A Price-and-Branch algorithm for a drayage problem with heterogeneous trucks

Ali Ghezelsoflu, Massimo Di Francesco, Paola Zuddas [1]

Department of Mathematics and Computer Science, University of Cagliari, Cagliari, Italy

Antonio Frangioni [2]

Department of Computer Science, University of Pisa, Pisa, Italy

Abstract

This paper investigates a drayage problem, which is motivated by a carrier providing *door-to-door* freight transportation services by trucks and containers. The trucks carry one or two containers to ship container loads from a port to importers and from exporters to the same port. The problem is modeled by a set covering formulation with integer variables. We propose a Price-and-Branch algorithm for this problem, in which the pricing problem is a pair of shortest path problems in a suitable graph. The algorithm can determine near-optimal solutions in a short time.

Keywords: Price-and-Branch, Drayage, Vehicle Routing Problem, Set Covering.

1 Introduction

In this problem a carrier manages a fleet of trucks to serve two types of customer requests: the delivery of container loads from a port to importers and the shipment of container loads from exporters to the same port. Some trucks carry one container and can serve up to one importer and one exporter in a route. Other trucks can carry up to two containers and can serve up to two

[1] Email: `ali.nov7@gmail.com, mdifrance@unica.it, zuddas@unica.it`

[2] Email: `frangio@di.unipi.it`

importers and two exporters in a route. Customers must be serviced according to a *stay-with* policy, i.e. drivers wait for containers during packing and unpacking operations and trucks carry the same containers throughout their routes. Each truck performs only one route, routing costs between any pair of locations depends only on their distances and the truck type, as two-containers trucks have higher costs per unitary distance than one-container trucks.
This problem has several original characteristics. Trucks were usually supposed to carry one container at a time and the distribution of more-than-one container per truck has received attention only in recent papers (e.g., (4)). Moreover, in many drayage problems trucks are decoupled from containers at customer locations and trucks can perform different tasks during packing and unpacking operations (see (1) for a detailed literature literature). Few papers investigated both multiple containers per truck and *stay-with* operations as in our problem (2; 3). These papers proposed node-arc formulations, which have very poor linear relaxations. A set-covering formulation for this problem was proposed by (1), where all feasible routes were enumerated and an off-the-shelf optimization solver was run to optimally solve realistic-sized instances. However, it runs out-of-memory in the case of large-scale problems.
This paper proposes a Price-and-Branch algorithm for the formulation of (1). In the first step, the linear programming relaxation of the set-covering problem is solved by column generation, where the pricing problem is a shortest path problem in a proper graph. The resulting fractional optimal solution value is a tight lower bound on the value of the optimal integer solution. Then, from the set of columns generated so far, an integer solution is sought by Cplex. The computational results are compared to those in (1).
The paper is organized as follows. In Section 2, the set mathematical model is presented. In section 3 the Price-and-Branch algorithm is proposed. In Section 4, the results of the computational experiments are presented. Finally, conclusions and further research perspectives are described in Section 5.

2 Modeling

Let I be the set of importers, E the set of exporters and $V = I \cup E$ the set of customers, let R_1 and R_2 be the set of all feasible routes performed by one-container and two-container trucks, respectively, and $R = R_1 \cup R_2$ the set of all feasible routes. More precisely, R_1 and R_2 are the sets of routes leaving from the port, serving up to two and four customers respectively and moving back to the same port. If $n = \max\{\,|E|\,,\,|I|\,\}$, the number of all the possible routes is $O(n^4)$. Moreover, we denote by k_1 and k_2 the number of available

trucks for one-container and two-containers routes, respectively
For each route $r \in R$ and each customer $v \in V$, define the coefficient α_r^v such that $\alpha_r^v = 0$ if customer $v \in V$ is not visited in route $r \in R$, $\alpha_r^v = 1$ if customer $v \in V$ is visited in route $r \in R$ to deliver or pick up one container load, $\alpha_r^v = 2$ if customer $v \in V$ is visited in route $r \in R_2$ to deliver or pick up two container loads. Let d_v be the demand of each customer $v \in V$, i.e., the number of containers which must be used to service customer v. The decision variable x_r is defined as the number of times in which route $r \in R$ is performed, each time paying the corresponding unitary cost c_r. According to this notation, the problem can be formulated as follows:

$$
\begin{aligned}
Z_P = \min \quad & \textstyle\sum_{r \in R} c_r x_r & & (1)\\
s.t. \quad & \textstyle\sum_{r \in R} \alpha_r^v x_r \geq d_v & v \in V \quad & (2)\\
& \textstyle\sum_{r \in R_1} x_r \leq k_1 & & (3)\\
& \textstyle\sum_{r \in R_2} x_r \leq k_2 & & (4)\\
& x_r \in \mathbb{Z}_+ & r \in R \quad & (5)
\end{aligned}
$$

Routing costs are minimized in the objective function (1). Constraints (2) ensure that all customers are served. Constraints (3) and (4) enforce that the number of routes is lower than the number of available corresponding trucks. Finally, constraint (5) defines the domain of decision variables.

3 Price-and-Branch algorithm

To solve the problem effectively, a Price-and-Branch algorithm is proposed. A restricted set of all possible routes is enumerated and the linear relaxation with this partial route set (or Restricted Master Problem [RMP]) is solved. Let $B_1 \subseteq R_1$ and $B_2 \subseteq R_2$ be the subsets of routes with one-container and two-containers trucks, respectively, where $B = B_1 \cup B_2$ and $B \subseteq R$. The RMP is formulated as follows:

$$
\begin{aligned}
z_{RMP} = \min \quad & \textstyle\sum_{r \in B} c_r x_r & & (6)\\
s.t. \quad & \textstyle\sum_{r \in B} \alpha_r^v x_r \geq d_v & v \in V \quad & (7)\\
& \textstyle\sum_{r \in B_1} x_r \leq k_1 & & (8)\\
& \textstyle\sum_{r \in B_2} x_r \leq k_2 & & (9)\\
& x_r \in \mathbb{R}_+ & r \in B \quad & (10)
\end{aligned}
$$

The optimal solution of the RMP may be sub-optimal for the master problem with all feasible routes. However, the values of the optimal dual variables

of the RMP can be used to identify if there are any routes not included in the RMP that can further reduce the objective function value z_{RMP}. Let ξ_v, π, and ϖ multipliers of constraints (7), (8), and (9) for each $v \in V$. The dual of the RMP is denoted by DMP and is formulated as follows:

$$
\begin{aligned}
z_{DMP} = \max \quad & \textstyle\sum_{v \in V} \xi_v d_v - \pi k_1 - \varpi k_2 & & (11)\\
s.t. \quad & \textstyle\sum_{v \in V} \xi_v \alpha_r^v - \pi - \varpi \leq c_r & r \in B \quad & (12)\\
& \xi_v \in \mathbb{R}_+ & v \in V \quad & (13)\\
& \pi, \varpi \in \mathbb{R}_+ & & (14)
\end{aligned}
$$

Let x^* be the optimal solution of RPM and ξ^*, π^*, ϖ^* the optimal solution of DMP. Although this solution satisfies constraint (12), there may exist at least a route $r \in R \setminus B$ violating this constraint. If so, this route is added to B, the RMP is solved again to determine a better value of z_{RMP} and so on until no violated constraint is found.
Determining these violated constraints means looking for columns with negative reduced costs, where the reduced cost of route r is $\{c_r^* = c_r - (\sum_{v \in V} \xi_v^* \alpha_r^v - \pi^* - \varpi^*)\}$. These columns are determined in the pricing problem, where one minimizes c_r^*, such that capacity constraints are met.
The key-point of this Price-and-Branch algorithm is that the pricing problem can be formulated as a pair of shortest path problems on the following acyclic graph, one for each the truck type.

