A multiperiod drayage problem with flexible service periods

Ali Ghezelsoflu[1], Massimo Di Francesco[2]
Antonio Frangioni[1], Paola Zuddas[2]

[1] Department of Computer Science, University of Pisa, Italy
ali.ghezelsoflu@di.unipi.it, frangio@di.unipi.it
[2] Department of Mathematics and Computer Science, University of Cagliari, Italy
mdifrance@unica.it, zuddas@unica.it

Abstract

We investigate a multiperiod drayage problem in which customers can request transportation services over several days and the carrier has some flexibility to change service periods. We discuss and compare three approaches for the problem: a path-based model with all feasible routes, a Price-and-Branch algorithm in which the pricing is formulated as a collection of shortest path problems in a cunningly constructed acyclic network, and a compact arc-flow formulation based on this network.

Keywords : *Combinatorial Optimization, Graphs, Price-and-Branch, Reformulations.*

1 Introduction

Although drayage operations represent a small fraction of the total distance of an intermodal shipment, they account for a relevant share of the overall shipping costs. As a result, they have been subject to a significant amount of research. However, it has typically focused on single-day problems and, thus, has not offered proper tools to evaluate the impact of flexibility on distribution costs. When customers accept to be served over different days and the storage space in port terminals (and in their close depots) is sufficient, one is requested to jointly plan when customers should be served and which routes should be made to minimize costs.

In this problem, two types of customers must be served: importers receiving container loads from port and exporters shipping container loads to the same port. Routes are performed by two types of trucks, which are based at the port and carry one or two containers at a time. Two-container trucks have slightly larger routing costs per unitary distance as opposed to one-container-trucks, but they can serve twice the demand. Each transportation request specifies where customers are located, how many container loads must be delivered or collected, and when they can be served so that all drayage operations are completed within the planning horizon. Customers can have different *flexibility levels* regarding their availability to receive a transportation request before or after the specified due-date. A customer without flexibility only accept to receive services at the specified due-date for each. For more flexible customers, the carrier pays customer-specific penalties for earlier/later than desired services within customer-specific flexibility periods, and there exists a customer-specific capacity on the maximum number of container loads that can be served early or late during flexibility periods. Furthermore, there exist a global capacity and a cost associated with containers left at the port for late delivery and early collection with respect to the desired service date.

We will discuss three different approaches to this problem. First, since up to four customers can be visited along each route in each period, a natural model enumerates all the (polynomially many) feasible routes in a period-dependent network, where nodes represent the port and customers and arcs indicate the possible connections among them. The resulting path-based formulation is solved with all feasible routes by an off-the-shelf MIP solver. Second, since the number of routes may be too large, the Linear Programming relaxation of this formulation is solved by Column Generation, where the pricing subproblem is a collection of acyclic shortest path problems on appropriate auxiliary networks. From the set of generated columns, an integer solution is sought by a MIP solver under the Price-and-Branch paradigm. Third, a arc-flow formulation based on a slight modification of the previous auxiliary network is proposed and solved very effectively by a general-purpose MIP solver.

2 First approach: the path-based formulation

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 $V = I \cup E$, in which I and E are the set of importers and exporters, respectively. Any arc $(i, j) \in A$ represents the direct truck trip between i and j, and has the two associated costs for one- and two-containers trucks, respectively. It is immediate to construct the (physical) sub-graphs $G^t_h = (N_h, A^t_h)$ of G for each period $h \in H$ and type of truck $t \in T = \{1, 2\}$, in which $N_h = \{p\} \cup V_h$, $V_h = I_h \cup E_h$, I_h and E_h denote the set of importers and exporters who accept to be served in period $h \in H$, respectively. The arc sets A^t_h are those induced by N_h and must be feasible for the given truck type: 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.

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 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.

Constructing a the path-based formulation (PBF) 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 served 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 served by one container, and 0 if v is not visited in route r. An integer decision variable represents the number of times in which route r is traversed. Moreover, 'slacks" s^{h+}_v and s^{h-}_v are introduced to model the amount of container loads that, respectively, have not been delivered/collected on time or have been delivered/retrieved earlier than agreed for customer $v \in V$ in day $h \in H$.

