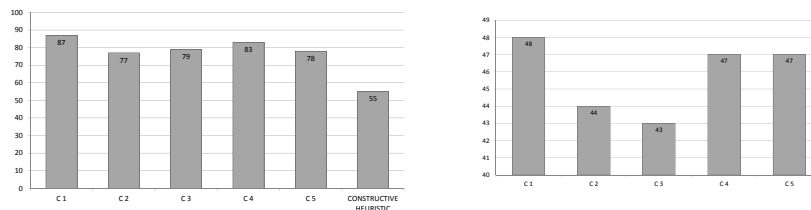
408 Twenty percent of the instances for each problem-size considers trucks carry-  
409 ing up to 1-container, 2-containers, 4-containers and 6-containers. This choice  
410 depends on the current rules adopted in several countries. Whenever the over-  
411 all load weight is not a constraint, almost all countries allow to carry up to 2  
412 containers, some others up to 4 containers. To our knowledge, in Australia up  
413 to 3 40 feet containers per truck are allowed when rail transportation is not  
414 available (Nagl, 2007). Nevertheless, these instances allow for experimenting  
415 with transportation units smaller than containers. The instances are available  
416 upon request.

### 417 5.1. Experimental Setting

418 The integer programming formulation (1)-(17) had been coded using IBM  
419 ILOG CPLEX Optimization Studio 12.5 and solved by the Branch & Cut of  
420 ILOG CPLEX 12.2. The meta-heuristic presented in Section 4 was coded in  
421 C++, and the integer model (18)-(21) is solved using the Callable Libraries of  
422 CPLEX 12.2. Experiments have been performed on a Linux four-CPU server  
423 2.67 GHz 64 GB RAM, with default parameter settings.

424 Although a major requirement for the carrier is to determine solutions in  
425 about 10 minutes, the solver execution has also been set to stop after 3 hours.  
426 This choice allows the solver to produce better upper and lower bounds and pro-  
427 vide a better term of comparison for assessing the quality of the meta-heuristic.

428 We set  $MAXTIME$  to 600 seconds, as suggested by the carrier,  $MAXIT =$   
429  $10000$ ,  $\gamma = |I|$  for penalties involving importers and  $\gamma = |E|$  for penalties  
430 involving exporters. Finally, the coefficient  $\alpha$  in Equation (23) takes value 10.  
431 These settings proved to provide good quality results in our preliminary testing.



(a) Number of best known solutions returned by the each calibration and the constructive heuristic. (b) Improvements with respect to the constructive heuristic.

432 The meta-heuristic depends also on the parameter  $\varphi$ , which sets the strategy  
 433 to update penalties: it takes value 1 for “Unchecked penalties”, 2 for “Unique  
 434 penalties” and 3 for “Incremental unique penalties” (see Section 4.4). All pe-  
 435 nalization strategies are tested and combined with several execution sequences  
 436 of the guidance mechanisms. The top five calibrations in our preliminary ex-  
 437 periments are denoted by  $C_1$ ,  $C_2$ ,  $C_3$ ,  $C_4$  and  $C_5$ , and are described hereafter:

438  $C_1$  each adaptive guidance mechanism has probability 33% to be performed at  
 439 each iteration. Penalties are updated according to the strategy “Unchecked  
 440 penalties”;

441  $C_2$  all adaptive guidance mechanisms are performed at each iteration. Penalties  
 442 are updated according to the strategy “Unchecked penalties”;

443  $C_3$  the *AGM1* is the only adaptive guidance mechanism and it is performed at  
 444 each iteration. Penalties are updated according to the strategy “Unchecked  
 445 penalties”;

446  $C_4$  each adaptive guidance mechanism has probability 33% to be performed at  
 447 each iteration. Penalties are updated according to the strategy “Incre-  
 448 mental unique penalties”;

449  $C_5$  each adaptive guidance mechanism has probability 25% to be performed at  
 450 each iteration. Penalties are updated according to the strategy “Incre-  
 451 mental unique penalties”.

