Università di Pisa

Dipartimento di Informatica

Technical Report

A multiperiod drayage problem with customer-dependent service periods

Ali Ghezelsoflu
Dipartimento di Informatica, Università di Pisa
ali.ghezelsoflu@di.unipi.it

Massimo Di Francesco
Department of Mathematics and Computer Science, University of Cagliari
mdifrance@unica.it

Antonio Frangioni
Dipartimento di Informatica, Università di Pisa
frangio@di.unipi.it

Paola Zuddas
Department of Mathematics and Computer Science, University of Cagliari
zuddas@unica.it

April 30, 2019

ADDRESS: Largo B. Pontecorvo 3, 56127 Pisa, Italy. TEL: +39 050 2212700 FAX: +39 050 2212726

A multiperiod drayage problem with customer-dependent service periods

Ali Ghezelsoflu
Dipartimento di Informatica, Università di Pisa
ali.ghezelsoflu@di.unipi.it

Massimo Di Francesco
Department of Mathematics and Computer Science, University of Cagliari
mdifrance@unica.it

Antonio Frangioni
Dipartimento di Informatica, Università di Pisa
frangio@di.unipi.it

Paola Zuddas
Department of Mathematics and Computer Science, University of Cagliari
zuddas@unica.it

April 30, 2019

Abstract

We investigate a routing problem arising in the domain of drayage operations. to determine mimimum-cost vehicle routes in several periods. We adapt a set-covering model, which is solved either with all feasible routes by an off-the-shelf MIP solver, or by and a Price-and-Branch algorithm in which the pricing problem is a formulated as a collection of shortest path problems in tailor-made auxiliary acyclic networks. We propose a new arc-flow formulation based on the previous auxiliary networks and show that solving it by a MIP solver is usually preferable. Finally, we characterize how possible changes in flexibility levels affect routing costs.

Keywords: *Drayage, Modeling, Reformulations*

1 Introduction

Although drayage operations represent a small fraction of the total distance of an intermodal shipment, they account for a substantial share of the overall shipping costs. As a result they have been subject to a significant amount of research [1]. Yet, of that little has focused on improving their efficiency over different time periods. This paper contributes to this research area by the proposing and comparing different formulations for a multiperiod drayage problem, which is motivated by the case of a carrier dealing with the distribution of container loads based at a port by trucks.

The carrier must meet two types of transportation requests: the delivery of container loads from the port to importers, and the shipment of container loads from exporters to the same port. Drivers wait for containers during packing and unpacking operations and trucks carry the same containers before and after the customer service, whereas in many papers containers and tractors are supposed to be decoupled during customer service [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]. In this problem, routes are performed by two types of trucks, which are based at the port and carry one or two containers at a time, respectively, whereas most of the papers on drayage ignore two-container trucks (few exceptions are [16, 17, 18]). Two-container trucks have slightly larger routing costs per unitary distance as opposed to one-container-trucks, but can serve twice the demand. Typically customers must be serviced by more than one container load. According to the current carrier policy, all importers are served before all exports in any route. Although it may possible for two-container trucks to service an importer after an exporter, these routes are quite unlikely to be part of the optimal solution and result in negligible savings, if any [1].

In the practical setting motivating our problem, the carrier moves container to be loaded and unloaded from a vessel calling at the port every three days. As a result, most of the transportation requests are known in advance. Each request specifies where customers are located, how many container loads must be delivered or collected, and when they can be services in the three-day planning horizon. Generally speaking, customers can be visited over multiple days (or periods) according to their flexibility. In our experiments we will consider three flexibility levels: *no-flexibility customers* must be serviced in a predefined day, *medium-flexibility customers* must be serviced in two days out of three of the planning horizon, and *high-flexibility customers* can be serviced in any day. The objective is to minimize routing costs, such that all customers are serviced as requested according to their flexibility and truck capacity constraints hold. To our knowledge, this problem setting has not yet been investigated in the literature on drayage, which typically focuses on single-period problems. Although in our experiments we restrict to the three-days instances motivating our research, our models immediately extend to periods of time of different length.

This paper provides different contributions to the problem:

1. Since up to four customers can be visited along each route in each period, a natural way to model the problem is to enumerate all feasible routes in a network, where nodes represent the port and customers in each period they can be serviced and arcs indicate the possible connections among them. The Set Covering (SC) formulation of [1] for the single-period problem is extended to the case of multiple periods and solved with all feasible routes by an off-the-shelf MIP solver.

2. Since the number of routes may be too large, the Linear Programming relaxation of the SC formulation is solved by Column Generation, where the pricing subproblem is a collection of acyclic shortest path problems on appropriate auxiliary networks, extending to the multi-period setting the approach by [19]. From the set of generated columns, an integer solution is sought by a MIP solver under the Price-and-Branch paradigm.

3. An innovative arc-flow formulation based on a slight modification of the previous auxiliary network is proposed and solved very effectively by a MIP solver.

These three approaches are extensively tested on several instances, in order to point out to what extent they can be adopted. The experiments show that the arc-flow formulation is by far the best performing, not only among the three but also against the closest formulation in the field, which was proposed by [20] for single-period problems. The effectiveness of the formulation allows us to thoroughly explore the connection between flexibility and routing costs, i.e., which savings can be obtained if customers accept extending the number of periods in which the transportation service is provided. This allows, for instance, the carrier to introduce discount fees in exchange of extended flexibility [21]. To summarize, the objectives of this paper are to:

- investigate a new setting of a routing problem in the field of drayage operations with two vehicle types and customer-dependent service times over multiple periods;

- propose different formulations this problem: we adapt to the case of multiple periods a SC formulation and a Price-and-Branch algorithm, and we introduce a new arc-flow model;

- compare the different approaches to the problem onto several instances and show how possible changes in the flexibility of customers affect routing costs.

This paper is organized as follows. In Section 2, a brief review of the related literature is provided. In Section 3, the Set Covering model and the Price-and-Branch algorithm are presented. In Section 4 the arc-flow formulation is proposed. In Section 5, the results of our extensive computational experience are shown. Moreover, we shed light on the connection between customer priorities and routing costs. Conclusions and future research perspectives are drawn in Section 6. The single-period model of [20] is reported in A, and the comparison on single-day problems is shown in B.

2 Related Literature

This problem is in the domain of the Vehicle Routing Problem (VRP), which is widely studied owing to its practical relevance and its considerable difficulty [22]. Three variants of routing problems deal address the service to customers over different periods: the Multiperiod Routing Problem (MRP), the Periodic Vehicle Routing Problem (PVRP) and the Inventory Routing Problem (IRP).

In MRP, each request specifies a demand quantity, a customer location and a set of consecutive periods during which the transportation service can take place. The carrier must plan its delivery routes over several days so as to minimize the routing cost, as well as possible additional costs, such as customer waiting, inventory and penalties for services postponed beyond due dates [23, 24]. Some flexibility in the dates of delivery is considered in [25]. In the MRP with service choice [26], the service frequency is a model decision variable, whereas in our problem it is a customer requirement. Our problem is clearly in the field of MRP, but no paper in this field accounts for all the features of this problem.

In PVRP customers are supposed to be served according a certain frequency and receive a fixed quantity at each visit [27], whereas in our problem frequencies are not given beforehand and the assumption on the fixed quantity delivered (or collected) at each visit holds only for customers with no flexibility. The PVRP was introduced by Beltrami and Bodin [28], who proposed two approaches to the problem: (i) building routes and assigning them to days; (ii) assigning customers to days and solving routing problems independently for each day. Relevant approaches for the PVRP were proposed by [29, 30, 31, 32, 33, 34].

In IRP [35] the daily consumption of the customers is given and the quantities to be delivered, day by day, have to be decided. One has to determine the routes such that the capacity of vehicles holds and each customer in each day has the quantity sufficient for the daily consumption. In our problem, consumption rates are not known in advance and customer choose the service times, thus the IRP is not appropriate. Recent papers oncrama2018stochastic IRP were proposed by [36], [37] and [38].

A closer problem setting is the Flexible Periodic Routing Problem (FPRP), which was introduced by [39]. In the FPRP, Each customer has a total demand that must be served within the horizon and a limit on the maximum quantity that can be delivered at each visit. The first characteristic holds only for customers with high flexibility in our problem, in which customers have service-dependent periods. A an iterative two-phase heuristic for the FPRP was proposed in [40].

To our knowledge, a drayage problem with multiple periods was investigated only in [41]. Nevertheless, their problem is different, as they considered several decision epochs during a day, only one-container trucks and uncertain requests.

3 Set-covering model and the Price-and-Branch algorithm

Consider a set H periods in the planning horizon. Let $G = (N, A)$ be the *physical* directed graph, in which $N = \{p\} \cup I \cup E$, i.e., nodes represent the port (p) and all possible customers $V = I \cup E$—be them importers (I) or exporters (E)—which will have to be serviced at any point of the planning horizon. Any arc $(i, j) \in A$ represents the direct truck trip between i and j, and has the two associated costs c^1_{ij} and c^2_{ij} for one- and two-containers trucks, respectively. Arc are directed both because costs, in this short-haul setting, may not be symmetric, e.g., due to one-way roads, and because importers and exporters are not equivalent in terms of the order in which they can be visited, as discussed in details later on. It is immediate to construct the (physical) sub-graphs $G^t_h = (N^t_h, A^t_h)$ of G for each period $h \in H$ and type of truck $t \in T = \{1, 2\}$. $N^t_h = \{p\} \cup V_h$, with $V_h = I_h \cup E_h$ and I_h, E_h denoting respectively the set of importers and exporters who accept to be serviced in period $h \in H$. Thus, the set of nodes does not depend on the truck type. It ought be immediately mentioned, however, that some customer may be both an importer and an exporter at some time period h. For the sake of this model, any such customer will be treated as a pair of distinct nodes, say $v \in I_h$ and $w \in E_h$ with $v \neq w$; just, the cost of the arcs (v, w) and (w, v) (if they are constructed) will be zero. The arc sets A^t_h are those induced by N_h that are feasible for the given truck type. In particular, since the demand of customers is in terms of container loads, two-container trucks ($t = 2$) can in principle move between any pair of customers, whereas one-container trucks ($t = 1$) can only go—besides from the port directly to a customer and back—from an importer to an exporter, but not vice-versa. Indeed, when the truck visits the exporter its only slot is filled up, and therefore it can then only go back to the port. Hence, $A^1_h \subseteq A^2_h$ for all h.

Figure 1 depicts one example of the physical sub-graphs when $H = \{1, 2, 3\}$, i.e., the planning horizon has three periods, each representing a day. In the example, G has the port p, two importers, represented by circular-shape nodes and denoted by 1 and 2, and three exporters, represented by square-shape nodes and denoted by 3, 4 and 5. Importer 1 and exporter 3 must be served in the first day (i.e., in our parlance, they have no flexibility), exporter 5 could be visited in day 1 or 2 (i.e., has medium flexibility), importer 2 and exporter 4 could be served in any day (i.e., they have high flexibility). The Figure illustrates the different arcs that could be traversed in the routes of one-container and two-container trucks. According to these arcs, all importers must be visited before all exporters, even if it is possible for two-container trucks to serve an importer after an exporter.

