ABSTRACT BOOKLET

ROUTE 2018

International Workshop on Vehicle Routing, Intermodal Transportation and Related Areas

May 27-30, 2018
WORKSHOP CO-SPONSORS

- Logis A/S
- Transport Innovation Network
- DTU Management Engineering, Management Science Division
WELCOME

I would like to welcome you to ROUTE 2018. As in 2014, ROUTE 2018 will take place at Comwell Borupgaard, Snekkersten, Denmark. The workshop will run from May 27 to 30, 2018.

Like previous editions, ROUTE 2018 aims to provide a forum for scientific exchange and cooperation in the fields of vehicle routing, intermodal transportation and related areas. This year we have a set of 35 presentations. To fit the single stream format, two “after dinner” sessions have been introduced on Sunday and Monday.

I would like to thank Rasmus Vilrik Bruun, Christina Scheel Persson and Kristoffer Nimb for their administrative support, and Oli Madsen, the founding father of the ROUTE series, for his advice and assistance.

I would also like to thank Logis A/S, the Transport Innovation Network and the DTU Department of Management Engineering, Management Science Division, for being co-sponsors of this event.

I look forward to an excellent workshop.

Harilaos N. Psaraftis

PS The titles of the sessions on Wednesday will be explained by the corresponding chairs.
## SCHEDULE AT A GLANCE

**Note:** Only the speakers’ names are shown. For all co-authors see abstracts.

### SUNDAY MAY 27

<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>1300</td>
<td>REGISTRATION - COFFEE</td>
</tr>
<tr>
<td>1345</td>
<td>WELCOMING REMARKS</td>
</tr>
<tr>
<td>1400</td>
<td>Exact approaches (chair: Jesper Larsen)</td>
</tr>
<tr>
<td>1430</td>
<td>Roberto Roberti: An Exact Method for the Consistent Vehicle-Routing Problem</td>
</tr>
<tr>
<td>1500</td>
<td>Magnus Stålhane: An exact solution method for the shortest path problem with speed optimization</td>
</tr>
<tr>
<td>1530</td>
<td>Roberto Uchoa: Towards General Exact Vehicle Routing Problem Solvers</td>
</tr>
<tr>
<td>1600</td>
<td>Dynamic/stochastic problems (chair: Gilbert Laporte)</td>
</tr>
<tr>
<td>1630</td>
<td>Guy Desaulniers: The vehicle routing problem with stochastic and correlated travel times</td>
</tr>
<tr>
<td>1700</td>
<td>Michel Gendreau: An Exact Algorithm to Solve the Vehicle Routing Problem with Stochastic Demands under an Optimal Restocking Policy</td>
</tr>
<tr>
<td>1730</td>
<td>Janny Leung: Real-time Re-scheduling for Public Transit</td>
</tr>
<tr>
<td>1830</td>
<td>DINNER</td>
</tr>
</tbody>
</table>

### MONDAY MAY 28

<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>0830</td>
<td>Maritime logistics I (chair: Pitu Mirchandani)</td>
</tr>
<tr>
<td>0900</td>
<td>Kjetil Fagerholt: Combining optimization and simulation for designing short sea feeder networks with transshipment at sea</td>
</tr>
<tr>
<td>0930</td>
<td>Irina Gribkovskaya: Flexible and robust supply vessels routing and scheduling</td>
</tr>
<tr>
<td>1000</td>
<td>Costa Koza: Integrated Liner Shipping Network Design and Scheduling, or: Solving a large service network design problem</td>
</tr>
<tr>
<td>1030</td>
<td>Electric vehicles + refueling (chair: Guy Desaulniers)</td>
</tr>
<tr>
<td>1100</td>
<td>Gilbert Laporte: Goods Distribution with Electric Vehicles</td>
</tr>
<tr>
<td>1130</td>
<td>Pitu Mirchandani: Routing and Detouring Consideration in Location of Refueling Points on a Network</td>
</tr>
<tr>
<td>1200</td>
<td>Lunch</td>
</tr>
<tr>
<td>1230</td>
<td>Air and combined transportation (chair: Janny Leung)</td>
</tr>
<tr>
<td>1400</td>
<td>Jan Ehnike: The Most Reliable Flight Itinerary Problem</td>
</tr>
<tr>
<td>1430</td>
<td>Martin Kidd: The value of integration in supply chain planning</td>
</tr>
<tr>
<td>1500</td>
<td>Frederic Semet: The last mile delivery problem</td>
</tr>
<tr>
<td>1530</td>
<td>COFFEE</td>
</tr>
<tr>
<td>1550</td>
<td>Heuristic approaches (chair: Stefan Røpke)</td>
</tr>
<tr>
<td>1600</td>
<td>Maria Battara: Hub Location and Arc Pricing Problem</td>
</tr>
<tr>
<td>1630</td>
<td>Andrea Mor: Heuristics for the traveling salesman problem with release dates and completion time minimization</td>
</tr>
<tr>
<td>1700</td>
<td>Thibaut Vidal: Heuristics for vehicle routing problems: Sequence or set optimization?</td>
</tr>
<tr>
<td>1830</td>
<td>DINNER</td>
</tr>
</tbody>
</table>