452 In order to select the best calibration among them, all generated instances  
 453 are solved with each setting of the meta-heuristic. Since 33 instances out of  
 454 140 are proven to be optimal by Cplex, we consider the remaining 107 instances  
 455 and we compute how many times the best solution is found by each setting of  
 456 the meta-heuristic and by the constructive heuristic. Results are represented in  
 457 Figure 1a.

458 Figure 1a shows that calibration  $C_1$  seems to be the most effective, in fact  
 459 it determines the best solution for 87 times out of 107 instances. Figure 1a also

460 shows that in 55 instances the constructive heuristic returns the best solution  
461 and no improvement is obtained by any proposed guidance mechanism.

462 Figure 1b shows how many times each setting of the metaheuristic improves  
463 the solution of the SVRPCB determined by the constructive heuristic. For  
464 example, calibration  $C_1$  improves the initial feasible solution of the SVRPCB  
465 for 48 times and calibration  $C_2$  for 44 times.

466 As Figures 1a and 1b show,  $C_1$  seems to be the most promising calibration.  
467 Therefore, the results obtained by this calibration are discussed hereafter.

### 468 5.2. Effectiveness of the adaptive guidance mechanisms

469 This section illustrates the improvements produced by the adaptive guidance  
470 mechanisms running under the calibration  $C_1$  with respect to the constructive  
471 heuristic solution.

472 In Table 2 each row describes a class of several instances.  $Q$  denotes the  
473 transportation capacity of the homogeneous fleet of trucks and Instances the  
474 number of instances in the considered class. Table 2 reports in the column  
475 *CONSTRUCTIVE HEURISTIC* Average t(s), which is the average time  
476 to determine the first feasible solution by the constructive heuristic. The column  
477 *ADAPTIVE GUIDANCE* indicates the average time in seconds to find the  
478 best feasible solution for the meta-heuristic (Average t(s)) and the average gap  
479 between the solution of the meta-heuristic and the solution of the constructive  
480 heuristic (Average % Gap). Negative values of this gap indicate the average  
481 improvement produced by guidance mechanisms on the class of instances con-  
482 sidered in that row.

483 Table 2 shows that interesting improvement opportunities can be obtained  
484 by the guidance mechanisms. Moreover, they seem to be more effective as the  
485 truck capacity increases.

### 486 5.3. Comparison with Cplex

487 This section compares solutions provided by the meta-heuristic with those  
488 obtained by state-of-art solver *CPLEX*. Computational results are reported in  
489 Table 3 following the additional notation:

- 490 • *Average % Gap 10 min*: Average percentage gaps with respect to best  
491 solutions provided by CPLEX in 10 minutes. When the solutions of the  
492 meta-heuristic are better than the best CPLEX upper bounds, or the  
493 meta-heuristic provides the optimal solutions, gaps are reported in bold.
- 494 • *Average % Gap 3 h*: Average percentage gaps with respect to best solu-  
495 tions provided by CPLEX in 3 hours. When solutions of the meta-heuristic  
496 are better than CPLEX upper bounds, or the meta-heuristic provides op-  
497 timal solutions, gaps are reported in bold.
- 498 • *n.a.*: No available gap with respect to CPLEX within its time limit, be-  
499 cause CPLEX did not find any feasible solution.