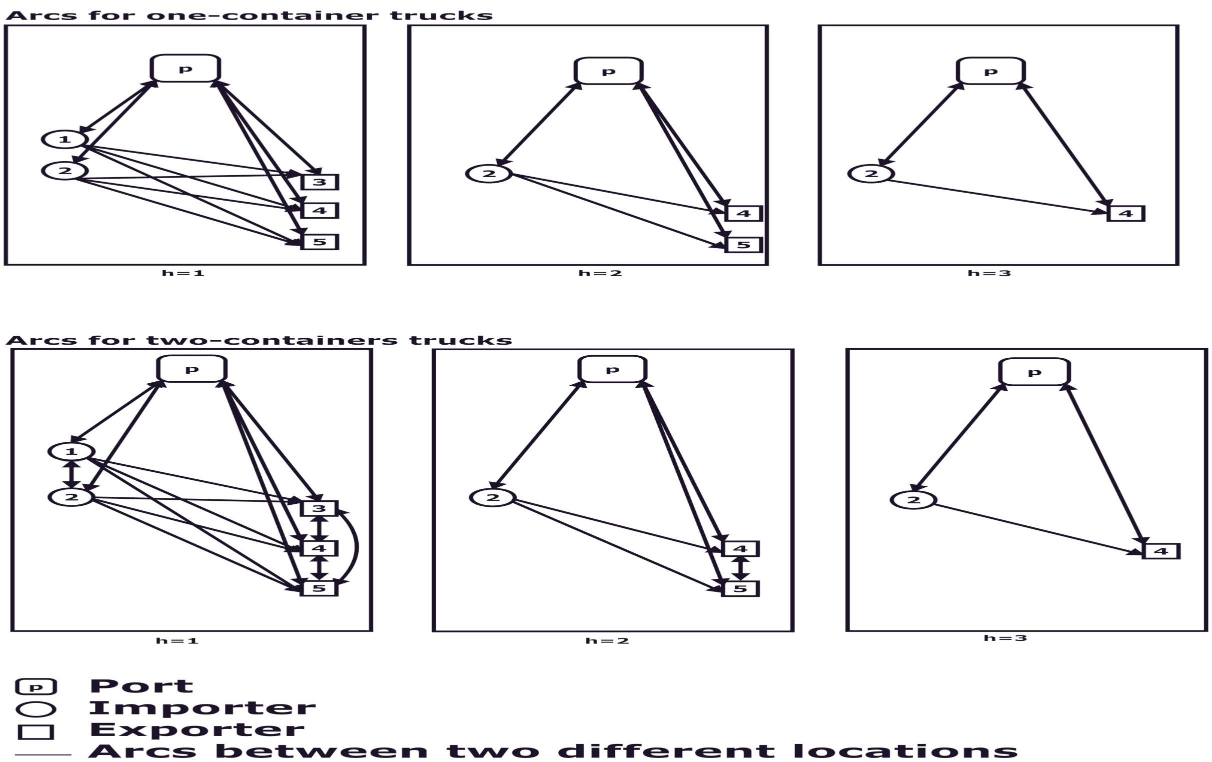

Figure 1: *Physical* sub-graph for the case of 3 periods

It is easy to construct an arc-flow mathematical model of the problem, at least for one single period, where variables are associated to arcs in the G^t_H physical graphs [42]. However, the continuous relaxation of such a formulation is usually rather weak, making it—despite the compact size—an inefficient choice to develop solution methods. Rather, following [1], path-based formulations are attractive, because, due to the restriction on the set of possible loading/unloading patterns, a truck cannot visit "many" nodes. Also, the continuous relaxation of the corresponding Set Covering-type formulations is known to usually provide much stronger lower bounds. This only requires defining the set $R(h)^t$ of all feasible routes for trucks of type t in period h. These are, on the outset, simple directed cycles in G^t_h starting and ending in p and visiting not more than t importers and exporters in the right order. Yet, it has to be remarked that, while visiting a given importer/exporter, a two-containers truck possibly has a choice, in that it can leave/collect either one container load or two. Of course, if it leaves/collects two, then this must be the only importer/exporter that is visited in the route. This information may be imagined encoded in the graph, and therefore in the routes, by allowing self-loops (i, i) among the arcs; each entry in a node (even from a self-loop) corresponds to the delivery/collection of a container load. However may the information be represented, we consider it embedded in the description of each route r, so that routes corresponding to the same cycle in G but having different patterns of loading/unloading operations will be distinct. Similarly, the same route (even with the same pattern of delivery/collection operations) for different periods will be considered two distinct routes. Hence, for each route r we will denote by $t(r)$ and $h(r)$ respectively the type of truck and the time period corresponding to it. We will then denote by $R(h) = R(h)^1 \cup R(h)^2$ the set of all feasible routes in period $h \in H$, and by $R = \cup_{h \in H} R(h)$ the set of all feasible routes over all periods. In addition, we will denote by k_h^t the number of trucks of type $t \in T$ which are available in period $h \in H$.

Constructing a SC formulation is now immediate. For each route $r \in R(h)$, let $\alpha_{v,r}$ be the coefficient which takes value 2 if customer $v \in N \setminus \{p\}$ is serviced by two container loads carried by a single (two-container) truck doing route $r \in R$ (then, clearly $r \in R(h)^2$ for some h), 1 if v is serviced by one container, and 0 if v is not visited in route r. Moreover, we let c_r be the cost of route $r \in R$, i.e., the sum of the costs $c_{ij}^{t(r)}$ of the corresponding directed cycle in G. Note that, in our data, servicing a customer twice with the same truck is considered to have 0 cost, i.e., there is no cost directly associated to loading/unloading operations; however, it would be easy to add this component to the cost of the route (e.g., by having it as the cost of self-loops). Finally, we denote by d_v the demand of customer $v \in V$, i.e., the number of containers which must be used to service customer v in the overall planning horizon. Letting x_r be the an integer decision variable representing the number of times in which route r is traversed, the following model readily ensues:

$$
\begin{array}{llll}
(SC) & \min & \sum_{r \in R} c_r x_r & (1) \\
 & s.t. & \sum_{r \in R} \alpha_{v,r} x_r \geq d_v \qquad v \in V & (2) \\
 & & \sum_{r \in R^t(h)} x_r \leq k_h^t \qquad t \in T \;,\; h \in H & (3) \\
 & & x_r \in \mathbb{N} \qquad r \in R & (4)
\end{array}
$$

The formulation is straightforward. Routing costs are minimized in the objective function (1). Constraints (2) ensure that all customers are served in the planning horizon. Constraints (3) enforce that the number of routes performed in each period is lower than the number of available trucks for each truck type. Finally, (4) define the domain of the decision variables. We will denote by $(\underline{SC})$ the continuous relaxation of (SC), simply obtained by replacing (4) with $x_r \geq 0$.

For a relatively small number of customers, which is reasonable in several practical applications, it is possible to just statically enumerate all possible routes. For instance, for $|I| = |E| = 20$, there are about 160000 routes per time period, which is still within the grasp of general-purpose MILP solvers, especially since the root node gap is usually a small fraction of 1%, if not downright 0, as expected for this type of formulation [1]. However, when the number of customers grow, the number of feasible routes quickly becomes rather large. In this case, $(\underline{SC})$ can be solved by a standard application of the well-known column generation technique. This starts by forming the *Restricted Master Problem* (RMP), which is $(\underline{SC})$ where the full set of routes R is replaced by a (much) smaller subset $\mathcal{R} \subset R$. One minor issue concerns the fact that the initial $\mathcal{R}$ should be chosen in a way that ensures that (RMP) does have a feasible (hence, an optimal) solution. This is in principle nontrivial, but in our applications the constraint on the truck number is not particularly tight and it is reasonably easy to find a feasible solution by a constructive heuristic which can be used to compute the initial $\mathcal{R}$; techniques ("phase 0") are, anyway, available to solve this issue in general. Once formed, the (RMP) is solved by any algorithm for Linear Programming, which necessarily at the same time solves its dual:

$$
\begin{array}{llll}
(DRMP) & \max & \sum_{v \in V} (\xi_v d_v - \sum_{h \in H} \sum_{t \in T} \pi_h^t k_h^t) & (5) \\
 & s.t. & \sum_{v \in V} \xi_v \alpha_{v,r} - \pi_{h(r)}^{t(r)} \leq c_r \qquad r \in \mathcal{R} & (6) \\
 & & \xi_v \in \mathbb{R}_+ \qquad v \in V & (7) \\
 & & \pi_h^t \in \mathbb{R}_+ \qquad t \in T \;,\; h \in H & (8)
\end{array}
$$

The key of the approach is that the dual of $(\underline{SC})$ has constraints (6) corresponding to each $r \in R$, as opposed to only those for $r \in \mathcal{R}$ that are present in $(DRMP)$. Hence, one only have to prove that all the constraints corresponding to each $r \in R$ are satisfied by the current dual solution ξ^*, π^* of $(DRMP)$, which proves the optimality for the whole $(\underline{SC})$ of the corresponding optimal solution x^* of (RMP), or find any route $\bar{r}$ whose constraint is violated, which can then be added to $\mathcal{R}$ for the process to be iterated.

The crucial step is therefore to find a route $\bar{r}$ that violates the constraint. This can be restated as the fact that the *reduced cost* of $\bar{r}$, i.e., $c_{\bar{r}}^* = c_{\bar{r}} - \sum_{v \in V} \xi_v^* \alpha_{v,\bar{r}} - (\pi_{h(\bar{r})}^{t(\bar{r})})^*$ is negative. The fundamental observation is that, discarding the last term which is constant, the reduced cost of $\bar{r}$ can be computed as the sum of the reduced costs of the arcs of the cycle (comprised the self-loops), where the reduced cost of arc (i,j) is $c_{ij}^* = c_{ij} - \xi_i^*$. Thus, determining a route of negative reduced cost can be reduced to a collection of Shortest Path Problems (SPP) on tailor-made acyclic networks $\bar{G}_h^t = (\bar{N}_h^t, \bar{A}_h^t)$, one for each $t \in T$ and $h \in H$. The topology of these graphs, which we can call the *step-expanded networks*, is readily obtained by that of the corresponding $G_h^t = (N_h^t, A_h^t)$ by cleverly "'unrolling the self-loops", when permitted, for at most one time each.

More precisely, for $t = 1$, the port is split into two nodes p' and p'' while each customer (importer or exporter) $v \in V_h$ is modeled by one node exactly like in N_h^1. The arcs in $\bar{A}_h^1$ represent possible trips performed by one-container trucks, i.e., they are basically those of A_h^1 save that arcs leaving p now leave p' and arcs entering p now enter p''. Note, that, as in A_h^1; arcs between $v \in I_h$ and $w \in E_h$ are only oriented (v, w), and not (w, v). The rationale, again, is that in order to serve an importer, a one-container truck has to leave the port with its only slot loaded: it cannot, therefore, visit an exporter before reaching one importer to deliver a container load. Thus, for $t = 1$ the trips are naturally ordered by types of customer: importers always come first than exporters.

For $t = 2$, however, the construction is considerably more involved. Not only the port is again split into two nodes p' and p''; also each customer (importer or exporter) $v \in V_h$ is modeled by two distinct nodes v' and v'', each associated with a container used to service this customer. This means that, basically, the original node set $V_h = I_h \cup E_h$ is replicated twice as two distinct sets $V_h' = I_h' \cup E_h'$ and $V_h'' = I_h'' \cup E_h''$. The construction of the arcs then has to follow from the actual operating rules of the carrier. As already remarked, it is common to require that all importers are serviced at the beginning of the trip, i.e., before visiting any of the exporters. If this is the case, the arcs represent possible the legs performed by two-container trucks are:

- Arcs (p', v') for all node $v' \in I_h'$, representing trips where the truck starts loaded with at least one container, and for all $v' \in E_h'$ representing trips where the truck starts with at least one slot free.
- Arcs (v', w'') for all $v' \in I_h'$ and all $w'' \in I_h''$, as well as for all $v' \in E_h'$ and all $w'' \in E_h''$. These represent the fact that, after having serviced for the first time an importer/exporter, the truck can surely still serve another of the same type. Note that here are comprised the arcs (v', v'') meaning the the truck is actually servicing the same customer v with two containers (either delivering two container loads, which means it started from p fully loaded, or picking them which means starting from p empty).
- Arcs (v', w') and (v'', w') where $v', v'' \in I_h$ while \$$w' \in E_h'$. This means that it is always possible to visit an exporter after having visited one or two importers (possibly, the same importer twice), but it is never possible to get back.
- Finally, arcs (v', p'') and (v'', p'') for all $v' \in V_h'$ and $v'' \in V_h''$, representing the possibility of always terminating a trip just after having serviced any customer of either type.

An important property of $\bar{G}^h$ is that is clearly acyclic. This allows both to use efficient acyclic SPP algorithms to solve the pricing problem, and especially to be guaranteed that no negative cost cycle can ever form. In fact, the reduced costs c_{ij}^* of the arcs—where the original cost $c_{ij} \geq 0$ for each $(i, j) \in \bar{A}_h^t$ is easily deduced from the construction—can well become negative; were there cycles in the graph, the shortest path may not exist as the problem may be unbounded below.

Figure 2 depicts the acyclic graphs associated with the original graph in Figure 1. In the case of two-container trucks, dashed lines show the "no-travel" arcs connecting the pair of nodes $v' \in V'$ and $v'' \in V''$ associated with the same customer v.

The above construction relies on the assumption that importers always have to be visited before exporters. While this is common practice, and more flexible rules usually yield only minimal advantages [1], it may be appropriate to remove this limitation and allow all routes that are physically possible. In particular, let i', j'' and e', f'' be the (at most) two importer and exporter nodes, respectively, visited by the truck, with $i = j$ and $e = f$ possible. The graph $\bar{G}_h^2$ as constructed above contains all paths of the form $i' \to j'' \to e' \to f''$ and all their sub-paths, including those visiting only one type of customer (once or twice). However, other physically possible sequences of operations are $i' \to e' \to j'' (\to f'')$ and $e' \to j'' (\to f'')$, representing visiting an exporter (either as first or second leg) before visiting an importer (and then possibly another exporter). Note that in the second type of path no node i' is visited, but this is immaterial; the two nodes j' and j'' are not meant to distinguish the first and the second visit to importer j, as the notation may seem to suggest, but only to represent that there can be up to two, and no more than that, visits for the same truck.