### TUESDAY MAY 29

<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>0830</td>
<td>Maritime logistics II (chair: Irina Gribkovskaya)</td>
</tr>
<tr>
<td>0900</td>
<td>Jesper Larsen: Tramp ship routing and scheduling with voyage separation requirements</td>
</tr>
<tr>
<td>0930</td>
<td>Rafael Martinelli: Ship routing and speed optimization with heterogeneous fuel consumption profiles</td>
</tr>
<tr>
<td>1000</td>
<td>Harilaos Psaraftis: The profit maximizing liner shipping problem with flexible frequencies</td>
</tr>
<tr>
<td>1130</td>
<td>Lunch</td>
</tr>
<tr>
<td>1230</td>
<td>City logistics (chair: Jacques Desrosiers)</td>
</tr>
<tr>
<td>1100</td>
<td>Evelien van den Hurk: Demand responsive public transport scheduling: Dial-a-ride versus Pick up and Delivery with Time Windows</td>
</tr>
<tr>
<td>1130</td>
<td>Tal Raviv: Routing of small parcels in a crowd-sourced network</td>
</tr>
<tr>
<td>1130</td>
<td>Martin Andersen: Emergency Medical Services – A practical application of the Vehicle Routing Problem</td>
</tr>
<tr>
<td>1200</td>
<td>Lunch</td>
</tr>
<tr>
<td>1230</td>
<td>EXCURSION TO LOUISIANA MUSEUM</td>
</tr>
<tr>
<td>1900</td>
<td>BANQUET</td>
</tr>
</tbody>
</table>

### WEDNESDAY MAY 30

<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>0830</td>
<td>Potage Touski (chair: Michel Gendreau)</td>
</tr>
<tr>
<td>0900</td>
<td>Dominique Fretil: Time-Dependent Vehicle Routing Problem with Time Windows and road network information</td>
</tr>
<tr>
<td>0930</td>
<td>Richard Hartli: Routing problems, multi-commodity fixed-charge network and variable splitting</td>
</tr>
<tr>
<td>1000</td>
<td>Marcus Poggi: A Benders Decomposition for the Inventory Routing Problem</td>
</tr>
<tr>
<td>1130</td>
<td>Lunch</td>
</tr>
<tr>
<td>1230</td>
<td>Kakavia (chair: Harilaos Psaraftis)</td>
</tr>
<tr>
<td>1230</td>
<td>LUNCH</td>
</tr>
<tr>
<td>1230</td>
<td>DEPARTURE</td>
</tr>
</tbody>
</table>
Vehicle-routing problems (VRPs) with consistency considerations have received substantial interest in recent years because of the practical importance of providing consistent service in many industries, such as, e.g., small package shipping, health care, or vendor-managed inventory systems (for a survey, see Kovacs et al. (2014a)). To boost customer satisfaction, customers should be served at roughly the same time (arrival time consistency, ATC) by the same driver (driver consistency, DC), or at least a small set of familiar drivers, each time they require service. Taking the drivers perspective, serving the same customers repeatedly makes the driver familiar with the geographic region and the characteristics of the customer, and thus more efficient in fulfilling his tasks.

The most prominent variant of the class of VRPs with consistency considerations is the Consistent VRP (ConVRP), introduced by Groer et al. (2009). The ConVRP is a multi-day VRP requiring that, in addition to the traditional constraints on vehicle capacity and route duration, the same driver serves the same customers at approximately the same time on each day that these customers require service, given by a maximum allowed difference between the arrival times on the different days. Originally, the problem is motivated from the delivery and collection operations at United Parcel Services, where strong emphasis is put on customer and employee satisfaction.