- In the case of one-container trucks, the port is split into two nodes p and p' and each customer $v \in V$ is modeled by one node. Arcs link p to any customer $v \in V$, any importer to any exporter, any customer $v \in V$ to p'. Each arc is associated with a unitary routing cost.
- In the case of two-container trucks, the port is again split into two nodes p and p' and each customer is modeled by two nodes $v' \in V'$ and $v'' \in V''$. Arcs link p to any node $v' \in V'$, any pair of nodes $v' \in V'$ and $v'' \in V''$ associated with each customer $v \in V$, any importer $i \in V''$ to any exporter $e \in V'$ and vice versa, any node $v' \in V'$ and $v'' \in V''$ to p'. Each arc connecting two different locations is associated with a unitary routing cost, whereas all arcs from $v' \in V'$ to $v'' \in V''$ have a null cost.

Fig.1 represents the acyclic graphs in the case of two importers denoted by 1 and 2, and three exporters denoted by 3, 4 and 5.

In the new graph, the pricing problem is a pair of two shortest path problems from p to p': the first determines a route for one-container trucks, the second a route for two-container trucks. If both solutions are negative, add

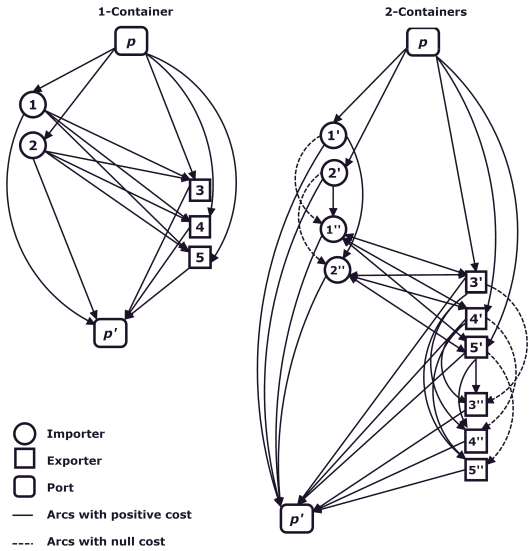

Fig. 1. Acyclic graph with five customers for one-container and two-containers trucks

both routes to the RMP. If only one of them is negative, add it to the RMP. The column generation stops when both routes determined in the pricing have a nonnegative reduced cost. If so, solve a RMP with the updated set B in which the decision variables are required to be integer.

4 Computational tests

Computational tests are carried out on some instances taken from (2). The RMP and the pricing problems in Section 3 has been solved by Cplex 12.5 on a Linux server with 3.00 Ghz processor, 16 GB of RAM. The restricted set of routes is initialized with all direct trips made by one-container and two-container trucks. The outcomes are reported in Tables 1. The following notation is used:

- it and $|B^*|$ are the number of iterations of the column generation algorithm and the number of columns of the RMP at the end of the column generation;
- t_{PB}, t_M, t_P and t_{lm} are times in seconds to solve the Price-and-Branch algorithm, the column generation algorithm, the linear relaxation of master problem at the end of the column generation, the integer master problem at the end of the column generation;
- τ_n and t_n are the preprocessing time in seconds to generate all $|R|$ feasible

routes for the set-covering model and the time in seconds to solve the set-covering model by Cplex, respectively;

- $Gap1$ is the relative gap between the upper bound determined by the integer RMP in Price-and-Branch algorithm and the optimal solutions of the set-covering formulation with all feasible routes;
- $Gap2$ is the relative gap between the optimal solutions of set-covering formulation and the linear relaxation of the RMP at the end of Price-and-Branch algorithm.

The results in Table 1 show the effectiveness of the Price-and-Branch algorithm. First, $|B^*|$ is significantly lower than $|R|$, thus it takes a short time to determine integer solutions from the RMP. Nevertheless, the solutions of the algorithm Price-and-Branch are near-optimal. Second, the time of the Price-and-Branch is a bit larger than sum of the overall time to enumerate all feasible routes and solve the set covering model in the case of all instances with 10 and 20 customers, but it is faster in almost of instances with 40 and 50 customers. Therefore, it is of interest to test if this trend holds event in the case of larger instances. They are taken from (1) and the related computational tests are reported in Table 2, where the same notation of Table 1 is adopted. The results in Table 2 show that the Price-and-Branch algorithm can determine near-optimal solutions for all the instances. The set covering with all feasible routes takes a much longer running time to solve instances H, I and J. Moreover, it does not return any feasible solution in the case of instances K, L, M, and N, as Cplex runs out-of-memory.