Allowing the two extra types of paths above clearly requires adding arcs (e', j'') to $\bar{A}_h^2$ for $e' \in E_h$ and $j'' \in I_h$. This, however, immediately creates a cycle with the existing arcs (j'', e'). In order to allow

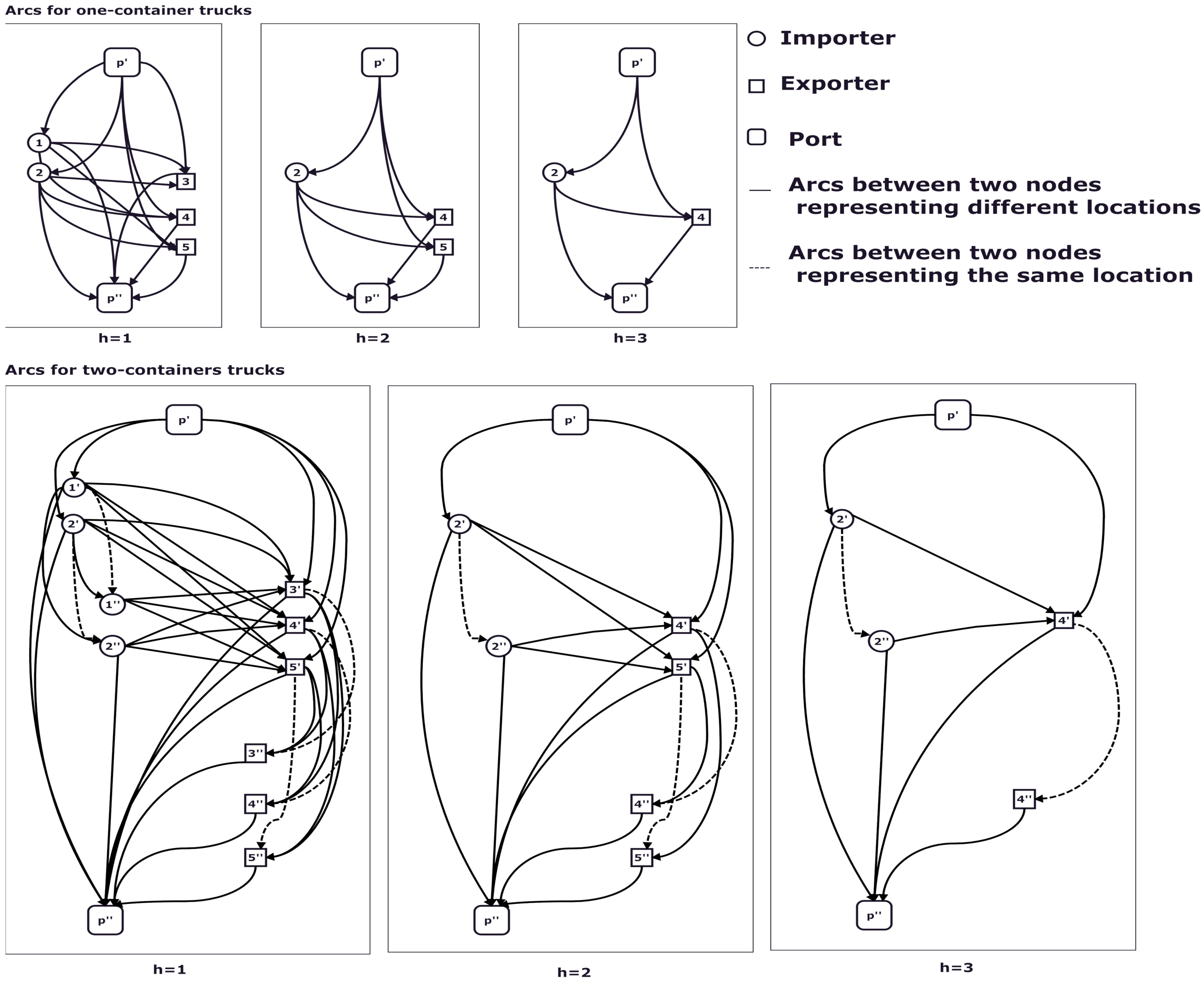

Figure 2: Acyclic networks for the example in Figure 1

these routes while keeping the graphs acyclic, it is therefore necessary to resort to having a dedicated subgraph, say $\tilde{G}^2_h = (\bar{N}^2_h, \tilde{A}^2_h)$ upon which independently solving an acyclic SPP. The graph is constructed analogously to $\bar{G}^2_h$, except that:

- Arcs (v'', w'') for $v'' \in I_h$ and $w'' \in E_h$ are included to $\tilde{A}^2_h$. These could actually have been introduced in $\bar{G}^t_h$ already, but they are useless there because, after having reached $w'' \in E_h$, it is always possible to go directly to p''. Since clearly the cost of (v'', w') and (v'', w'') is the same, routes visiting two importers and one exporter can always be taken to pass through a w' node, thus allowing to avoid inserting several arcs. This is no longer possible in $\tilde{G}^2_h$, as the route may already have passed via an exporter before having reached the second importer.

- Arcs (w', v'') for $v' \in I_h$ and $w'' \in E_h$ are included to $\tilde{A}^2_h$.

- Arcs (v'', w') where $v'' \in I_h$ and $w' \in E_h$ are *not* included to $\tilde{A}^2_h$, unlike in $\bar{A}^t_h$.

Clearly some of the routes feasible for $\bar{G}^2_h$ are no longer so in $\tilde{G}^2_h$, but this is compensated by the fact that the two kind of physically possible routes illustrated above are now represented, Thus, at the (likely, negligible) cost of solving one extra anyclic SPP per time period, column generation can be implemented that can construct all physically possible routes.

With the ability of efficiently solving the *pricing problem*(s) for ($\underline{SC}$), it is almost immediate to implement the column generation approach and solve it expeditiously, even as the number of customers grows significantly. While this ultimately provides the same bound as solving it in one blow, and this bound is usually quite good, it does not in general provide any integer feasible solution (although for several instances the continuous solution is in fact integer). However, the set of routes generated during the approach can be reasonably expected to contain "good" ones that can therefore be used to construct "good" integer solutions. This immediately suggests a *Price-and-Branch* (P&B) approach, whose implementation at least is straightforward when the column generation is available: simply, pass the final set of routes $\mathcal{R}$ of (RMP) to a general-purpose MILP solver and just solve the corresponding (small-ish) program to integer optimality. Customarily, the Linear Programming component of the "final" MILP solver is actually used to implement the column generation algorithm, so that the columns are already loaded in the MILP solver and implementing the P&B basically boils down to invoking a single function of the MILP solver API once the column generation is finished. Also, solution times of the MILP are usually quite low due to the afore-mentioned small gaps. Although more sophisticated approaches can be implemented, up to a full-blown *Branch-and-Price* (B&P) where the column generation is repeated at all nodes of the enumeration tree, the P&B can be expected to be quite effective and efficient when the root node gap of the formulation is low. This actually happens in [19].

4 Compact arc-flow formulation

A different possibility exists for exploiting the results of the previous paragraph in order to produce a "compact" formulation that shares the same strong bound as the (SC) one. The idea is, on the outset, simple: the pricing problem of the column generation approach, being a SPP, has a compact (flow-based) formulation. Basically, all that is needed is to use this to construct a formulation to the entire problem.

To do that, it is only necessary to minimally expand the step-expanded networks $\bar{G}^t_h$ of the previous paragraph by adding the single "return arc" (p'', p'). To alleviate the notation, we will assume that the arc was already present in $\bar{A}^t_h$ (as it well could have, as it would have no influence on the solution of the p'-p'' shortest path save for creating a cycle that can clearly be ignored). Owing to the presence of multiple networks $\bar{G}^t_h$, one for each $t \in T$ and $h \in H$, the formulation is, basically, an *integer Multicommodity Flow problem.* For each arc $(i, j) \in \bar{A}^t_h$, we define the integer arc-flow variable x^{th}_{ij} denoting the number of trucks of type t doing that particular leg (comprised the "no-travel arcs" (v', v'') for some customer $v \in V_h$) at time period h. With these variables, the $(SEAF)$ (Arc-Flow in the Step-Expanded networks) formulation of the problem is:

$$\min \sum_{h\in H} \sum_{t\in T} \sum_{(i,j)\in \bar{A}^t_h} c^t_{ij} x^{th}_{ij} \tag{9}$$

$$s.t. \sum_{(j,i)\in BS^t_h(i)} x^{th}_{ji} - \sum_{(i,j)\in FS^t_h(i)} x^{th}_{ij} = 0 \qquad i \in \bar{N}^t_h \ , \ t \in T \ , \ h \in H \tag{10}$$

$$\sum_{h\in H} \Big(\sum_{(j,v)\in BS^1_h(v)} x^{1h}_{jv} + \sum_{(j,v')\in BS^2_h(v')} x^{2h}_{jv'} + \sum_{(j,v'')\in BS^2_h(v'')} x^{2h}_{jv''} \Big) \geq d_v \quad v \in V \tag{11}$$

$$x^{th}_{p''p'} \leq k^t_h \qquad t \in T \ , \ h \in H \tag{12}$$

$$x^{th}_{ij} \in \mathbb{N} \qquad (i, j) \in \bar{A}^t_h \ , \ t \in T \ , \ h \in H \tag{13}$$

In the formulation, FS^t_h and BS^t_h denote the standard *forward star* and *backward star* of a node, i.e., respectively the set of arcs leaving/entering the node in the network $\bar{G}^t_h$. Thus, (10) are the standard flow conservation constraints for the "commodity" (t, h). The corresponding flow is a *circulation*, i.e., no node produces flow. Together with the fact that the all cycles in the networks necessarily use the return arc (p'', p'), all flow leaving the port (node p') eventually has to return it by reaching node p'' and then traversing the return arc (p'', p'). Thus, constraints (12) enforce the availability of trucks of either type at each time instant; all other arcs of $\bar{A}^t_k$ have no capacity, although of course it could be set to k^t_h as well. Constraints (11) ensure that the demand of each customer $v \in V$ is served, counting services by all truck types and over all possible time instants, Note that there are three terms in the summation. The first corresponds to arcs in $\bar{A}^1_h$ entering the single node v representing the customer in the networks for one-container trucks. The two other terms correspond, respectively, to arcs in $\bar{A}^2_h$ entering the two separate nodes, v' and v'', representing the customer in the networks for two-container trucks. If $v \notin V_h$ then $v \notin \bar{N}^1_h$ and v', $v'' \notin \bar{N}^2_h$; in this case it is obviously intended that the backward stars of these nodes are empty (the corresponding flow variables are not defined), and therefore the three terms of the summation corresponding to that instant h are empty as well. Finally, the objective (9) minimize the overall routing costs; the costs c^t_{ij} of the arcs in $\bar{A}^t_h$ are defined in the obvious way, as described in the previous paragraph, out of the costs of the corresponding arcs in the physical graph G. As already mentioned, the costs of "no-travel legs" (v', v'') are taken to be 0 in our data, although costs related to loading/unloading operations could easily be accounted for if present.

Strictly speaking, the notation of the above formulation only covers the "importers (if any) first, exporters (if any) second" kind of routes. However, it is immediate to expand the formulation by having the flow variables and conservation constraints for both the networks $\bar{G}^2_h$ and $\tilde{G}^2_h$. This clearly means that (12) have to account for the sum of the flow passing through both "return arcs" for the graphs $\bar{G}^2_t$ and $\tilde{G}^2_t$, rather than being simple bound constraints, but this does not really change the model. Thus, at the cost of a (significant) increase in the size of the formulation, all kind of physically feasible routes can be represented. Experience seem to show that the "nonstandard" routes carry such a little benefit [1] as to not warrant the extra computational burden, and besides they may not be available due to organizational constraints of the carrier. However, the opportunity of including them if appropriate is definitely there. Yet, to simplify the discussion we will keep on referring to $(SEAF)$ as using $\bar{G}^2_h$ alone.

