Column generation for the truck and trailer routing problem with time windows

Sophie N. Parragh\textsuperscript{1}, Jean-François Cordeau\textsuperscript{2}

\textsuperscript{1} Institute for Business Administration; University of Vienna; Oskar-Morgenstern-Platz 1; 1090 Vienna; Austria
sophie.parragh@univie.ac.at

\textsuperscript{2} Canada Research Chair in Logistics and Transportation and CIRRELT; HEC Montréal; 3000, chemin de la Côte-Sainte-Catherine; Montréal H3T 2A7; Canada
jean-francois.cordeau@hec.ca

Keywords: column generation, truck and trailer routing, time windows

1 Introduction

Motivated by the field staff routing and scheduling problem of an Austrian infrastructure service provider, which involves the planning of subroutes, we study the truck and trailer routing problem (TTRP) \cite{lin2015}. In the TTRP, two types of customers are considered: customers that can be visited by a truck pulling a trailer (denoted as vehicle customers) and customers that can only be visited by a truck alone (denoted as truck customers). Each customer has a given demand. In order to serve the customers, three different kinds of routes can be planned: routes that are carried out by a single truck (denoted as truck routes); routes that are carried out by a truck pulling a trailer (denoted as vehicle routes) without any intermediate decoupling; and routes carried out by a truck pulling a trailer involving truck-only subroutes. Each such subroute starts and ends at a vehicle customer. At this customer the trailer is decoupled and then re-coupled again at the end of the subroute. The truck has a capacity of $Q_{\text{truck}}$ and the trailer a capacity of $Q_{\text{trailer}}$. Shifting loads from the trailer to the truck before departing on the next truck-only subroute is possible. The objective is to minimize the total routing cost which corresponds to the total distance traveled by the available vehicle fleet. An important aspect is that a trailer may not be picked up by a truck different from the one which decoupled it. Furthermore, each customer may only be visited once. This implies that a decoupling point cannot be used more than once: whenever a vehicle customer is used to park the trailer, it has to be served. However, several consecutive subroutes may start and end at the same vehicle customer.

The truck and trailer routing problem with time windows (TTRPTW) has first been stated and solved by Lin et al. \cite{lin2015}. However, they make the restrictive assumption that a vehicle customer where the trailer is decoupled has to be visited before the first truck alone subroute starts. Derigs et al. \cite{derigs2015} do not make this restrictive assumption, i.e. the customer used to park the trailer may be visited before or after the subroute. They show that this leads to considerable improvements in terms of solution quality. This is the problem version considered in this paper.

Truck and trailer routing problems belong to the class of two-echelon vehicle routing problems which has recently been surveyed by Cuda et al. \cite{cuda2015}. To the best of our knowledge only two contri-
butions exist in which exact algorithms are designed for variants of the TTRP: Belenguer et al. [1] develop a branch-and-cut algorithm for the single TTRP with satellite depots and Drexl [5] proposes a branch-and-price algorithm for a generalized TTRP in which additional decoupling points are considered. In this paper, we formulate the TTRPTW in terms of a path-based model and we solve its linear relaxation by means of column generation, using a different methodology than Drexl [5].

2 Problem formulation

In order to formulate the TTRPTW, we denote by \( N \) the set of customers, by \( K_{\text{truck}} \) the set of available trucks, by \( K_{\text{trailer}} \) the set of available trailers, by \( \Omega \) the set of all feasible routes and by \( c_r \) the total routing cost of route \( r \). Since, especially when working with real world data, it cannot always be guaranteed that a feasible solution serving all customers with the available vehicle fleet exists, we also consider outsourcing costs \( o_i \) for each customer \( i \). Finally, we use parameter \( a_r \) to indicate whether a trailer is used by route \( r \) and \( b_{ir} \) if a customer \( i \) is visited by route \( r \). Using this notation and the following two types of binary decision variables,

\[
\begin{align*}
    u_r &= \begin{cases} 
    1, & \text{if route } r \text{ is used,} \\
    0, & \text{otherwise,}
    \end{cases} \\
    z_i &= \begin{cases} 
    1, & \text{if customer } i \text{ is outsourced,} \\
    0, & \text{otherwise,}
    \end{cases}
\end{align*}
\]

we formulate the TTRPTW in terms of a path-based model:

\[
\min \sum_{r \in \Omega} c_r u_r + o_i z_i 
\]

subject to:

\[
\begin{align*}
    \sum_{r \in \Omega} b_{ir} u_r + z_i &= 1 & \forall i \in N \quad (2) \\
    \sum_{r \in \Omega} u_r &\leq |K_{\text{truck}}| & \forall r \in \Omega \quad (3) \\
    \sum_{r \in \Omega} a_r u_r &\leq |K_{\text{trailer}}| & \forall r \in \Omega \quad (4) \\
    u_r &\in \{0,1\} & \forall r \in \Omega \quad (5) \\
    z_i &\in \{0,1\} & \forall i \in N \quad (6)
\end{align*}
\]

The objective function (1) minimizes the total routing and outsourcing costs. Constraints (2) make sure that each customer is either visited or outsourced and inequalities (3) and (4) guarantee that at most \( |K_{\text{truck}}| \) trucks and \( |K_{\text{trailer}}| \) trailers are used. Relaxing the integrality requirements, we obtain a linear program that can be solved by means of column generation.