500  
501 *CPLEX 10 min and 3h*



Q	Instances	CONSTRUCTIVE HEURISTIC		ADAPTIVE GUIDANCE	
		Average t(s)	Average t(s)	Average % Gap	
10 CUSTOMERS					
1	3		0.23	0.23	0.00
2	3		0.18	0.18	0.00
4	3		5.19	5.19	0.00
6	3		4.22	74.04	0.00
20 CUSTOMERS					
1	5		1.97	1.97	0.00
2	5		1.11	1.11	0.00
4	5		6.96	173.79	<b>-0.57</b>
6	5		8.70	96.09	<b>-1.70</b>
30 CUSTOMERS					
1	7		7.57	7.57	0.00
2	7		8.63	41.68	<b>-0.36</b>
4	7		13.21	408.39	<b>-1.98</b>
6	7		13.77	73.67	<b>-1.19</b>
40 CUSTOMERS					
1	9		23.50	23.50	0.00
2	9		23.33	188.15	<b>-0.31</b>
4	9		12.39	195.06	<b>-1.02</b>
6	9		17.26	228.82	<b>-1.92</b>
50 CUSTOMERS					
1	11		31.69	31.69	0.00
2	11		23.28	48.19	<b>-0.04</b>
4	11		16.31	131.61	<b>-0.28</b>
6	11		19.38	198.04	<b>-0.26</b>

Table 2: Adaptive guidance effectiveness

- 502 • *Optimality / Feasibility*: The first number indicates the number of op-  
503 timal solutions obtained in that class of instances; the second number  
504 indicates the number of feasible solutions for which the optimality cannot  
505 be demonstrated;
- 506 • *Average Opt. Gap*: The optimality gap between upper and lower bounds  
507 determined by CPLEX in 10 minutes and 3h, respectively;
- 508 • *n.s.*: No feasible solution determined by CPLEX within the time limit.

509 It is important to note that each row of Table 3 represents average percentage  
510 gaps over a class of instances.

511 Table 3 shows that the meta-heuristic provides exact solutions in instances  
512 where the transportation capacity is 1 container. CPLEX outperforms the meta-  
513 heuristic in few small instances; when  $n = 10$  and  $Q = 6$ , there are exact  
514 solutions at most 1.62% better than those determined by the meta-heuristic.

515 In the instances with 20 customers, CPLEX outperforms the meta-heuristic  
516 only when it is executed for 3h: the gaps are 0.12% and 0.41% for  $Q$  equal 2 and  
517 6, respectively. Nevertheless, the solutions obtained by CPLEX in 10 minutes  
518 are up to 12.19% worse on average than those of the meta-heuristic.

519 In case of instances with 30 customers, the meta-heuristic outperforms sys-  
520 tematically CPLEX, both when it is executed for 10min and 3h. The average

Q	Instances	ADAPTIVE GUIDANCE			CPLEX 10 min		CPLEX 3 h	
		Average t(s)	Average % Gap 10 min	Average % Gap 3 h	Optimality / Feasibility	Average %Opt. Gap	Optimality / Feasibility	Average % Opt. Gap
10 CUSTOMERS								
1	3	0.23	0.00	0.00	3 / 0	0.00	3 / 0	0.00
2	3	0.18	0.00	0.00	1 / 2	3.18	1 / 2	2.33
4	3	5.19	0.00	0.00	1 / 2	1.45	3 / 0	0.00
6	3	74.04	1.62	1.62	3 / 0	0.00	3 / 0	0.00
20 CUSTOMERS								
1	5	1.97	0.00	0.00	5 / 0	0.00	5 / 0	0.00
2	5	1.11	<b>-0.49</b>	0.12	0 / 5	5.20	0 / 5	4.46
4	5	173.79	<b>-4.73</b>	<b>-0.30</b>	0 / 5	16.27	0 / 5	10.97
6	5	96.09	<b>-12.19</b>	0.41	0 / 5	21.22	0 / 5	7.63
30 CUSTOMERS								
1	7	7.57	0.00	0.00	7 / 0	0.00	7 / 0	0.00
2	7	41.68	<b>-10.91</b>	<b>-1.42</b>	0 / 5	15.70	0 / 7	6.14
4	7	408.39	<b>-29.10</b>	<b>-15.28</b>	0 / 7	37.51	0 / 7	25.33
6	7	73.67	<b>-42.78</b>	<b>-24.20</b>	0 / 5	48.86	0 / 7	33.62
40 CUSTOMERS								
1	9	23.50	0.00	0.00	2 / 0	0.00	9 / 0	0.00
2	9	188.15	n.a.	n.a.	0 / 0	n.s.	0 / 0	n.s.
4	9	195.06	<b>-31.51</b>	<b>-24.82</b>	0 / 4	36.94	0 / 5	30.28
6	9	228.82	<b>-54.78</b>	<b>-49.95</b>	0 / 1	62.71	0 / 6	57.68
50 CUSTOMERS								
1	11	31.69	n.a.	0.00	0 / 0	n.s.	9 / 0	0.00
2	11	48.19	n.a.	n.a.	0 / 0	n.s.	0 / 0	n.s.
4	11	131.61	n.a.	n.a.	0 / 0	n.s.	0 / 0	n.s.
6	11	198.04	n.a.	n.a.	0 / 0	n.s.	0 / 0	n.s.