				Price-and-Branch						*Enumeration*				
$\|I\|$	$\|E\|$	k_1	k_2	it	$\|B^*\|$	t_{PB}	t_M	t_P	t_{lm}	R	τ_n	t_n	$Gap1$	$Gap2$
2	8	2	9	14	40	0.575	0.554	0.008	0.077	462	0.02	0.09	0.003	0.100
5	5	2	7	14	42	0.482	0.457	0.008	0.062	810	0.10	0.17	0.005	0.093
8	2	5	9	13	40	0.455	0.434	0.007	0.070	366	0.03	0.07	0.004	0.114
2	8	0	10	14	40	0.658	0.637	0.007	0.053	462	0.02	0.09	0.014	0.120
5	5	0	8	14	42	0.603	0.577	0.007	0.041	810	0.10	0.17	0.024	0.076
8	2	0	12	13	40	0.572	0.547	0.008	0.046	366	0.03	0.07	0.005	0.105
2	18	8	22	24	81	1.129	1.071	0.036	0.062	1792	0.05	0.27	0.006	0.066
5	15	7	19	30	86	1.383	1.284	0.049	0.159	7095	0.16	0.40	0.023	0.070
10	10	5	14	31	84	1.474	1.396	0.050	0.054	11220	0.34	0.28	0.016	0.066
15	5	7	19	28	83	1.544	1.412	0.047	0.216	6345	0.26	1.75	0.005	0.072
18	2	5	24	26	81	1.199	1.132	0.042	0.058	1716	0.05	0.31	0.009	0.052
2	18	0	26	24	81	1.115	1.052	0.035	0.081	1792	0.05	0.35	0.004	0.056
5	15	0	23	30	86	1.498	1.405	0.053	0.068	7095	0.16	0.46	0.006	0.041
10	10	0	17	31	84	2.403	2.331	0.048	1.186	11220	0.35	0.61	0.000	0.051
15	5	0	23	28	83	1.585	1.500	0.050	0.134	6345	0.26	0.94	0.014	0.063
18	2	0	27	26	81	1.442	1.377	0.040	0.184	1716	0.05	0.37	0.004	0.057
2	28	13	33	39	124	1.986	1.819	0.118	0.102	5578	0.29	0.26	0.002	0.069
5	25	12	30	42	124	2.117	1.966	0.115	0.137	19555	1.49	1.12	0.007	0.072
10	20	10	25	46	127	3.215	3.007	0.157	0.147	44730	1.72	4.39	0.028	0.045
15	15	8	19	54	143	3.542	3.336	0.152	0.313	54480	2.11	3.48	0.021	0.051
20	10	10	26	48	131	2.746	2.563	0.136	0.072	42730	2.15	4.27	0.002	0.070
25	5	12	32	38	122	1.788	1.655	0.100	0.065	17055	1.49	1.29	0.008	0.060
28	2	14	35	33	119	1.607	1.491	0.087	0.089	4122	0.83	0.28	0.004	0.059
2	28	0	40	39	124	2.192	2.043	0.117	0.187	5578	0.29	0.44	0.032	0.040
5	25	0	36	42	124	2.313	2.151	0.123	0.077	19555	1.49	3.08	0.030	0.049
10	20	0	30	46	127	2.521	2.297	0.168	0.216	44730	1.72	6.31	0.020	0.053
15	15	0	23	54	143	3.090	2.908	0.135	0.057	54480	2.11	6.72	0.014	0.058
20	10	0	31	48	131	3.201	3.004	0.153	0.052	42730	2.15	9.62	0.011	0.063
25	5	0	38	38	122	2.260	2.116	0.101	0.116	17055	1.38	4.10	0.025	0.043
28	2	0	42	33	119	1.870	1.749	0.093	0.079	4122	0.83	0.28	0.025	0.030
2	38	20	49	44	160	2.230	1.988	0.201	0.138	10228	0.09	1.68	0.067	0.031
5	35	18	45	49	161	2.578	2.293	0.237	0.107	38215	1.72	2.34	0.020	0.066
10	30	14	38	57	165	4.085	3.754	0.274	0.106	100340	4.19	5.37	0.047	0.032
15	25	12	31	60	171	4.481	4.011	0.351	0.144	151265	6.83	7.11	0.034	0.048
20	20	12	29	69	179	4.448	3.998	0.377	0.098	168840	8.35	18.4	0.030	0.062