It is immediate to see that formulation $(SEAF)$ is correct; it is also possible to show that its continuous relaxation, which we denote by $(\underline{SEAF})$, provides the same lower bound as $(\underline{SC})$. To see that, it is sufficient to consider its Lagrangian relaxation w.r.t. constraints (11) and (12), which decomposes into as many flow subproblems as there are networks $\bar{G}^t_h$. By calling ξ_v and π^t_h respectively the Lagrangian multipliers of (11) and (12), it is immediate to see that the structure of the *Lagrangian costs* is precisely the same as that of the reduced costs of the previous section. Therefore, the solution of the Lagrangian subproblem actually reduces to precisely the same acyclic SPP between p' and p'' as the pricing problem for $(\underline{SC})$. Indeed, if the (reduced = Lagrangian) cost of the shortest path on $\bar{G}^t_h$ (minus π^t_h) is non-negative, then the optimal solution is clearly the all-0 flow vector. Otherwise, the shortest path together with the return arc (p'', p') clearly forms a negative-cost reduced cycle, which proves that the Lagrangian problem is unbounded below (since all its arcs have infinite capacity). Therefore, the characteristic vector of the path (route) plus the return arc is a unbounded ray for the polytope of the Lagrangian subproblem. In other words, solving the Lagrangian Dual with Kelley's Cutting-Plane algorithm, which is equivalent to applying Dantzig-Wolfe decomposition to $(SEAF)$ with (11)–(12) as "complicating" constraints [43], exactly reproduces the row-generation approach to the dual of $(\underline{SC})$, i.e., the column generation approach of the previous paragraph. This proves that the Lagrangian Dual of $(SEAF)$ w.r.t. (11)–(12) is equivalent to $(\underline{SC})$. On the other hand, because the Lagrangian subproblems obviously all have the integrality property, the Lagrangian Dual is equivalent to $(\underline{SEAF})$ [43], which proves that $(SEAF)$ is exactly "as tight" a formulation as (SC).

Since the bound is the same, the two formulations have basically to be compared on the efficiency of the solution methods for the continuous relaxation. The main advantage of $(SEAF)$ is that it is a "more compact" formulation, in that its size grows about quadratically in the number of users rather than quartically. This means that the formulation can be passed whole to a general-purpose solver even for large instances. On the other hand, (SC) can be solved by column generation, which means that the final number of routes in $\mathcal{R}$ can potentially be (much) smaller than the size of the whole $(SEAF)$; yet, these have to be incrementally generated by repeatedly solving the master problem and the subproblems. Although techniques exist to improve the efficiency of column generation approaches (e.g. [44]), the overall cost of this process may be larger than applying just once Linear Programming techniques to $(SEAF)$. All in all, which of the two approaches ends up to be faster in practice can only be determined computationally, which is the aim of the next paragraph. A definite advantage of $(SEAF)$, however, is that very few general-purpose solvers support the full-fledged B&P approach. Thus, using (SC) implies either to significantly restrict the choice of the solution tools, or to resort to heuristics like the P&B, on top of requiring the implementation of the pricing problem, whereas using $(SEAF)$ is significantly simpler with a much larger variety of available tools.

We finish this section by briefly mentioning that $(SEAF)$ could also be solved incrementally and iteratively with extensions of the Dantzig-Wolfe decomposition algorithm [45]; this, however, requires an even more significant programming effort and it is therefore left for further studies.

5 Computational Evaluation

This section is organized in two parts. The objective of the fist part is to analyze the effectiveness in solving the proposed models in reasonable time. In the second part, we aim to quantify which savings can be obtained when increasing flexibility levels are introduced. All tests have been implemented in Python and solved by Cplex 12.5 on a Linux server with 3.00 GHz processor and 16 GB of RAM. Unless otherwise specified, each MIP model has been solved with default parameters, except a maximum running time of 3 hours (10800 seconds) and a required relative gap of 1e-4 (0.01%).

5.1 Computational experiments

We consider a panning horizon of three days. The customers are about equally splitted between no flexibility, medium flexibility, and high flexibility ones. 90% of trucks are supposed to be two-containers ones, and the number of available trucks is the same in every period. The experiments are carried out on small, medium and large instances in the literature.

The small instances are taken from [42]. They are divided into five classes: 6 instances with 10 customers serviced by 28 containers; 10 instances with 20 customers serviced by 61 containers; 14 instances with 30 customers serviced by 88 containers; 18 instances with 40 customers serviced by 125 containers; 22 instances with 50 customers serviced by 141 containers.

The medium-sized instances have adapted from those in [46] for the *Vehicle Routing with Backhauls*, and are denoted from A to N. In order to represent realistic quantities of containers, customer demands have been divided by 100 and rounded to the nearest integer.

The largest instances are adapted from those in [47] for the *Capacitated Vehicle Routing problem.* Nodes have been randomly subdivided between importers and exporters, and customer demands have been divided by 100 and rounded to the nearest integer in order to represent realistic container demands.

The outcomes are reported in Table(1, 2, and 3) for small, medium and large instances, respectively. Each table reports problem data (i.e. $|I|$, $|E|$, k_1, k_2) and three main groups of columns denoted by *SC*, *Price-and-Branch* and *SEAF*. In the first, we report the results in the case of complete enumeration of all feasible routes; in the second, we show the results obtained by the Price-and-Branch algorithm described in Section 3; in the third, we report the results provided by the arc-flow model in Section 4.

For *SC* the following data is reported:

- $|R|$: the total number of feasible routes;
- τ_n: preprocessing time (in seconds) to generate all feasible routes;
- t_n: time (in seconds) to solve the instances by Cplex.

For the Price-and-Branch (*P&B*) the following data is reported:

- *it* is the number of iterations of the column generation algorithm;
- $|B^*|$ is the final number of columns of the RMP at the end of the Price-and-Branch algorithm;
- t_{PB}, t_M, t_P and t_{lm} are the total running time (in seconds) of the Price-and-Branch algorithm, the time for solving the master problem, the time for solving the pricing problem, and the time to solve the restricted integer master problem, respectively.

For *SEAF*, the following data is reported:

- t_{NA}: time (in seconds) for Cplex to solve the model;
- τ_{NA}: preprocessing time (in seconds) to load the model in Cplex;
- N: the total number of branch-and-bound nodes;
- *cut*: the total number of cuts added by Cplex;
- gap_B: the gap (in percentage) between the lower bound at the root node and the optimal solution *before* Cplex cuts; note that this is the same as the gap between the linear relaxation of the RMP and the optimal solution at the end of the column generation;
- gap_A: the gap (in percentage) between the lower bound at the root node and the optimal solution *after* Cplex cuts;

- t_r: time (in seconds) for Cplex to determine the lower bound at root node;
- gap_r: the gap (in percentage) between best integer and lower bound at the root node;

Finally, column *gap* reports the relative gap (in percentage) between the upper bound determined by the integer solution of the RMP in Price-and-Branch algorithm and the optimal solution (found by the other two formulations).

Table 1 shows that small instances can be solved by all methods proposed in the paper. For the smallest of small instances the SC model with complete routes enumeration can actually be the best solution. When size grows, the Price-and-Branch approach can be significantly faster, as it ends up generating only a small fraction of the total routes. However, this comes at the cost of a non-negligible deterioration of the quality of the returned solution (cf. column gap_3), which is acceptable but not optimal. The $(SEAF)$ formulation is clearly the best one, in that it always produces the optimal solution, almost invariably with much shorter running times. It is worth noting that all these instances are solved at the root node by $(SEAF)$; in fact, column N is not reported in that it only contains 0. Thus, while $(\underline{SC})$ and $(\underline{SEAF})$ provide the same bound, which is reasonably tight to start with (cf. column gap_B), Cplex cuts applied to the $(SEAF)$ formulation are enough to completely close the gap (cf. column gap_A), while the columns accrued by the Price-and-Branch to produce the same (starting) lower bound do not allow to generate the optimal solution.

Table 2 shows that only the smallest medium-sized instances can be solved by SC with complete routes enumeration. Although all feasible routes can be generated in a reasonable time, no feasible solution is obtained for instances from H to N within the time limits. The Price-and-Branch approach becomes instrumental here, as it allows to produce solutions for these instances in less than a minute. These solutions are probably still accurate enough for practical use, but they can be in excess of 5% far from the optimal one. Conversely, the $(SEAF)$ formulation obtains optimal solutions in a few seconds; were it not for a single one, all these instances would again be solved at the root node.

Table 3 only shows the total number of routes for the complete SC formulations: these cannot even be generated, much less solved, within the time limit. Except for the three largest instances, Price-and-Branch can provide reasonable-quality solutions in less than one hour. However, even if the tolerance is reduced to 1e-3, the Price-and-Branch does not terminate within three hours and has to be stopped, albeit still providing decent integer solutions in little time once it is. The $(SEAF)$ formulation is capable of always terminating within three hours providing solutions with the required tolerance (albeit, for the largest three instances, that is reduced); save for the very largest instance, no more than half an hour is required.

Thus, according to our experiments the $(SEAF)$ formulation is largely the preferable one. This is even more striking when it is compared with the flow-based compact formulation proposed in [20] for the single-period problem; in order not to move the focus away from our multi-period setting that formulation is recalled in A, and the comparison on single-period instances is reported in B.

5.2 Flexibility and routing costs

Potential savings in routing costs can be achieved when increasing levels of flexibility are introduced. Conversely, when flexibility levels are reduced, it is important to control the increase in routing costs. Therefore, this section aims at investigating how much routing costs are affected by different levels of flexibility. The experiments are carried out on the medium-sized instances of Table 2, which are solved very efficiently by the $(SEAF)$ formulation.

We consider the most restrictive configuration F_0 in which all customers must be served in the first day of the three-days planning horizon and we determine the related routing costs. Next, we consider the following configurations:

- F_1, in which 25% of customers are supposed to have medium flexibility and 75% of them keep having no flexibility;
- F_2, in which 25% of additional customers supposed to have medium flexibility with respect to configuration F_1 (i.e. 50% of customers have medium flexibility and the other customers have no flexibility);