In the academic literature, the ConVRP has received adequate attention from the heuristic side (see Groer et al. (2009), Sungur et al. (2010), Tarantilis et al. (2012), Kovacs et al. (2014b)). On the contrary, to the best of our knowledge, no exact approach to the ConVRP has been proposed yet. The only two papers addressing consistency considerations in an exact fashion are owed to Subramanyam and Gounaris (2016) and Subramanyam and Gounaris (2017), who study the consistent traveling-salesman problem, i.e., only one route per day is planned and routes must adhere to the ATC requirements.

Most of the state-of-the-art exact methods to solve VRPs are based on column generation applied to formulations where each variable represents a feasible route, and the pricing problem is solved via dynamic programming. However, these methods cannot be directly extended to solve the ConVRP because the linear relaxation of route-based formulations provides weak lower bounds due to the interdependency between the daily routes, which is caused by the required ATC at customers.

In this talk, we propose an exact method based on column generation applied to a formulation in which each variable represents the set of routes assigned to a vehicle over the planning horizon. The exact method initially takes into account DC only, and addresses ATC at a later stage. Computational results show that the proposed exact method is able to solve small and medium sized instances with up to five planning periods and 30 customers.

1Speaker
References


An exact solution method for the shortest path problem with speed optimization

Magnus Stålhane, Marielle Christiansen, Troels Range
Department of Industrial Economics and Technology Management
Norwegian University of Science and Technology, Norway
December 15, 2017

1 Introduction

We present an exact solution method for the shortest path problem with resource constraints and speed optimization (SPPRC-SO). The problem appears as a subproblem when solving a number of different vehicle routing problems with speed optimization using branch-and-price. To the authors knowledge exact solution methods to this problem has not been studied in the literature.

The problem is defined on a graph $G = (\mathcal{N}, \mathcal{A})$, where $\mathcal{N} = \{0, \ldots, n+1\}$ is the set of nodes, and $\mathcal{A} = \{(i, j) \mid i, j \in \mathcal{N} \land i \neq j\}$ is the set of arcs connecting the nodes. Node 0 is the start node, node $n + 1$ is the end-node. For all other nodes $i$ we associate a revenue $R_i$ if it is serviced, and a time window $[T_i, T_i]$ which gives an earliest and a latest time for starting service, and a service time $T_i$. For each arc $(i, j) \in \mathcal{A}$ there is a corresponding non-negative distance $D_{ij}$. The vehicle has a range of feasible speeds given by the interval $[\underline{V}, \overline{V}]$, and a corresponding cost function $c(v)$ that gives the cost of moving one distance unit at a given speed.

Let $x_{ij}$ be equal to one if the vehicle traverses the arc from node $i$ to node $j$, and zero otherwise, and $v_{ij}$ be the speed of the vehicle on the corresponding arc. Further, let $t_i$ be the time the vehicle arrives at node $i$. Using the notation described above the problem can be formulated as follows:

$$\min \sum_{(i,j) \in \mathcal{A}} (D_{ij}c(v_{ij}) - R_i)x_{ij},$$  \hspace{1cm} (1)
subject to:

\[ \sum_{j \in \mathcal{N}} x_{0j} = 1, \quad (2) \]

\[ \sum_{j \in \mathcal{N}} x_{ji} - \sum_{j \in \mathcal{N}} x_{ij} = 0, \quad i \in \mathcal{N} \setminus \{0, n + 1\}, \quad (3) \]

\[ \sum_{i \in \mathcal{N}} x_{i(n+1)} = 1, \quad (4) \]

\[ (t_i + T_i + \frac{D_{ij}}{v_{ij}} - t_j)x_{ij} \leq 0, \quad (i, j) \in \mathcal{A}, \quad (5) \]

\[ T_i \leq t_i \leq T_i, \quad i \in \mathcal{N}, \quad (6) \]

\[ V \leq v_{ij} \leq V, \quad (i, j) \in \mathcal{A}, \quad (7) \]

\[ x_{ij} \in \{0, 1\}, \quad (i, j) \in \mathcal{A}. \quad (8) \]

The objective function (1) minimizes the total cost (or negative profit) of the vehicle route, while constraints (2) – (4) force each vehicle to travel a continuous path through the network from its origin node to its destination node. Further, constraints (5) update the arrival time at a node, given the start time at the previous node along the route and the speed kept by the vehicle on the arc between the two nodes. Finally, constraints (6) ensure that all nodes are visited within their time windows, constraints (7) keep the vehicle speed within the feasible range, and constraints (8) put binary restrictions on the \(x\)-variables.