Table 3: Comparison with the exact algorithm.

521 gaps are up to 42.78% and 24.20% for 10min and 3h, when  $Q = 6$ . A simi-  
522 lar trend of improvement can be observed when  $n = 40$ , even if there are few  
523 instances where CPLEX was capable of generating an upper bound and, thus,  
524 the direct comparison of the methods is less significant. When  $n = 50$ , CPLEX  
525 cannot determine any upper bound in 10 minutes and returns only 9 upper  
526 bounds out of 44 instances in 3h.

527 Tests show that the meta-heuristic improves most of the upper bounds pro-  
528 duced by the exact algorithm, when the instance size is larger than 20-30 cus-  
529 tomers. Moreover, CPLEX is not able to provide feasible solutions for 77 out  
530 of 140 instances within a time limit of 10 minutes, and 51 out of 140 instances  
531 within a time limit of 3 hours. From the point of view of the execution time,  
532 the meta-heuristic provides all feasible solutions in less than 10 minutes.

533 Finally, Figure 1 analyses how larger capacities remarkably decrease the  
534 routing cost of the distribution. As Figure 1 shows, whenever the trucks have a  
535 larger capacity, the distribution is performed at a lower cost:

- 536 • If we consider the instances with capacity  $Q = 2$  with respect to the  
537 instances with capacity  $Q = 1$ , the routing cost decreases by 47.05% in  
538 the case of 20 customers, up to 58.22% in the case of 10 customers.
- 539 • If we consider the instances with capacity  $Q = 4$  with respect to the

540 instances with capacity  $Q = 2$ , the routing cost decreases by 38.72% in  
 541 the case of 10 customers, up to 46.06% in the case of 40 customers.

- 542 • If we consider the instances with capacity  $Q = 6$  with respect to the  
 543 instances with vehicles  $Q = 4$ , the routing cost decreases by 20.01% in the  
 544 case of 10 customers, up to 26.94% in the case of 20 customers.

545 Note that the marginal improvement due to the vehicles with capacity  $Q = 6$   
 546 with respect to trucks with capacity  $Q = 4$  is relatively small, but if we consider  
 547 the instances with vehicles capacity  $Q = 6$ , with respect to the instances with  
 548 vehicles capacity  $Q = 1$ , the routing cost decrease by 77.99% in the case of 20  
 549 customers and up to 79.52% in the case of 10 customers.

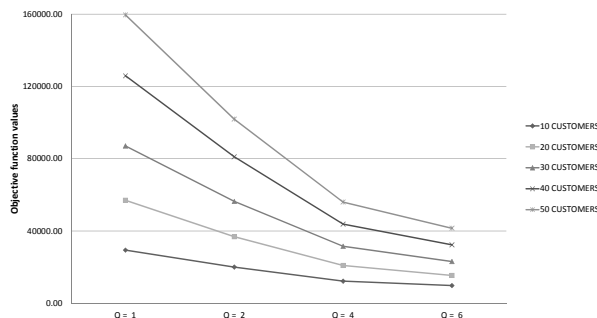


Figure 1: Efficiency of the distribution with larger transportation capacities

## 550 6. Conclusions

551 This paper addressed the SVRPCB, which is rich vehicle routing problem  
 552 originating from a real world application. Although there are many papers on  
 553 VRPCB and SVRP, to our knowledge, their integration was seldom investigated.  
 554 In the specific field of container transportation, this is an interesting variant of  
 555 drayage problems, due to the coupling between containers and trucks, each of  
 556 which can carry more than one container. In this paper we have presented a  
 557 mathematical model for the SVRPCB.