3 Column generation

The subproblem we have to solve in order to populate the column pool corresponds to an (elementary) shortest path problem with resource constraints (the elementary property does not hold for vehicle customer nodes that serve as decoupling points). It is solved by means of a labeling algorithm. In each call to the labeling algorithm, in a first step, two labels at the start depot are generated: one for a route with a trailer and one for a route without. Then, in each iteration, a label is considered
for extension. If it is not dominated by any other existing label at the same node $i$, it is extended along each arc $(i, j)$ that results in a feasible partial path. If $j$ is a truck customer, only one new label is generated at $j$. However, if $j$ is a vehicle customer up to three new labels are generated. At a vehicle customer, a trailer may be decoupled and re-coupled and both may be combined with serving or not serving $j$, depending on the partial route leading to $j$; or no de- or re-coupling may be performed. Our labeling algorithm terminates as soon as a given number of reduced cost labels have been generated at the end depot. If no additional columns of reduced cost can be found, column generation terminates and the optimal solution to the linear relaxation of model (1)–(6) has been found.

In order to speed up our labeling algorithm, we use several enhancements. The first enhancement is based on the notion that, given the triangle inequality holds for the cost matrix, in an optimal solution, no subroute exists that contains only vehicle customers. This observation is exploited in the label extension step.

The second enhancement is the following. Instead of considering all types of routes from the beginning, in the first phase of our labeling algorithm, we only try to find columns that correspond to truck routes. If we fail to find any of negative reduced cost, in a second phase we try to find columns of negative reduced cost that correspond to vehicle routes, without considering the generation of subroutes. Only if this fails as well we try to generate columns that correspond to vehicle routes with subroutes.

As a third enhancement, we use heuristic pricing schemes. The first heuristic pricing scheme (HP1) is based on our labeling algorithm, using ideas from [7]. However, we do not use a two-stage pricing scheme but we pre-compute a set of potential subroutes for each of the vehicle customers. Each set is generated as follows. In a first step, for each vehicle customer, we identify the ten closest customers. If at least one of these customers is a truck customer, the respective vehicle customer is considered as a potential starting point for a subroute. Then, for each potential starting point, we enumerate all sets of size two to six of the ten closest customers, such that each set contains at least one truck customer. For each of these computed sets we generate all feasible routes visiting all customers in the set and we only keep non-dominated ones. Then, whenever the pricing algorithm is called, we go through all pre-computed subroutes once and check if they have negative reduced cost. If they do, they are considered in the pricing heuristic. In addition, we assume that in each vehicle route, only one vehicle customer can be used as a decoupling point and at most three times consecutively (i.e. at most three consecutive subroutes may be appended).

The second heuristic pricing scheme (HP2) uses ideas from large neighborhood search [8]. We retrieve all routes of the current basis and in a destroy step 40% of these routes are selected randomly. Considering all customers served by any of the selected routes (but at max 18), we generate a new restricted TTRPTW instance that is solved by means of column generation in a repair step. All generated routes of negative reduced cost for the original problem are put into the column pool. Destroy and repair steps are iterated 10 times.

Finally, as a fourth enhancement, we generate initial columns by means of a large neighborhood search algorithm tailored to the TTRPTW.

4 First results and next steps

We applied our column generation algorithm to instances from the literature with 50 customers. These instances were proposed by Lin et al. [6]. They are based on the first instance of each data set of the well known Solomon instances [9] for the vehicle routing problem with time windows. In
Table 1: Number of instances solved with 25 customers with the different column generation schemes

<table>
<thead>
<tr>
<th></th>
<th>Standard pricing</th>
<th>3-Phase pricing (3-P.)</th>
<th>3-P.+HP1+initLNS</th>
<th>3-P.+HP2+initLNS</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>R1</td>
<td>29</td>
<td>32</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>RC1</td>
<td>11</td>
<td>12</td>
<td>14</td>
<td>14</td>
</tr>
</tbody>
</table>

order to obtain a larger test bed, we generated additional instances taking each Solomon instance from the sets C1, R1 and RC1 as a basis, considering 25 customers of which 25%, 50%, and 75% are truck customers. Our experiments show that the proposed enhancements help to decrease the run times and thus to solve additional instances (see Table 1). They also indicate that instances with clustered customer locations are harder to solve than those with randomly spread ones. We currently solve some of the instances with 50 customers. In a next step, further enhancements to the pricing mechanism will be implemented and tested. Then, the proposed column generation algorithm will be embedded into a branch and bound tree. The goal is to solve small to medium-sized instances to optimality and to derive a heuristic algorithm for the solution of larger instances.

Acknowledgments This research is supported by the Austrian Science Fund (FWF): T514-N13. This support is gratefully acknowledged.

References