### 2 Solution method

To solve the problem we have used a labeling algorithm, based on the description given by Irnich and Desaulniers (2005). For the SPPRC-SO the challenge of applying this approach is to find a valid and efficient dominance step. Both the time and cost resource is dependent on the speed on each arc traversed on a (partial) path.

For a given arrival time \(t\) at the last node it is possible to calculate the exact cost of a (partial) path using the recursive smoothing algorithm described in Hvattum et al. (2013). However, if we have to compare two paths for all possible points in time then the time dependent part becomes challenging, as the dominance criteria must compare functions rather than values.

We present a sufficient dominance criteria for the SPPRC-SO by, which handles the time-cost dependencies by approximating the cost function for given points in time, and then uses the relation between the cost values and points to discard dominated paths in the labeling algorithm.
3 Preliminary results

Table 1 shows some preliminary results from testing our methodology on the Solomon instances (Solomon, 1987), using branch-and-price. All instances were run with an upper limit of 2 hours (7200 seconds) of computing time. The table compares solving our implementation with a constant speed of one, and with speed ranges from $[0.9, 1.1]$ and $[0.75, 1.25]$, respectively, using a convex cost function. For each speed range we give the number of instances solved, the average solution time of the solved instances, and for the latter two we also give the average improvement in the objective value compared to the constant speed solution.

Table 1: Results of testing our methodology on the solomon instances

<table>
<thead>
<tr>
<th># customers</th>
<th>$v_{ij} = 1$</th>
<th>$v_{ij} \in [0.9, 1.1]$</th>
<th>$v_{ij} \in [0.75, 1.25]$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># solved</td>
<td>avg. Time</td>
<td># solved</td>
</tr>
<tr>
<td>25</td>
<td>54</td>
<td>321</td>
<td>49</td>
</tr>
<tr>
<td>50</td>
<td>31</td>
<td>1121</td>
<td>24</td>
</tr>
<tr>
<td>100</td>
<td>8</td>
<td>637</td>
<td>6</td>
</tr>
</tbody>
</table>

The results show that we are able to solve most of the 25 customer instances, about one third of the 50 customer instances, and a few of the 100 customer instances within 2 hours. We would like to point out that the branch-and-price algorithm is a basic implementation without any of the state of the art techniques used to solve the largest solomon instances. The obtained average improvement show that there are significant savings that can be obtained from optimizing speeds, and since the improvement is, on average, larger than the possible speed reduction, the improvement cannot come only from slowing down the vehicles, but must come from improved routing decisions.

References


Towards General Exact Vehicle Routing Problem Solvers

Artur Pessoa
Universidade Federal Fluminense

Ruslan Sadykov
INRIA Bordeaux – Sud-Ouest

Eduardo Uchoa
Universidade Federal Fluminense

François Vanderbeck
University of Bordeaux

In recent years the community has accomplished significant advances in its ability to solve Vehicle Routing Problems (VRPs) to optimality. A milestone in these breakthroughs was the Branch-Cut-and-Price (BCP) algorithm of [9, 10]: it solved CVRP instances with up to 360 customers, a major improvement upon the previous record of 150 customers. The algorithm exploits many elements from previous works, usually enhancing them and combining them with new features.

- It introduces the concept of limited memory cuts as a key original element. The idea is to control and dynamically adjust the weakening of the Subset Row Cuts (SRCs) [7], conserving in good part their very effective impact on the lower bound improvement while inducing a much lower impact in the pricing complexity.

- The underlying formulation is extended to include arc-load variables. The latter allow an effective scheme for fixing of variables by reduced costs, that improves performances compared to the reduced cost fixing scheme of [6].

- The columns are defined by ng-routes [2]. The pricing subproblem is solved using a bidirectional labeling algorithm [15] with a new balancing mechanism; it also exploits completion bounds [4]; furthermore, effective heuristics are put to play.

- In some instances, column generation is prone to severe convergence problems. When this is detected, a dual stabilization by an automated adjusted smoothing scheme [14] is performed.

- The BCP hybridizes branching with route enumeration [1]. Actually, it performs an aggressive hierarchical strong branching, improving over the scheme of [16]. The mechanism includes tree size evaluation and keeps a history of all previous branches.

- If the node gap is sufficiently small, the elementary routes that can be part of the optimal solution (up to 20M routes) are enumerated into a pool, as in [4]. Once the node is in this enumerated mode, pricing is performed by inspection. Hence, general clique and odd hole cuts can be separated without harm to the pricing. Fixing by reduced cost is performed in the pool. If the number of routes in the pool decreases to less than 20k, the BPC process switches to a call to a MIP solver on the full static formulation. However, differently from [4], traditional branching may be performed in enumerated mode.