558 The proposed solution method is a meta-heuristic based on adaptive guid-  
 559 ance mechanisms. It determines feasible solutions for SVRPCB by a construc-  
 560 tive heuristic decomposing the problem into two simpler SVRPs, each solved by  
 561 a TS, and exactly merging routes by an assignment problem. However, these  
 562 feasible solutions may be inefficient, since too many splits may be performed,

563 highly expensive arcs may be used and the first or the last importer and/or  
564 exporter in any route may not be appropriate.

565 The proposed meta-heuristic aims to improve these solutions by detecting  
566 predefined drawbacks and guiding the TS in the SVRPs, in order to produce  
567 the desired diversification in SVRPCB solutions. More precisely, four guidance  
568 mechanisms are implemented by perturbing in the subsequent iterations the  
569 costs of the SVRPs, in order reduce splits, use less expensive arcs and change  
570 the first and/or the last customer in current routes.

571 Our experimentation indicates that some guidance mechanisms are more  
572 effective than others, but usually they are all able to improve initial feasible  
573 solutions. In our experimentation the most effective guidance mechanism is  
574 obtained when all proposed guidance mechanisms are randomly combined and  
575 arcs already perturbed can be penalized further. Moreover, the meta-heuristic  
576 is much more effective than a state-of-art solver in solving artificial instances  
577 with 20 and 30 customers, yielding considerable savings in terms of travelled  
578 distances. Therefore, the meta-heuristic represents a promising instrument to  
579 improve the decision-making process and provides a quantitative estimation of  
580 savings obtainable by increasing transportation capacities.

581 To conclude, the adaptive guidance mechanism is a general approach, which  
582 is based on the iterative analysis of current solutions and perturbation of simpler  
583 subproblems by problem-specific adaptive guidance mechanisms. It is important  
584 to note that this approach can exploit existing heuristics for subproblems of the  
585 problem at hand. Hence, one may easily adapt modules of code already in use,  
586 minimizing the inconvenience of adopting a new software. Easy pieces of code  
587 are also easier to maintain and possibly adapt to incorporate more advanced  
588 problem features. Further research will be carried out to implement guidance  
589 mechanisms on rich vehicle routing problems.

## 590 **7. Acknowledgements**

591 This work was partially supported by Grendi Trasporti Marittimi. The  
592 authors are also grateful to Gunes Erdogan for his careful comments on previous  
593 versions of the paper and Teodor Gabriel Crainic for the enlightening talks on  
594 this topic.

## 595 **References:**

- 596 Anily, S., 1996. The vehicle-routing problem with delivery and back-haul op-  
597 tions. *Naval Research Logistics* 43 (3), 415–434.
- 598 Archetti, C., Speranza, M. G., 2012. Vehicle routing problems with split deliv-  
599 eries. *International Transactions in Operational Research* 19 (1-2), 3–22.
- 600 Archetti, C., Speranza, M. G., Hertz, A., 2006. A tabu search algorithm for the  
601 split delivery vehicle routing problem. *Transportation science* 40 (1), 64–73.