25	15	14	36	59	160	3.855	3.533	0.233	0.062	147515	6.39	7.47	0.029	0.072
30	10	17	43	54	164	3.142	2.859	0.228	0.111	94340	4.01	5.27	0.028	0.061
35	5	19	48	45	159	3.289	3.026	0.221	0.131	32965	1.51	2.12	0.028	0.061
38	2	20	51	45	162	2.462	2.228	0.188	0.208	7492	0.09	0.81	0.023	0.062
2	38	0	59	44	160	2.964	2.720	0.200	0.187	10228	0.09	5.38	0.069	0.019
5	35	0	54	49	161	2.982	2.712	0.225	0.291	38215	1.72	3.41	0.070	0.025
10	30	0	45	57	165	4.588	4.236	0.297	0.167	100340	4.19	8.35	0.089	0.020
15	25	0	37	60	171	5.718	5.255	0.390	0.498	151265	6.83	7.89	0.081	0.021
20	20	0	35	69	179	5.252	4.794	0.388	0.126	168840	8.35	18.5	0.070	0.022
25	15	0	43	59	165	4.451	4.018	0.367	0.605	147515	6.39	7.52	0.068	0.021
30	10	0	51	54	164	3.384	3.112	0.226	0.105	94340	4.01	4.37	0.076	0.023
35	5	0	58	45	159	2.850	2.597	0.207	0.132	32965	1.51	1.68	0.070	0.018
38	2	0	61	45	162	2.669	2.402	0.189	0.188	7492	0.09	1.14	0.066	0.019
2	48	22	56	56	203	3.250	2.757	0.437	0.232	16278	1.02	0.56	0.024	0.070
5	45	21	54	62	202	3.796	3.286	0.449	0.307	62850	4.19	1.67	0.033	0.069
10	40	18	50	70	202	5.620	4.957	0.588	0.100	177750	11.0	8.01	0.072	0.034
15	35	17	42	70	201	5.150	4.411	0.586	1.522	295500	17.9	14.0	0.073	0.038
20	30	13	37	80	210	7.013	6.250	0.671	0.084	379350	23.1	23.2	0.073	0.038
25	25	11	32	77	218	4.702	4.133	0.492	0.094	407550	24.7	16.2	0.071	0.035
30	20	12	32	78	219	4.881	4.334	0.473	0.079	373350	22.2	16.2	0.094	0.034
35	15	15	39	73	206	4.453	3.936	0.451	0.183	285000	17.4	13.0	0.091	0.031
40	10	17	46	73	211	4.537	4.010	0.461	0.109	165750	9.96	4.61	0.085	0.039
45	5	20	50	62	205	3.872	3.389	0.405	0.123	53850	3.39	2.01	0.064	0.035
48	2	22	55	55	201	3.393	2.955	0.381	0.150	11862	0.73	1.14	0.049	0.035
2	48	0	67	56	203	3.941	3.464	0.428	0.145	16278	1.02	1.17	0.088	0.015
5	45	0	65	62	202	4.072	3.607	0.395	0.151	62850	4.19	2.71	0.095	0.015
10	40	0	59	70	202	5.599	4.970	0.540	0.266	177750	11.0	5.11	0.099	0.015
15	35	0	51	70	201	7.258	6.335	0.709	1.431	295500	17.9	14.9	0.100	0.020
20	30	0	44	80	210	7.990	7.150	0.737	0.155	379350	23.1	13.2	0.096	0.024
25	25	0	38	77	218	5.144	4.591	0.480	0.073	407550	24.7	17.4	0.094	0.021
30	20	0	38	78	219	5.413	4.870	0.470	0.084	373350	22.2	16.7	0.122	0.015
35	15	0	47	73	206	4.856	4.368	0.420	0.259	285000	17.4	7.92	0.122	0.019
40	10	0	55	73	211	4.931	4.439	0.430	0.121	165750	9.96	3.81	0.118	0.014
45	5	0	60	62	205	4.490	4.046	0.383	0.166	53850	3.39	1.48	0.098	0.020
48	2	0	66	55	201	3.803	3.346	0.399	0.142	11862	0.73	1.23	0.085	0.019