				SC			P&B						SEAF					
$\lvert I\rvert$	$\lvert E\rvert$	k_1	k_2	$\lvert R\rvert$	τ_n	t_n	it	$\lvert B^*\rvert$	t_{PB}	t_M	t_P	t_{lm}	t_{NA}	τ_{NA}	cut	gap_B	gap_A	gap
2	8	2	9	1368	0.02	0.09	10	56	1.95	1.85	0.05	0.20	0.03	0.08	8	0.42	0.00	1.20
5	5	2	7	1258	0.10	0.17	11	54	0.97	0.87	0.05	0.10	0.08	0.03	11	0.89	0.00	1.01
8	2	5	9	884	0.03	0.07	8	50	0.73	0.65	0.04	0.14	0.02	0.07	10	0.90	0.00	1.31
2	8	0	10	1368	0.02	0.09	10	72	1.13	1.02	0.46	0.10	0.04	0.06	15	0.46	0.00	1.74
5	5	0	8	1258	0.10	0.17	9	65	1.14	1.04	0.05	0.12	0.06	0.12	15	0.72	0.00	0.40
8	2	0	12	884	0.03	0.07	8	62	1.22	1.12	0.05	0.53	0.11	0.03	20	0.39	0.00	0.43
2	18	8	22	1792	0.05	0.27	15	89	1.75	1.47	0.16	0.13	0.09	0.12	34	0.19	0.00	1.62
5	15	7	19	7095	0.16	0.40	19	107	1.99	1.71	0.19	0.13	0.06	0.36	8	0.87	0.00	1.36
10	10	5	14	16932	0.34	0.28	22	119	4.08	3.77	0.22	0.57	0.31	0.19	12	0.73	0.00	1.17
15	5	7	19	6345	0.26	1.75	19	107	2.18	1.87	0.21	0.10	0.16	0.82	21	0.24	0.00	0.80
18	2	5	24	1716	0.05	0.31	17	94	2.03	1.77	0.18	0.09	0.23	0.38	18	0.80	0.00	0.42
2	18	0	27	1792	0.05	0.35	18	131	2.99	2.69	0.20	0.16	0.08	0.31	34	0.34	0.00	1.90
5	15	0	23	7095	0.16	0.46	20	133	2.22	1.88	0.22	0.14	0.09	0.28	17	0.34	0.00	1.41
10	10	0	17	16932	0.35	0.61	18	122	4.57	4.29	0.18	0.10	0.09	0.11	17	0.81	0.00	0.77
15	5	0	23	6345	0.26	0.94	21	139	2.88	2.54	0.23	0.09	0.14	0.14	20	0.94	0.00	1.72
18	2	0	27	1716	0.05	0.37	18	131	2.12	1.86	0.17	0.24	0.27	0.24	25	0.83	0.00	2.30
2	28	13	33	5578	0.29	0.26	27	146	3.23	2.63	0.47	0.86	0.13	0.38	34	0.26	0.00	0.10
5	25	12	30	19555	1.49	1.12	23	143	3.00	2.47	0.40	0.15	0.16	0.77	44	0.68	0.00	0.82
10	20	10	25	44730	1.72	4.39	25	156	3.74	3.10	0.49	0.14	0.14	0.79	40	0.79	0.00	2.50
15	15	8	19	54480	2.11	3.48	33	204	4.97	3.99	0.73	0.20	0.23	0.47	58	0.81	0.00	2.43
20	10	10	26	42730	2.15	4.27	26	160	3.91	3.32	0.43	0.17	0.20	0.61	42	0.73	0.00	0.42
25	5	12	32	17055	1.49	1.29	27	143	3.51	2.84	0.51	0.21	0.16	0.15	21	0.63	0.00	0.65
28	2	14	35	4122	0.83	0.28	27	139	3.13	2.57	0.44	0.18	0.14	0.16	19	1.04	0.00	0.46
2	28	0	41	5578	0.29	0.44	29	219	4.12	3.36	0.54	0.13	0.14	0.38	52	1.18	0.00	0.31
5	25	0	36	19555	1.49	3.08	29	199	4.53	3.79	0.53	0.11	0.30	0.77	51	1.01	0.00	2.90
10	20	0	30	44730	1.72	6.31	26	186	7.79	7.22	0.40	0.50	0.11	0.27	41	1.04	0.00	2.10
15	15	0	23	54480	2.11	6.72	36	237	5.60	4.64	0.62	0.14	0.19	0.47	53	1.37	0.00	2.50
20	10	0	31	42730	2.15	9.62	29	186	3.77	3.06	0.51	0.15	0.34	0.61	42	0.66	0.00	2.53
25	5	0	38	17055	1.38	4.10	29	203	3.76	2.99	0.56	0.13	0.14	0.15	26	0.89	0.00	2.11
28	2	0	42	4122	0.83	0.26	28	204	3.58	2.89	0.47	0.15	0.13	0.16	21	1.18	0.00	1.66
2	38	20	49	10228	0.09	1.68	32	176	4.56	3.52	0.83	0.62	0.17	0.10	68	1.20	0.00	3.40
5	35	18	45	38215	1.72	2.34	33	192	4.34	3.28	0.85	0.25	0.20	0.32	31	0.86	0.00	2.71
10	30	14	38	100340	4.19	5.37	32	195	4.75	3.74	0.79	0.26	0.20	0.15	50	0.62	0.00	3.55
15	25	12	31	151265	6.83	7.11	35	219	5.37	4.15	0.93	0.17	0.31	0.19	49	0.51	0.00	3.01
20	20	12	29	252000	8.35	18.4	40	231	7.74	6.41	1.02	0.31	0.19	0.21	50	1.26	0.00	2.44
25	15	14	36	147515	6.39	7.47	38	209	5.21	4.14	0.83	0.12	0.20	0.51	43	0.86	0.00	3.19
30	10	17	43	94340	4.01	5.27	35	195	9.72	8.34	1.10	0.68	0.36	0.31	32	1.19	0.00	2.02
35	5	19	48	32965	1.51	2.12	35	178	5.62	4.54	0.88	0.48	0.17	0.38	50	0.88	0.00	2.22
38	2	20	51	7492	0.09	0.81	30	171	3.83	2.96	0.70	1.15	0.33	0.09	54	1.02	0.00	2.17
2	38	0	60	10228	0.09	5.38	33	259	4.72	3.57	0.82	0.19	0.17	0.10	64	0.31	0.00	2.60
5	35	0	54	3 8215	1.72	3.41	31	248	4.41	3.30	0.82	0.16	0.23	0.32	47	0.33	0.00	1.16
10	30	0	45	100340	4.19	8.35	32	231	8.20	7.18	0.73	0.27	0.28	0.15	48	0.77	0.00	1.18
15	25	0	38	151265	6.83	7.89	35	253	5.56	4.47	0.80	0.11	0.55	0.19	61	1.05	0.00	1.08
20	20	0	35	252000	8.35	18.6	41	280	6.98	5.57	1.05	0.11	0.17	0.21	55	0.40	0.00	0.89
25	15	0	43	147515	6.39	7.52	39	249	5.69	4.43	0.95	0.38	0.31	0.53	50	1.27	0.00	0.88
30	10	0	51	94340	4.01	4.37	37	263	5.96	4.71	0.92	0.54	0.22	0.31	40	0.60	0.00	1.05
35	5	0	58	32965	1.51	1.68	34	255	5.22	4.06	0.86	0.17	0.34	0.38	46	1.01	0.00	1.12
38	2	0	61	7492	0.09	1.14	33	272	4.68	3.49	0.85	0.28	0.41	0.09	49	1.13	0.00	0.95
2	48	22	56	16278	1.02	0.56	37	214	11.1	9.64	1.22	0.50	0.31	0.08	52	1.17	0.00	0.55
5	45	21	54	62850	4.19	1.67	38	217	5.58	3.92	1.39	0.30	0.27	0.13	52	1.18	0.00	0.56
10	40	18	50	177750	11.02	8.01	41	231	6.74	5.07	1.38	0.15	0.25	0.36	67	1.65	0.00	0.60
15	35	17	42	295500	17.97	14.05	41	259	7.38	5.61	1.42	0.18	0.47	0.81	50	0.80	0.00	0.68
20	30	13	37	379350	23.13	23.23	56	301	10.2	8.10	1.73	0.20	0.33	0.67	46	1.25	0.01	0.75
25	25	11	32	407550	24.79	16.23	51	302	8.28	6.22	1.63	0.11	0.30	0.46	37	1.64	0.00	0.79
30	20	12	32	563670	22.27	16.23	47	304	8.79	6.74	1.60	0.13	0.48	0.45	42	1.17	0.00	0.91
35	15	15	39	285000	17.43	13.02	48	285	10.1	7.95	1.77	0.19	0.80	0.35	51	1.49	0.00	0.79
40	10	17	46	165750	9.96	4.61	46	272	7.89	5.94	1.53	0.17	0.39	0.52	39	0.52	0.00	0.74
45	5	20	50	53850	3.39	2.01	41	236	8.19	5.95	1.84	0.27	0.42	0.88	37	0.98	0.00	0.65
48	2	22	55	11862	0.73	1.14	39	226	5.67	4.08	1.31	0.24	0.39	0.41	72	0.59	0.00	0.52
2	48	0	68	16278	1.02	1.17	37	310	5.95	4.30	1.24	0.24	0.36	0.08	12	1.39	0.00	1.44
5	45	0	65	62850	4.19	2.71	39	308	6.52	4.76	1.35	0.26	0.36	0.13	54	1.19	0.00	1.39
10	40	0	60	177750	11.02	5.11	45	323	8.93	6.67	1.76	0.26	0.36	0.29	67	0.71	0.00	1.34
15	35	0	51	295500	17.97	14.91	42	289	6.41	4.66	1.37	0.20	0.42	0.79	57	1.52	0.00	1.40
20	30	0	44	379350	23.13	13.29	48	319	7.66	5.63	1.58	0.17	0.31	0.56	29	1.21	0.00	1.30
25	25	0	38	407550	24.79	17.49	43	303	7.16	5.36	1.38	0.10	0.36	0.31	24	1.66	0.00	1.33
30	20	0	38	563670	22.27	16.75	46	313	7.37	5.40	1.49	0.18	0.27	0.67	46	1.07	0.00	1.47
35	15	0	47	285000	17.43	7.92	48	329	7.59	5.56	1.56	0.39	0.33	0.99	46	0.88	0.00	1.64
40	10	0	55	165750	9.96	3.81	45	307	6.92	5.11	1.41	0.81	0.38	0.70	47	1.63	0.00	1.49
45	5	0	60	53850	3.39	1.84	42	323	6.99	5.15	1.39	0.25	0.36	0.83	57	0.93	0.00	1.47
48	2	0	66	11862	0.73	1.23	39	317	5.93	4.17	1.34	0.24	0.31	0.41	51	1.47	0.00	1.36

Table 1: Experiments on small instances

					SC			P&B						SEAF					
	$\|I\|$	$\|E\|$	k_1	k_2	$\|R\|$	τ_n	t_n	it	$\|B^*\|$	t_{PB}	t_M	t_P	t_{lm}	t_{NA}	τ_{NA}	cut	gap_B	gap_A	gap
A	20	5	4	56	2.1e+4	1.36	0.43	18	116	2.16	1.88	0.19	0.09	0.16	0.83	19	2.45	0.00	3.04
B	20	10	3	51	8.5e+4	2.33	1.86	27	144	2.93	2.40	0.36	0.16	0.34	0.82	26	1.92	0.00	4.18
C	20	20	4	47	3.3e+5	9.62	2.24	35	210	5.01	4.17	0.68	0.13	0.61	0.90	35	2.44	0.00	5.75
D	30	8	7	74	1.2e+5	6.12	9.13	24	158	3.05	2.45	0.45	0.13	0.49	0.59	76	1.78	0.00	4.28
E	30	15	6	69	3.5e+6	7.90	14.1	31	204	4.39	3.54	0.72	0.11	0.72	0.56	69	3.31	0.00	4.67
F	30	30	7	79	1.6e+6	18.01	211	55	328	9.46	7.13	1.87	0.26	0.77	0.52	267	2.44	0.00	5.81
G	45	12	8	112	4.9e+5	10.52	31.5	35	248	5.52	4.05	1.15	0.14	0.62	0.29	49	1.66	0.00	5.52
H	45	23	6	97	1.8e+6	16.83	-	55	308	9.54	6.99	2.12	0.18	1.18	0.84	239	2.22	0.00	6.55
I	45	45	9	129	8.3e+6	37.89	-	81	479	22.9	16.1	5.80	1.00	2.29	0.88	181	1.60	0.00	4.82
J	75	19	12	194	4.1e+6	23.63	-	62	423	13.4	8.80	3.92	0.29	3.82	0.63	188	1.75	0.01	5.81
K	75	38	13	177	1.6e+7	55.3	-	85	513	20.8	12.8	6.89	0.20	4.59	0.81	128	1.46	0.00	5.52
L	75	75	14	202	6.3e+7	190.2	-	140	836	46.8	27.0	17.1	0.42	5.19	0.62	183	1.70	0.00	6.69
M	100	25	17	236	1.2e+7	42.12	-	77	542	19.9	11.4	7.36	0.22	4.08	0.74	140	1.47	0.00	3.22
N	100	50	21	258	4.9e+7	139.9	-	97	660	29.4	15.6	12.1	0.23	5.09	0.88	216	1.46	0.00	4.44

Table 2: Experiments on medium instances

					SC	P&B						SEAF									
	$\|I\|$	$\|E\|$	k_1	k_2	$\|R\|$	it	$\|B^*\|$	t_{PB}	t_M	t_P	t_{lm}	t_{NA}	τ_{NA}	cut	N	gap_B	gap_A	t_r	gap_r	gap
	72	37	55	494	1.5e+7	79	480	18.7	11.5	6.23	0.41	5.3	0.42	126	0	1.17	0.01	4.26	0.62	2.62
	88	20	56	503	9.9e+6	80	517	32.9	17.5	13.5	0.72	16.7	0.66	645	783	0.42	0.03	5.52	0.19	4.76
	103	44	54	489	3.9e+7	111	696	50.0	23.8	25.0	0.24	11.2	0.83	172	4	0.93	0.00	6.50	0.30	5.01
	192	16	143	1283	1.7e+7	104	751	51.1	31.0	17.1	0.19	78	1.20	176	3432	0.40	0.02	10.5	0.09	3.68
	75	175	131	1177	3.5e+8	201	1150	126	46.4	79.0	0.48	81	1.49	492	1352	0.47	0.04	11.2	1.43	4.00
	28	298	225	2016	1.2e+8	251	1108	142	38.7	100	0.39	137	1.72	585	1782	0.45	0.02	18.7	0.77	3.82
	196	196	146	1311	2.7e+9	557	2640	681	220	458	1.63	530	1.93	808	1856	0.49	0.02	32.4	0.62	4.34
	144	335	251	2256	4.5e+9	401	2215	572	126	444	0.34	752	2.11	779	1810	0.30	0.03	46.4	2.47	3.44
	258	254	147	1316	8.3e+9	645	3223	1021	262	756	0.82	924	2.38	1030	1952	0.55	0.04	129	2.19	6.12
	392	168	218	1954	8.5e+9	461	2496	779	152	623	0.71	658	2.53	787	2542	0.55	0.04	192	0.98	4.99
	500	85	380	341	3.3e+9	425	2589	828	136	690	0.54	483	2.34	582	1746	0.31	0.02	124	3.33	3.73
	438	188	329	2958	1.3e+10	566	3083	1173	219	951	0.63	653	2.48	637	9424	0.23	0.02	142	0.14	3.54
	490	210	271	2437	1.9e+10	614	3507	1532	262	1266	0.83	1062	2.81	818	1786	0.24	0.02	335	0.79	5.11
*	251	585	432	3881	4.1e+10	1682	8208	10800	5174	5624	1.89	1497	2.94	704	1765	0.30	0.07	249	48	3.92
*	140	775	579	5205	2.1e+10	1206	6034	10800	4161	6637	1.19	1270	3.24	469	1769	0.23	0.06	252	0.58	4.02
*	500	500	276	2482	1.2e+11	2364	11763	10800	4439	6358	2.18	10467	3.52	661	1775	0.34	0.08	1407	62	6.94

Table 3: Experiments on large instances

- F_3, in which 25% of additional customers supposed to have medium flexibility with respect to configuration F_2 (i.e. 75% of customers have medium flexibility and the other customers have no flexibility);
- F_4, in which 25% of additional customers supposed to have medium flexibility with respect to configuration F_3 (i.e. all customers have medium flexibility).