- 602 Bai, R., Burke, E. K., Gendreau, M., Kendall, G., 2007. A simulated annealing  
603 hyper-heuristic: Adaptive heuristic selection for different vehicle routing prob-  
604 lems. In: The 3rd Multidisciplinary International Conference on Scheduling:  
605 Theory and Applications, MISTA 2007, Paris, France.
- 606 Battarra, M., Monaci, M., Vigo, D., 2009. An adaptive guidance approach for  
607 the heuristic solution of a minimum multiple trip vehicle routing problem.  
608 *Computers & Operations Research* 36 (11), 3041–3050.
- 609 Berbeglia, G., Cordeau, J., Gribkovskaia, I., Laporte, G., 2007. Static pickup  
610 and delivery problems: a classification scheme and survey. *Top* 15 (1), 1–31.
- 611 Braekers, K., Caris, A., Janssens, G., 2013. Integrated planning of loaded and  
612 empty container movements. *OR Spectrum* 35 (2), 457–478.
- 613 Brandão, J., 2006. A new tabu search algorithm for the vehicle routing problem  
614 with backhauls. *European Journal of Operational Research* 173, 540–555.
- 615 Caris, A., Janssens, G., 2009. A local search heuristic for the pre- and end-  
616 haulage of intermodal container terminals. *Computers and Operations Re-*  
617 *search* 36 (10), 2763–2772.
- 618 Cheung, R., Shi, N., Powell, W., Simao, H., 2008. An attributed decision model  
619 for cross-border drayage problem. *Transportation Research Part E: Logistics*  
620 *and Transportation Review* 44 (2), 217–234.
- 621 Chung, K., Ko, C., Shin, J., Hwang, H., Kim, K., 2007. Development of math-  
622 ematical models for the container road transportation in Korean trucking in-  
623 dustries. *Computers & Industrial Engineering* 53 (2), 252–262.
- 624 Clarke, G., Wright, J. W., 1964. Scheduling of vehicles from a central depot to  
625 a number of delivery points. *Operations Research* 12 (4), 568–581.
- 626 Gendreau, M., Hertz, A., Laporte, G., 1992. New insertion and postoptimization  
627 procedures for the traveling salesman problem. *Operations Research* 40, 1086–  
628 1094.
- 629 Glover, F., Laguna, M., 1998. *Tabu Search*. Springer.
- 630 Goetschalckx, M., Jacobs-Blecha, C., 1989. The vehicle routing problem with  
631 backhauls. *European Journal of Operational Research* 42 (1), 39–51.
- 632 Gribkovskaia, I., Laporte, G., 2008. One-to-many-to-one single vehicle pickup  
633 and delivery problems. In: Golden, B., Raghavan, S., Wasil, E., Sharda, R.,  
634 Voss, S. (Eds.), *The Vehicle Routing Problem: Latest Advances and New*  
635 *Challenges*. Vol. 43. Springer, Berlin, pp. 359–377.
- 636 Hart, S., 2005. Adaptive heuristics. *Econometrica* 73 (5), 1401–1430.