Table 1

Comparison on realistic-sized instances between the Price-and-Branch algorithm and the set-covering formulation enumerating all feasible routes (1)

			Price-and-Branch						*Enumeration*				
	$\|I\|$	$\|E\|$	it	$\|B^*\|$	t_{PB}	t_M	t_P	t_{lm}	R	τ_n	t_n	$Gap1$	$Gap2$
A	20	5	28	59	1.149	1.093	0.043	0.133	1.1e+4	1.01	0.22	0.00025	0.00036
B	20	10	47	93	2.051	1.907	0.112	0.059	4.3e+4	1.94	1.19	0.00061	0.00045
C	20	20	62	136	2.971	2.691	0.236	0.048	1.7e+5	8.98	4.00	0.00026	0.00024
D	30	8	46	94	2.060	1.877	0.155	0.245	6.1e+4	5.03	9.09	0.00021	0.00024
E	30	15	67	132	3.449	3.084	0.319	0.130	2.1e+5	6.83	13.3	0.00041	0.00028
F	30	30	103	206	6.028	5.139	0.811	0.059	8.4e+5	15.7	20.3	0.00084	0.00091
G	45	12	67	136	3.439	2.900	0.488	0.160	3.0e+5	8.65	16.3	0.00034	0.00017
H	45	23	95	194	6.283	5.292	0.922	0.805	1.1e+6	13.06	2455.7	0.00029	0.00019
I	45	45	148	300	11.73	9.184	2.434	0.092	4.2e+6	30.48	782.9	0.00037	0.00024
J	75	19	115	230	8.137	6.025	2.028	0.642	2.1e+6	18.49	5560.3	0.00034	0.00023
K	75	38	177	340	16.02	11.46	4.422	0.138	8.2e+6	49.25	n.s.	-	-
L	75	75	275	556	34.55	22.34	11.86	0.115	3.2e+7	157.29	n.s.	-	-
M	100	25	162	313	15.04	10.06	4.837	0.619	6.3e+6	38.14	n.s.	-	-
N	100	50	215	425	24.53	15.38	8.940	0.988	2.5e+7	107.32	n.s.	-	-

Table 2
Comparison on larger instances

5 Conclusion

In this paper we have investigated a drayage problem faced by a carrier. We have proposed a Price-and-Branch algorithm, in which a pair of shortest path problems is solved at each iteration of the pricing problem. The algorithm can determine near-optimal solutions in short time intervals and is very useful to solve large instances efficiently. An extension of this research will be the development of a Branch-and-Price algorithm to determine optimal solutions.

References

[1] Ghezelsoflu, A. and Di Francesco, M. and Frangioni, A. and Zuddas, P., *A Set-Covering formulation for a drayage problem with single and double container loads*, Journal of Industrial Engineering International, DOI (2018), 10.1007/s40092-018-0256-8.

[2] Lai, M. and Crainic, T. G. and Di Francesco, M. and Zuddas, P., *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.

[3] Lai, M. and Battarra, M. and Di Francesco, M. and Zuddas, P., *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.

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