The outcomes of these experiments are reported in Figure 3. For each instance denoted from A to N, it reports the routing costs associated with these configurations of flexibility. A more careful analysis of these outcomes indicates that the average decrease in routing costs is 2.50% from F_0 to F_1 (standard deviation 2.30%, minimum 0.00%, maximum 7.11%), 5.25% from F_1 to F_2 (standard deviation 3.53%, minimum 1.20%, maximum 15.29%), 3.92% % from F_2 to F_3 (standard deviation 3.01%, minimum 0.00%, maximum 9.03%) and 4.75% from F_3 to F_4 (standard deviation 2.78%, minimum 1.88%, maximum 9.93%).

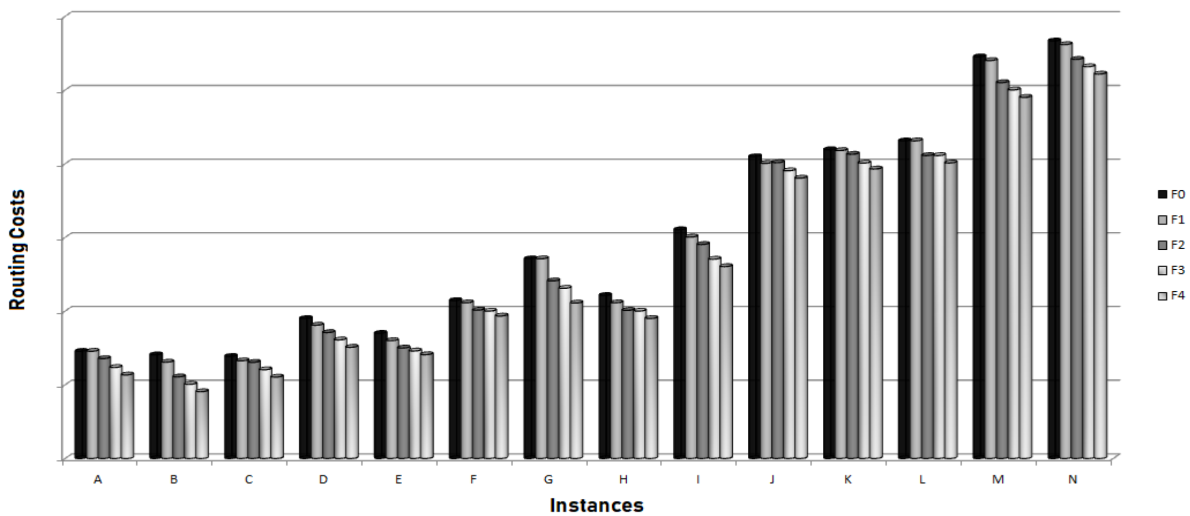

Figure 3: Routing costs if the medium level of flexibility is introduced

The experiments are repeated from configuration F_4, in which all customers must be have medium flexibility, and different levels of high flexibility are introduced. We consider the following configurations:

- F_5, in which 25% of customers are supposed to high medium flexibility and 75% of them keep having medium flexibility;
- F_6, in which 25% of additional customers supposed to have high flexibility with respect to configuration F_5 (i.e. 50% of customers have high flexibility and the other customers have medium flexibility);
- F_7, in which 25% of additional customers supposed to have high flexibility with respect to configuration F_6 (i.e. 75% of customers have high flexibility and the other customers have medium flexibility);
- F_8, in which 25% of additional customers supposed to have high flexibility with respect to configuration F_7 (i.e. all customers have high flexibility).

The outcomes of these experiments are reported in Figure 4, which is organized as Figure 3. These outcomes also indicate that the average decrease in routing costs is 3.21% from F_4 to F_5 (standard deviation 2.76%, minimum 0.57%, maximum 9.05%), 3.60% from F_5 to F_6, (standard deviation 2.68%, minimum 0.04%, maximum 9.96%), 3.81% from F_6 to F_7(standard deviation 2.69%, minimum 0.50%, maximum 9.95%), and 3.16% from F_7 to F_8 (standard deviation 2.28%, minimum 0.37%, maximum 7.64%).

The experiments show that, on these instances, convincing 25% of customers to move one step up in the flexibility ladder is worth about 4% of the routing costs. There is a significant variance in the results,

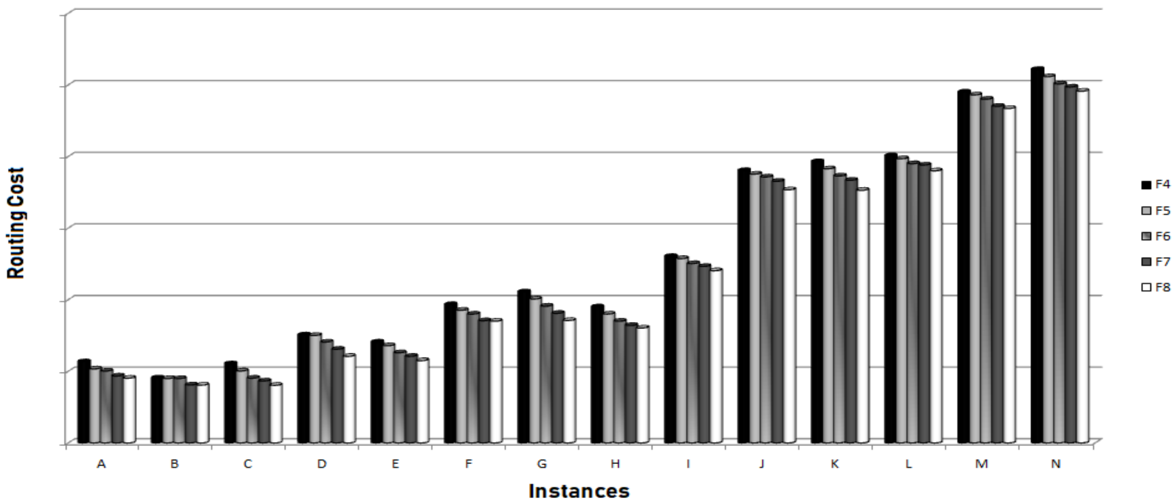

Figure 4: Routing costs if the high level of flexibility is introduced

and the move from no flexibility to medium flexibility seems to be somehow more profitable than that from medium to high flexibility. However, the experiment illustrates that the model can help a carrier to properly set flexibility incentives for its customer.

6 Conclusion

This paper has investigated a multi-period drayage problem with customer-dependent service times. This is a significant contribution in that the literature on drayage problems, despite being quite rich, typically focused on single-day problems and, thus, does not offer proper tools to evaluate the impact of flexibility on distribution costs.

In order to address this problem, different approaches have been adopted. We extended a Set Covering model, and a Price-and-Branch algorithm to the case of multiple periods. Furthermore, we proposed a new compact arc-flow formulation. Extensive experiments have shown the the latter formulation is by far the most effective, as it can determine very quickly the optimal solutions by a MIP solver (very often at the root node). As shown in the Appendices, this model also improves the results obtained for single-day problems. Generally speaking, these outcomes shed light on the importance of proper reformulations, in order to solve problems effectively.

Using the proposed formulation makes it possible to efficiently solve several times even large instances, which allows to quantify the savings obtained by extending service times. From a managerial viewpoint, these outcomes can be useful to negotiate price discounts in exchange for the extended flexibility.

As a future development, we believe that the compact arc-flow formulation can be solved even more efficiently by structured versions of the Dantzig-Wolfe decomposition algorithm. Moreover, we will extend our research by the incorporation of uncertainty in the arrival of new customer requests or cancellations and the possible acceptance of discounts.

Acknowledgements

This research has been partially supported by the PRIN program of the Ministry of Instruction University and Research (MIUR) by the project "Smart PORt Terminals" - SPORT, code 2015XAPRKF. The third author gratefully acknowledges the partial financial support from the Europeans Union's EU Framework Programme for Research and Innovation Horizon 2020 under the Marie Skłodowska-Curie Actions Grant Agreement No 764759 "Mixed-Integer Non Linear Optimisation: Algorithms and Applications".

A The node-arc model by [20]

The model by [20] was proposed to address the single-period version of the drayage problem considered in this paper. That model assumes that all importers must be served before all exporters in any route, an assumption that can be relaxed (albeit at a cost) in the model discussed in this paper. The points in which that model is more general than those in this paper are that:

- it allows to specify different routing costs for each truck;
- it allows to consider truck capacity larger than 2.

However, in practice routing costs only significantly differ between one-container and two-container trucks, being virtually indistinguishable between trucks of the same type. Also, in most (but not all) countries, at most 2 containers can be moved by the same truck.

The model is defined via arc-flow variables on the *physical* directed graph $G = (N, A)$ of Section 3. The model is stil of the *Integer Multicommodity Flow* type, where commodities correspond to all the different trucks k of the available fleet K. In particular, three arc-flow variables are defined:

- x^k_{ij}: binary variable equal to 1 if arc $(i, j) \in A$ is traversed by truck $k \in K$;
- y^k_{ij}: integer variable representing the number of loaded containers moved along arc $(i, j) \in A$ by truck $k \in K$;
- z^k_{ij}: integer variable representing the number of empty containers moved along arc $(i, j) \in A$ by truck $k \in K$.

With these, the problem is formulated as follows:

$$\begin{aligned}
\min\ & \textstyle\sum_{k\in K}\sum_{(i,j)\in A} c^k_{ij}x^k_{ij} && && (14)\\
s.t.\ & \textstyle\sum_{k\in K}\sum_{l\in N} y^k_{il} = \sum_{k\in K}\sum_{j\in p\cup I} y^k_{ji} - d_i && i\in I && (15)\\
& \textstyle\sum_{k\in K}\sum_{l\in N} z^k_{il} = \sum_{k\in K}\sum_{j\in p\cup I} z^k_{ji} + d_i && i\in I && (16)\\
& \textstyle\sum_{l\in N} y^k_{il} \le \sum_{j\in p\cup I} y^k_{ji} && i\in I\ ,\ k\in K && (17)\\
& \textstyle\sum_{l\in N} z^k_{il} \ge \sum_{j\in p\cup I} z^k_{ji} && i\in I\ ,\ k\in K && (18)\\
& \textstyle\sum_{k\in K}\sum_{l\in p\cup E} y^k_{il} = \sum_{k\in K}\sum_{j\in N} y^k_{ji} + d_i && i\in E && (19)\\
& \textstyle\sum_{k\in K}\sum_{l\in p\cup E} z^k_{il} = \sum_{k\in K}\sum_{j\in N} z^k_{ji} - d_i && i\in E && (20)\\
& \textstyle\sum_{l\in p\cup E} y^k_{il} \ge \sum_{j\in N} y^k_{ji} && i\in E\ ,\ k\in K && (21)\\
& \textstyle\sum_{l\in p\cup E} z^k_{il} \le \sum_{j\in N} z^k_{ji} && i\in E\ ,\ k\in K && (22)\\
& \textstyle\sum_{(ji)\in A}(y^k_{ji} + z^k_{ji}) = \sum_{(il)\in A}(y^k_{il} + z^k_{il}) && i\in I\cup E\ ,\ k\in K && (23)\\
& y^k_{ij} + z^k_{ij} \le u_k x^k_{ij} && (i,j)\in A\ ,\ k\in K && (24)\\
& \textstyle\sum_{j\in N} x^k_{ji} - \sum_{l\in N} x^k_{il} = 0 && i\in N\ ,\ k\in K && (25)\\
& \textstyle\sum_{j\in N} x^k_{ij} \le 1 && i\in N\ ,\ k\in K && (26)\\
& \textstyle\sum_{k\in K}\sum_{i\in I\cup E} z^k_{ip} - \sum_{k\in K}\sum_{i\in I\cup E} z^k_{pi} = \sum_{i\in I} d_i - \sum_{i\in E} d_i && && (27)\\
& x^k_{ij} \in \{0,1\} && (i,j)\in A\ ,\ k\in K && (28)\\
& y^k_{ij} \in \mathbb{N} && (i,j)\in A\ ,\ k\in K && (29)\\
& z^k_{ij} \in \mathbb{N} && (i,j)\in A\ ,\ k\in K && (30)
\end{aligned}$$

Routing costs are minimized in the objective function (14); c^k_{ij} is cost for truck $k \in K$ to traverse arc $(i, j) \in A$, which is allowed to change truck-wise (but, in real data, only does so type-wise). Constraints from (15) to (18) concern the movement of containers to importers. Constraints (15) and (16) are the flow conservation constraints of loaded and empty containers, respectively, at each importer node. Constraint (17) and (18) check the number of loaded and empty containers in each truck entering and leaving from importers: the number of loaded containers cannot be increased after a service at each importer, whereas the number of empty containers cannot be reduced.