- 637 Imai, A., Nishimura, E., Current, J., 2007. A lagrangian relaxation-based heuristic  
638 for the vehicle routing with full container load. *European journal of Op-*  
639 *erational Research* 176 (1), 87–105.
- 640 Jula, H., Dessouky, M., Ioannou, P., A., C., 2005. Container movement by  
641 trucks in metropolitan networks: modeling and optimization. *Transporta-*  
642 *tionnResearch Part E: Logistics and Transportation Review* 41 (3), 235–259.
- 643 Kramer, O., 2008. *Self-adaptive heuristics for evolutionary computation.*  
644 Springer.
- 645 Lai, M., Crainic, T. G., Di Francesco, M., Zuddas, P., 2013. An heuristic search  
646 for the routing of heterogeneous trucks with single and double container loads.  
647 *Transportation Research Part E: Logistics and Transportation Review* 56,  
648 108–118.
- 649 Macharis, C., Bontekoning, Y., 2004. Opportunities for or in intermodal freight  
650 transportresearch: A review. *European journal of Operational Research*  
651 153 (2), 400–416.
- 652 Mingozzi, A., Giorgi, S., R., B., 1999. An exact method for the vehicle routing  
653 problem with backhauls. *Transportation Science* 33 (3), 315–329.
- 654 Mitra, S., 2005. An algorithm for the generalized vehicle routing problem with  
655 backhauling. *Asia-Pacific journal of Operational Research* 22 (2), 153–169.
- 656 Mitra, S., 2008. A parallel clustering technique for the vehicle routing problem  
657 with split deliveries and pickups. *Journal of the Operational Research Society*  
658 59 (11), 1532–1546.
- 659 Nagl, P., 2007. Longer combination vehicles (lcv) for asia and pacific region:  
660 Some economic implications.  
661 URL [http://www.unescap.org/pdd/publications/workingpaper/wp\\_07\\_](http://www.unescap.org/pdd/publications/workingpaper/wp_07_02.pdf)  
662 [02.pdf](http://www.unescap.org/pdd/publications/workingpaper/wp_07_02.pdf)
- 663 Namboothiri, R., Erera, A., 2008. Planning local container drayage operations  
664 given a port access appointment system. *Transportation Research Part E:*  
665 *Logistics and Transportation Review* 44 (2), 185–202.
- 666 Nossack, J., Pesch, E., 2013. A truck scheduling problem arising in intermodal  
667 container transportation. *European Journal of Operational Research* 230 (3),  
668 666–680.
- 669 Olivera, A., Viera, O., 2007. Adaptive memory programming for the vehi-  
670 cle routing problem with multiple trips. *Computers & Operations Research*  
671 34 (1), 2847.
- 672 Osman, I., Wassan, N., 2002. A reactive tabu search meta-heuristic for the  
673 vehicle routing problem with back-hauls. *Journal of Scheduling* 5, 263–285.

- 674 Parragh, S. N., Doerner, K. F., Hartl, R. F., 2008. A survey on pickup and deliv-  
675 ery problems. part i: Transportation between customers and depot. *Journal*  
676 *fur Betriebswirtschaft* 58 (1), 21–51.
- 677 Ropke, S., Pisinger, D., 2006. A unified heuristic for a large class of vehicle  
678 routing problems with backhauls. *European Journal of Operational Research*  
679 171 (3), 750–775.
- 680 Sterzik, S., Kopfer, H., 2013. A tabu search heuristic for the inland container  
681 transportation problem. *Computers & Operations Research* 40 (4), 953–962.
- 682 Toth, P., Vigo, D., 1997. An exact algorithm for the vehicle routing problem  
683 with backhauls. *Transportation Science* 31 (4), 372–385.
- 684 Toth, P., Vigo, D., 1999. A heuristic algorithm for the symmetric and asymmet-  
685 ric vehicle routing problem with backhauls. *European Journal of Operational*  
686 *Research* 113 (3), 528–543.
- 687 Toth, P., Vigo, D., 2002. VRP with backhauls. In: Toth, P., Vigo, D. (Eds.),  
688 *The Vehicle Routing Problem. SIAM Monographs on Discrete Mathematics*  
689 *and Applications*, Philadelphia, PA, pp. 195 – 224.
- 690 Vidal, T., Crainic, T., Gendreau, M., Prins, C., 2012. A unified solution frame-  
691 work for multi-attribute vehicle routing problems. Tech. Rep. CIRRELT 2012-  
692 23, University of Montreal, Canada.
- 693 Vidovic, M., Radivojevic, G., Rakovic, B., 2011. Vehicle routing in containers  
694 pickup up and delivery processes. *Procedia-Social and Behavioral Sciences* 20,  
695 335–343.
- 696 Zachariadis, E., Kiranoudis, C., 2012. An effective local search approach for the  
697 vehicle routing problem with backhauls. *Expert Systems with Applications*  
698 39, 3174–3184.
- 699 Zhang, R., Yun, W., Kopfer, H., 2010. Heuristic-based truck scheduling for  
700 inland container transportation. *OR Spectrum* 32 (3), 787–808.
- 701 Zhang, R., Yun, W., Moon, I., 2011. Modeling and optimization of a container  
702 drayage problem with resource constraints. *International journal of Produc-*  
703 *tion Economics* 133 (1), 351–359.