Routing costs and penalties for customers are minimized in the objective function. Constraints ensure that routes serve the overall demand customers, but it is possible to deliver/collect container loads to/from the customers either earlier or later than the expected date. Moreover, the number of routes performed in each period is lower than the number of available trucks for each type. Constraints enforce the maximum number of container loads that can be delivered/collected early or late to/from the customers and kept in the port.

3 Second approach: the Price-and-Branch algorithm

For a relatively small number of customers, it is possible to just statically enumerate all possible routes. However, when the number of customers grow, the number of feasible routes quickly becomes rather large. In this case, the LP relaxation ($\underline{PBF}$) of (PBF) can be solved by the well-known column generation technique. This starts by forming the *Restricted Master Problem* (RMP), which is ($\underline{PBF}$) where the full set of routes R is replaced by a (much) smaller subset $\mathcal{R} \subset R$.

The key of the approach is that the dual of ($\underline{PBF}$) has constraints corresponding to each $r \in R$, as opposed to only those for $r \in \mathcal{R}$ that are present in the dual of (RMP). Hence, one only have to prove that all the constraints corresponding to each $r \in R$ are satisfied by the current dual solution, which proves the optimality for the whole ($\underline{PBF}$) of the corresponding optimal primal solution 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 $\bar{r}$ is negative. The fundamental observation is that, discarding a constant term, 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). Thus, determining a route of negative reduced cost can be reduced to a collection of Shortest Path Problems (SPP) on tailor-made acyclic networks $\bar{G}^t_h = (\bar{N}^t_h, \bar{A}^t_h)$, one for each $t \in T$ and $h \in H$. The topology of these graphs is readily obtained by that of the corresponding (physical) sub-graphs $G^t_h = (N_h, A^t_h)$ 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. The arcs in $\bar{A}^1_h$ represent possible trips performed by one-container trucks, i.e., they are basically those of A^1_h save that arcs leaving p now leave p' and arcs entering p now enter p''. For $t = 2$, 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 serve 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. An important property of $\bar{G}^t_h$ is that is 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.

With the ability of efficiently solving the *pricing problem*(s) for ($\underline{PBF}$), 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. 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: 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. The P&B is expected to be quite effective and efficient when the root node gap of the formulation is low.

4 Third approach: a compact 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 (PBF) 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. This formulation is called ($SEAF$).

To do that, it is only necessary to minimally expand the former acyclic networks $\bar{G}^t_h$ by adding the single "return arc" (p'', p'). 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) and we keep using "slacks" variables as in (PBF).

Routing costs and penalties for customers are minimized in the objective function. Constraints ensure that routes serve the overall demand customers, but it is possible to deliver/collect container loads to/from the customers either earlier or later than the expected date. Moreover, the number of routes performed in each period is lower than the number of available trucks for each type. Constraints also enforce the maximum number of container loads that can be delivered/collected early or late to/from the customers and kept in the port.

We will show that its continuous relaxation, which we denote by ($\underline{SEAF}$), provides the same lower bound as ($\underline{PBF}$) by a Lagrangian relaxation. By giving names to the Lagrangian multipliers equal to that of dual variables in ($DRMP$), we will show that the structure of the *Lagrangian costs* is precisely the same as that of the reduced costs of the previous section.

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 customers 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, (PBF) 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. A definite advantage of ($SEAF$), however, is that very few general-purpose solvers support the full-fledged B&P approach. Thus, using (PBF) 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.

5 Conclusion

Extensive experiments have shown that the third approach is by far the most effective, as it can determine very quickly optimal solutions by just running a general-purpose, off-the-shelf MILP on standard hardware. Unless for the largest, and possibly unrealistic, instances we tested, the optimal solution is often found at the root node by strengthening the bound via standard polyhedral techniques; this is a testament of the very good quality of the initial bound that the new formulation provides while requiring a much smaller number of variables than the previous ones.

The proposed (re)formulation makes it possible to efficiently solve even large instances. In turn, this allows to perform experiments in order to quantify the savings that can be obtained by convincing customers to take a more flexible stance about service times. From a managerial viewpoint, these outcomes can be useful to negotiate price discounts in exchange for the extended flexibility.

References

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