Constraints from (19) to (22) concern the allocation of containers to exporters. Constraints (19) and (20) are the flow conservation constraints of loaded and empty containers, respectively, for each exporter node. Constraint (21) and (22) control the number of loaded and empty containers in each truck entering and leaving from exporters: the number of loaded containers cannot be reduced after a service at each exporter, whereas the number of empty containers cannot be increased.

Constraint (23) guarantees that the number of containers carried by each truck does not change after visiting a customer. Constraint (24) imposes that the number of containers moved by each truck is not larger than its transportation capacity u_k. In this problem, u_k takes value 1 or 2. Constraints (25) are the flow conservation constraints for trucks at each node. Constraint (26) guarantees that trucks are not used more than once. Constraint (27) represents the flow conservation of empty containers at port p. Finally, constraints (28), (29) and (30) define the domain of decision variables.

B Solutions for single-day problems

Single-day instances coming from real-world data are taken from [1]. The results are reported in Table 4 and Table 5, which are organized as Table 1 and 2 apart from column gap_F, which reports the optimality gap for the model (14)–(30) (or "-" when no feasible solution is obtained). The computational setting is the same as the other experiments, except that the maximum running time set at 1 hour for all the approaches.

The experiments show that model (14)–(30) can solve to optimality only five small instances, it does not prove the optimality of 25 feasible solutions, and cannot provide any feasible solution for 54 out of 84 instances. This is due to the combination of its larger size, having truck-specific variables, and a much weaker lower bound. The (SC) model with enumeration of all feasible routes is viable to get the optimal solution of small instances, but it quickly becomes impractical as the size grows, as the medium instances show. The Price-and-Branch is viable for all size and returns very good solutions for small instances (cf. Column gap_3 in Table 4), but the quality of the solution is significantly worse for medium ones (cf. Column gap_3 in Table 5). Remarkably, while the small instances come from real-world data of an actual drayage application, the medium ones come from adapting data for other versions of the Vehicle Routing Problem; thus, these results may indicate that real-world instances are in fact easier to solve by the Price-and-Branch than what Table 5 would seem to indicate. However, the point is largely moot, since the $(SEAF)$ formulation clearly outperforms all the others by optimally solving all the single-period instances very quickly *at the root node*. The inherent gap of the formulation (cf. Column gap_B) is very small already for small instances, that are in fact solved very well by the Price-and-Branch, but it is much larger for medium ones: yet, Cplex cuts are capable of completely closing it without any branching, ensuring exceptionally low computing times in all cases. This proves that the $(SEAF)$ formulation is also very effective in the single-period case.

References

[1] A. Ghezelsoflu, M. Di Francesco, A. Frangioni, p. Zuddas, A set-covering formulation for a drayage problem with single and double container loads, Journal of Industrial Engineering International 4 (14) (2018) 665–676.

[2] X. Wang, A. C. Regan, Local truckload pickup and delivery with hard time window constraints, Transportation Research Part B: Methodological 36 (2002) 97–112.

[3] K. Chung, C. Ko, J. Shin, H. Hwang, K. Kim, Development of mathematical models for the container road transportation in korean trucking industries, Computers & Industrial Engineering 53 (2) (2007) 252–262.

[4] P. Francis, G. Zhang, K. Smilowitz, Improved modeling and solution methods for the multi-resource routing problem, European Journal of Operational Research 180 (2007) 1045–1059.

[5] R. Zhang, W. Yun, I. Moon, A reactive tabu search algorithm for the multi-depot container truck transportation problem, Transportation Research Part E: Logistics and Transportation Review 45 (6) (2009) 904–914.

					SC				*P&B*					*SEAF*						
$\|I\|$	$\|E\|$	k_1	k_2	R	τ_n	t_n	it	$\|B^*\|$	t_{PB}	t_M	t_P	t_{lm}	t_{NA}	τ_{NA}	cut	gap_B	gap_A	gap	gap_F
2	8	2	9	462	0.02	0.09	14	40	0.57	0.55	0.00	0.07	0.06	0.06	8	0.10	0.00	0.003	0.00
5	5	2	7	810	0.10	0.17	14	42	0.48	0.45	0.00	0.06	0.10	0.04	8	0.09	0.00	0.005	0.00
8	2	5	9	366	0.03	0.07	13	40	0.45	0.43	0.00	0.07	0.01	0.05	0	0.11	0.00	0.004	0.00
2	8	0	10	462	0.02	0.09	14	40	0.65	0.63	0.00	0.05	0.07	0.07	4	0.12	0.00	0.014	3.76
5	5	0	8	810	0.10	0.17	14	42	0.60	0.57	0.00	0.04	0.02	0.05	12	0.07	0.00	0.024	0.00
8	2	0	12	366	0.03	0.07	13	40	0.57	0.54	0.00	0.04	0.16	0.03	6	0.10	0.00	0.005	3.83
2	18	8	22	1792	0.05	0.27	24	81	1.12	1.07	0.03	0.06	0.07	0.09	17	0.06	0.00	0.006	2.51
5	15	7	19	7095	0.16	0.40	30	86	1.38	1.28	0.04	0.15	0.12	0.21	9	0.07	0.00	0.023	0.00
10	10	5	14	11220	0.34	0.28	31	84	1.47	1.39	0.05	0.05	0.07	0.11	19	0.06	0.00	0.016	2.26
15	5	7	19	6345	0.26	1.75	28	83	1.54	1.41	0.04	0.21	0.19	0.63	25	0.07	0.00	0.005	2.87
18	2	5	24	1716	0.05	0.31	26	81	1.19	1.13	0.04	0.05	0.19	0.37	13	0.05	0.00	0.009	2.53
2	18	0	27	1792	0.05	0.35	24	81	1.11	1.05	0.03	0.08	0.11	0.28	17	0.05	0.00	0.004	5.74
5	15	0	23	7095	0.16	0.46	30	86	1.49	1.40	0.05	0.06	0.11	0.17	13	0.04	0.00	0.006	2.99
10	10	0	17	11220	0.35	0.61	31	84	2.40	2.33	0.04	1.18	0.08	0.12	13	0.05	0.00	0.000	3.91
15	5	0	23	6345	0.26	0.94	28	83	1.58	1.50	0.05	0.13	0.22	0.10	15	0.06	0.00	0.014	6.01
18	2	0	27	1716	0.05	0.37	26	81	1.44	1.37	0.04	0.18	0.10	0.19	25	0.05	0.00	0.004	4.96
2	28	13	33	5578	0.29	0.26	39	124	1.98	1.81	0.11	0.10	0.14	0.29	12	0.06	0.00	0.002	7.54
5	25	12	30	19555	1.49	1.12	42	124	2.11	1.96	0.11	0.13	0.13	0.50	14	0.07	0.00	0.007	-
10	20	10	25	44730	1.72	4.39	46	127	3.21	3.00	0.15	0.14	0.22	0.44	20	0.04	0.00	0.028	2.98
15	15	8	19	54480	2.11	3.48	54	143	3.54	3.33	0.15	0.31	0.12	0.19	9	0.05	0.00	0.021	2.69
20	10	10	26	42730	2.15	4.27	48	131	2.74	2.56	0.13	0.07	0.32	0.37	25	0.07	0.00	0.002	9.08
25	5	12	32	17055	1.49	1.29	38	122	1.78	1.65	0.10	0.06	0.12	0.11	14	0.06	0.00	0.008	3.21
28	2	14	35	4122	0.83	0.28	33	119	1.60	1.49	0.08	0.08	0.12	0.12	10	0.05	0.00	0.004	-
2	28	0	41	5578	0.29	0.44	39	124	2.19	2.04	0.11	0.18	0.30	0.24	14	0.04	0.00	0.032	7.02
5	25	0	36	19555	1.49	3.08	42	124	2.31	2.15	0.12	0.07	0.14	0.51	12	0.04	0.00	0.003	11.4
10	20	0	30	44730	1.72	6.31	46	127	2.52	2.29	0.16	0.21	0.15	0.19	25	0.05	0.00	0.002	9.38
15	15	0	23	54480	2.11	6.72	54	143	3.09	2.90	0.13	0.05	0.29	0.38	40	0.05	0.00	0.014	6.61
20	10	0	31	42730	2.15	9.62	48	131	3.20	3.00	0.15	0.05	0.26	0.52	41	0.06	0.00	0.011	7.16
25	5	0	38	17055	1.38	4.10	38	122	2.26	2.11	0.10	0.11	0.14	0.09	14	0.04	0.00	0.025	7.07
28	2	0	42	4122	0.83	0.26	33	119	1.87	1.74	0.09	0.07	0.25	0.14	16	0.03	0.00	0.025	3.79
2	38	20	49	10228	0.09	1.68	44	160	2.23	1.98	0.20	0.13	0.24	0.13	11	0.03	0.00	0.067	-
5	35	18	45	38215	1.72	2.34	49	161	2.57	2.29	0.23	0.10	0.18	0.25	8	0.06	0.00	0.020	-
10	30	14	38	100340	4.19	5.37	57	165	4.08	3.75	0.27	0.10	0.17	0.09	8	0.03	0.00	0.047	-
15	25	12	31	151265	6.83	7.11	60	171	4.48	4.01	0.35	0.14	0.53	0.11	27	0.04	0.00	0.034	-
20	20	12	29	168840	8.35	18.43	69	179	4.44	3.99	0.37	0.09	0.27	0.17	7	0.06	0.00	0.030	-
25	15	14	36	147515	6.39	7.47	59	160	3.85	3.53	0.23	0.06	0.18	0.38	16	0.07	0.00	0.029	-
30	10	17	43	94340	4.01	5.27	54	164	3.14	2.85	0.22	0.11	0.20	0.20	10	0.06	0.00	0.028	-
35	5	19	48	32965	1.51	2.12	45	159	3.28	3.02	0.22	0.13	0.16	0.21	13	0.06	0.00	0.028	-
38	2	20	51	7492	0.09	0.81	45	162	2.46	2.22	0.18	0.20	0.33	0.07	12	0.06	0.00	0.023	-
2	38	0	60	10228	0.09	5.38	44	160	2.96	2.72	0.20	0.18	0.36	0.06	18	0.01	0.00	0.069	-
5	35	0	54	38215	1.72	3.41	49	161	2.98	2.71	0.22	0.29	0.21	0.19	16	0.02	0.00	0.070	-
10	30	0	45	100340	4.19	8.35	57	165	4.58	4.23	0.29	0.16	0.20	0.12	20	0.02	0.00	0.089	-
15	25	0	38	151265	6.83	7.89	60	171	5.71	5.25	0.39	0.49	0.71	0.13	42	0.02	0.00	0.081	-
20	20	0	35	168840	8.35	18.59	69	179	5.25	4.79	0.38	0.12	0.17	0.17	13	0.02	0.00	0.070	-
25	15	0	43	147515	6.39	7.52	59	165	4.45	4.01	0.36	0.60	0.29	0.37	23	0.02	0.00	0.068	18.9
30	10	0	51	94340	4.01	4.37	54	164	3.38	3.11	0.22	0.10	0.20	0.17	17	0.02	0.00	0.076	-
35	5	0	58	32965	1.51	1.68	45	159	2.85	2.59	0.20	0.13	0.40	0.26	19	0.01	0.00	0.070	-
38	2	0	61	7492	0.09	1.14	45	162	2.66	2.40	0.18	0.18	0.31	0.07	26	0.01	0.00	0.066	13.6
2	48	22	56	16278	1.02	0.56	56	203	3.25	2.75	0.43	0.23	0.34	0.05	19	0.07	0.00	0.024	-
5	45	21	54	62850	4.19	1.67	62	202	3.79	3.28	0.44	0.30	0.31	0.08	24	0.06	0.00	0.033	-
10	40	18	50	177750	11.02	8.01	70	202	5.62	4.95	0.58	0.10	0.56	0.18	15	0.03	0.00	0.072	-
15	35	17	42	295500	17.97	14.05	70	201	5.15	4.41	0.58	1.52	0.38	0.53	19	0.03	0.00	0.073	-
20	30	13	37	379350	23.13	23.23	80	210	7.01	6.25	0.67	0.08	0.32	0.45	19	0.03	0.00	0.073	-
25	25	11	32	407550	24.79	16.23	77	218	4.70	4.13	0.49	0.09	0.37	0.37	16	0.03	0.00	0.071	-
30	20	12	32	373350	22.27	16.23	78	219	4.88	4.33	0.47	0.07	0.34	0.31	19	0.03	0.00	0.094	-
35	15	15	39	285000	17.43	13.02	73	206	4.45	3.93	0.45	0.18	0.33	0.18	16	0.03	0.00	0.091	-
40	10	17	46	165750	9.96	4.61	73	211	4.53	4.01	0.46	0.10	0.32	0.27	17	0.03	0.01	0.085	-
45	5	20	50	53850	3.39	2.01	62	205	3.87	3.38	0.40	0.12	0.33	0.63	18	0.03	0.00	0.064	-
48	2	22	55	11862	0.73	1.14	55	201	3.39	2.95	0.38	0.15	0.79	0.30	26	0.03	0.00	0.049	-
2	48	0	68	16278	1.02	1.17	56	203	3.94	3.46	0.42	0.14	0.30	0.11	26	0.01	0.00	0.088	-
5	45	0	65	62850	4.19	2.71	62	202	4.07	3.60	0.39	0.15	0.26	0.06	24	0.01	0.00	0.095	-
10	40	0	60	177750	11.02	5.11	70	202	5.59	4.97	0.54	0.26	0.34	0.23	32	0.01	0.00	0.099	-
15	35	0	51	295500	17.97	14.91	70	201	7.25	6.33	0.70	1.43	0.41	0.56	35	0.02	0.00	0.010	-
20	30	0	44	379350	23.13	13.29	80	210	7.99	7.15	0.73	0.15	0.22	0.42	26	0.02	0.00	0.096	-
25	25	0	38	407550	24.79	17.49	77	218	5.14	4.59	0.48	0.07	0.22	0.26	16	0.02	0.00	0.094	-
30	20	0	38	373350	22.27	16.75	78	219	5.41	4.87	0.47	0.08	0.23	0.19	19	0.01	0.00	0.012	-
35	15	0	47	285000	17.43	7.92	73	206	4.85	4.36	0.42	0.25	0.19	0.47	16	0.01	0.00	0.012	-
40	10	0	55	165750	9.96	3.81	73	211	4.93	4.43	0.43	0.12	0.38	0.54	22	0.01	0.00	0.011	-
45	5	0	60	53850	3.39	1.84	62	205	4.49	4.04	0.38	0.16	0.29	0.66	28	0.02	0.00	0.098	-
48	2	0	66	11862	0.73	1.23	55	201	3.80	3.34	0.39	0.14	0.28	0.32	28	0.01	0.00	0.085	-

Table 4: Experiments with small single-period instances

					SC			P&B						SEAF						
	$\|I\|$	$\|E\|$	k_1	k_2	$\|R\|$	τ_n	t_n	it	$\|B^*\|$	t_{PB}	t_M	t_P	t_{lm}	t_{NA}	τ_{NA}	cut	gap_B	gap_A	gap	gap_F
A	20	5	4	56	1.1e+4	1.01	0.22	9	59	1.14	0.83	0.27	0.03	0.06	0.65	3	2.45	0.00	3.04	-
B	20	10	3	51	4.3e+4	1.94	1.19	14	94	2.06	1.24	0.66	0.15	0.11	0.68	5	1.92	0.00	3.97	-
C	20	20	4	47	1.7e+5	8.98	1.02	19	128	2.97	1.64	1.04	0.22	0.47	0.74	30	2.44	0.00	5.08	-
D	30	8	7	74	6.1e+4	5.03	9.09	14	94	2.06	1.28	1.05	0.27	0.19	0.32	18	1.78	0.00	4.29	-
E	30	15	6	69	2.1e+5	6.83	13.3	21	132	3.49	2.08	1.02	0.30	0.41	0.22	31	3.31	0.00	4.11	-
F	30	30	7	79	8.4e+5	15.7	202	31	206	6.12	3.92	1.61	0.49	0.63	0.29	12	2.44	0.00	5.85	-
G	45	12	8	112	3.0e+5	8.65	16.4	22	136	3.43	1.98	0.99	0.42	0.31	0.13	27	1.66	0.00	4.61	-
H	45	23	6	97	1.1e+6	13.1	2455	29	194	6.48	4.71	1.11	0.37	0.92	0.50	40	2.22	0.00	7.04	-
I	45	45	9	129	4.2e+6	30.5	782	48	300	11.7	9.17	2.16	0.31	1.23	0.79	42	1.60	0.00	5.18	-
J	75	19	12	194	2.1e+6	18.5	5560	35	230	8.13	5.81	1.75	0.48	3.12	1.07	371	1.75	0.00	5.03	-
K	75	38	13	177	8.2e+6	49.3	-	53	340	16.2	11.4	4.17	0.40	2.38	1.09	60	1.46	0.00	4.79	-
L	75	75	14	202	3.2e+7	157	-	89	556	34.5	22.4	11.6	0.36	3.12	2.39	92	1.41	0.00	7.16	-
M	100	25	17	236	6.3e+6	38.1	-	51	313	15.3	10.8	3.99	0.20	2.53	1.11	56	1.32	0.00	4.79	-
N	100	50	21	258	2.5e+7	107	-	68	425	24.7	16.2	8.01	0.34	3.51	2.01	62	1.35	0.01	4.80	-

Table 5: Experiments with single-period medium instances

[6] R. Zhang, W. Yun, H. Kopfer, Heuristic-based truck scheduling for inland container transportation, OR Spectrum 32 (3) (2010) 787–808.

[7] R. Zhang, W. Yun, I. Moon, Modeling and optimization of a container drayage problem with resource constraints, International journal of Production Economics 133 (1) (2011) 351–359.

[8] M. Vidovic, G. Radivojevic, B. Rakovic, Vehicle routing in containers pickup up and delivery processes, Procedia-Social and Behavioral Sciences 20 (2011) 335–343.

[9] K. Braekers, A. Caris, G. Janssens, Integrated planning of loaded and empty container movements, OR Spectrum 35 (2) (2013) 457–478.

[10] S. Sterzik, H. Kopfer, A tabu search heuristic for the inland container transportation problem, Computers & Operations Research 40 (4) (2013) 953–962.

[11] W. F. Wang, W. Y. Yun, Scheduling for inland container truck and train transportation, International journal of Production Economics 143 (2013) 349–356.

[12] J. Nossack, E. Pesch, A truck scheduling problem arising in intermodal container transportation, European Journal of Operational Research 230 (3) (2013) 666–680.

[13] K. Braekers, A. Caris, G. K. Janssens, Bi-objective optimization of drayage operations in the service area of intermodal terminals, Transportation Research Part E: Logistics and Transportation Review 65 (2014) 50–69.

[14] Z. Xue, C. Zhang, W.-H. Lin, L. Miao, P. Yang, A tabu search heuristic for the local container drayage problem under a new operation mode, Transportation Research Part E: Logistics and Transportation Review 62 (2014) 136–150.

[15] Y. Song, J. Zhang, Z. Liang, C. Ye, An exact algorithm for the container drayage problem under a separation mode, Transportation Research Part E: Logistics and Transportation Review 106 (2017) 231–254.

[16] R. Zhang, W. Yun, H. Kopfer, Multi-size container transportation by truck: modeling and optimization, Flexible Services & Manufacturing 2-3 (27) (2015) 403–430.

[17] J. Funke, H. Kopfer, A model for a multi-size inland container transportation problem, Transportation Research Part E: Logistics and Transportation Review 89 (2016) 70–85.

[18] M. Vidović, D. Popović, B. Ratković, G. Radivojević, Generalized mixed integer and vns heuristic approach to solving the multisize containers drayage problem, International Transactions in Operational Research 24 (3) (2017) 583–614.

[19] A. Ghezelsoflu, M. Di Francesco, A. Frangioni, p. Zuddas, A price-and-branch algorithm for a drayage problem with heterogeneous trucks, Journal of Electronic Notes in Discrete Mathematics 69 (2018) 237–244.

[20] M. Lai, M. Battarra, M. Di Francesco, P. Zuddas, An adaptive guidance meta-heuristic for the vehicle routing problem with splits and clustered backhauls, Journal of the Operational Research Society 66 (2015) 1222–1236.

[21] A. Estrada-Moreno, M. Savelsbergh, A. A. Juan, J. Panadero, Biased-randomized iterated local search for a multiperiod vehicle routing problem with price discounts for delivery flexibility, International Transactions in Operational Research 26 (4) (2019) 1293–1314.

[22] P. Toth, D. Vigo, Vehicle routing problems, methods, and applications, SIAM (2014) 1 – 465.

[23] M. Wen, J. F. Cordeau, G. Laporte, J. Larsen, The dynamic multi-period vehicle routing problem, Computers & Operations Research 9 (37) (2010) 1615–1623.

[24] C. Archetti, O. Jabali, M. G. Speranza, Multi-period vehicle routing problem with due dates, Computers & Operations Research 61 (2015) 122 – 134.

[25] J. Pacheco, A. Alvarez, I. García, F. Angel-Bello, Optimizing vehicle routes in a bakery company allowing flexibility in delivery dates, Journal of the Operational Research Society 63 (5) (2012) 569–581.

[26] P. Francis, K. Smilowitz, M. Tzur, The period vehicle routing problem with service choice, Transportation Science 40 (4) (2006) 439–454.

[27] A. Campbell, J. Wilson, Forty years of periodic vehicle routing, Networks 63 (3) (2014) 2–15.

[28] E. Beltrami, L. Bodin, Networks and vehicle routing for municipal waste collection, Networks 4 (1) (1974) 65–94.

[29] R. Baldacci, E. Bartolini, A. Mingozzi, A. Valletta, An exact algorithm for the period routing problem, Operations Research 59 (1) (2011) 228 – 241.

[30] V. Hemmelmayr, R. Hartl, K. Doerner, A variable neighborhood search heuristic for periodic routing problems, European Journal of Operational Research 195 (2009) 791–802.

[31] T. Vidal, G. T.G., M. Gendreau, N. Lahrichi, W. Rei, a hybrid genetic algorithm for multidepot and periodic vehicle routing problems, Operational Research 60 (3) (2012) 611–624.

[32] V. Cacchiani, V. Hemmelmayr, F. Tricoire, A set-covering based heuristic algorithm for the periodic vehicle routing problem, Discrete Applied Mathematics 163 (2014) 53–64.

[33] Y. Crama, M. Rezaei, T. Van Woensel, A branch-and-price algorithm for 2-period vehicle routing problems, Business & economic sciences : Production, distribution & supply chain management (2015) 1–39.

[34] F. Hernandez, M. Gendreau, J. Potvin, Heuristics for tactical time slot management: a periodic vehicle routing problem view, IFORS 24 (6) (2017) 1233–1252.

[35] L. Bertazzi, M. G. Speranza, Inventory routing problems with multiple customers, EURO Journal on Transportation and Logistics 2 (3) (2013) 255–275.

[36] C. Archetti, N. Boland, M. Grazia Speranza, A matheuristic for the multivehicle inventory routing problem, INFORMS Journal on Computing 29 (3) (2017) 377–387.

[37] Y. Crama, M. Rezaei, M. Savelsbergh, T. Van Woensel, Stochastic inventory routing for perishable products, Transportation Science 52 (3) (2018) 526–546.

[38] C. Archetti, M. Christiansen, M. G. Speranza, Inventory routing with pickups and deliveries, European Journal of Operational Research 268 (1) (2018) 314–324.

[39] C. Archetti, E. Fernández, D. L. Huerta-Muñoz, The flexible periodic vehicle routing problem, Computers & Operations Research 85 (2017) 58–70.

[40] C. Archetti, E. Fernández, D. L. Huerta-Muñoz, A two-phase solution algorithm for the flexible periodic vehicle routing problem, Computers & Operations Research 99 (2018) 27–37.

[41] G. Zhang, K. Smilowitz, A. Erera, Dynamic planning for urban drayage operations, Transportation Research Part E: Logistics and Transportation Review 47 (5) (2011) 764 – 777.

[42] M. Lai, T. G. Crainic, M. Di Francesco, P. Zuddas, An heuristic search for the routing of heterogeneous trucks with single and double container loads, Transportation Research Part E: Logistics and Transportation Review 56 (2013) 108–118.

[43] A. Frangioni, About Lagrangian Methods in Integer Optimization, Annals of Operations Research 139 (1) (2005) 163–193.

[44] H. Ben Amor, J. Desrosiers, A. Frangioni, On the Choice of Explicit Stabilizing Terms in Column Generation, Discrete Applied Mathematics 157 (6) (2009) 1167–1184.

[45] A. Frangioni, B. Gendron, A Stabilized Structured Dantzig-Wolfe Decomposition Method, Mathematical Programming 140 (2013) 45–76.

[46] M. Goetschalckx, C. Jacobs-Blecha, The vehicle routing problem with backhauls, European Journal of Operational Research 42 (1) (1989) 39–51.

[47] E. Uchoa, D. Pecin, A. Pessoa, M. Poggi, T. Vidal, A. Subramanian, New benchmark instances for the capacitated vehicle routing problem, European Journal of Operational Research 257 (3) (2017) 845 – 858.