- The lm-SRCs are non-robust cuts. The algorithm may reach a stage where several hundreds such cuts are well handled in the pricing oracle; then, the separation of a few dozen additional lm-SRCs yields a pricing procedure that is 100 times slower. In this situation the BCP performs a rollback, removing those cuts, then proceeds by branching.

Since then, those elements are being enhanced and adapted for creating powerful BCP algorithms for other classical VRP variants.

- The lm-SRCs were generalized to limited memory Rank-1 Cuts (lm-R1Cs) in [11, 3].
The original concept of limited memory on nodes [9] was sharpened to limited memory on arcs in [8], an improvement that was shown to be important for solving harder VRP with Time Windows (VRPTW) instances. A further sharpening in the context of Heterogeneous Fleet VRP (HFVRP) was the concept of limited memory (on nodes or arcs) depending on vehicle type [13].

The concept of balanced bi-directional label correcting algorithm created in [9] for the CVRP was shown to be effective in other variants in [18].

A new implementation of the bi-directional label correcting algorithm in which the labels are stored and extended according to so-called bucket graph is proposed in [17]. Such organization of labels helps to decrease significantly the number of dominance checks and the running time of the pricing oracle.

Overall, the ability to solve VRP problems exactly has significantly improved for the following classical variants: VRP with Time Windows [8, 17], Heterogeneous Fleet VRP and Multi-Depot VRP [13, 17], Distance-Constrained VRP [17] and Capacitated Arc Routing [12].

Unhappily, coding each of those very complex and sophisticated BCPs has been a highly demanding task, measured on several work-months of a skilled team. The present work presents a VRP framework where state-of-the-art BCP algorithms for many other specific variants can be built much more easily.

It is built on top of a general BCP framework, namely BapCod [19]. This platform takes as an input the set of constraints and variables of the problem in its natural/compact MIP formulation, and the distribution of the constrains between master problem and pricing subproblems. The framework then generates automatically the Danzig-Wolfe reformulation and applies the Branch-and-Price algorithm to solve it using stabilized column generation procedure and branching on original (compact) variables. The user can additionally provide customized routines for solving the pricing problem, for the separation of cutting planes and/or to perform branching.

The additional VRP module provides the most effective cutting and pricing routines used in those recent BCPs for specific VRP variants (they have been re-implemented in a very generic way). To access this module, the user need not provide a detailed formulation of the Resource Constrained Shortest Path pricing subproblems. Instead, the user is invited to specify a high-level description of the underlined graphs and the associated resources.

The overall framework has a “friendly” Julia language interface built on the top of recent modeling language for mathematical optimization called JuMP [5]. In order to derive a BCP for a new VRP variant, some additional coding may be needed. However, it is expected that this should not require much time, allowing the user to focus on testing and tuning of parameters, in order to obtain a good performance on the instances of the targeted variant.

Extensive experiments on the most classical VRP variants show that the framework obtains results that are almost as good as the best found in the recent literature. Moreover, we also give examples of the use of the framework for obtaining effective exact algorithms on a wide range of other relevant variants.

References


The vehicle routing problem with stochastic and correlated travel times

Guy Desaulniers, Polytechnique Montreal and GERAD, Canada
Co-authors: Borzou Rostami, Fausto Errico, Andrea Lodi

In this paper, we consider the capacitated vehicle routing problem (CVRP) with stochastic and correlated travel times where the uncertain travel times are assumed to be correlated and follow a multi-variate distribution whose first and second moments are known. To find an efficient routing solution, we use a mean-variance approach where the routes with high travel time variability are penalized. This leads to a parametric binary quadratic program for which we propose two alternative set partitioning reformulations and show how to exploit certain special structure of the correlation matrix. One of these models retain arc flow variables and a quadratic term in the objective function. The other is an integer linear program where the quadratic costs are included in the cost coefficient of the route variables. For each model, we develop an exact branch-price-and-cut algorithm. In the first, the objective function of the master problem is quadratic or needs to be linearized. In the second, the column generation subproblem involves a quadratic cost function and requires the development of a specialized labeling algorithm. The first algorithm can handle a general case where there can exist a correlation between the travel times of any pair of arcs and a simpler case, called the adjacent case, where such correlation exists only for adjacent arcs. The second algorithm is restricted to the latter case. Our computational results on instances derived from existing instances of the CVRP and of the vehicle routing problem with time windows show the efficiency of our proposed algorithms in finding solutions with different means and variances of the total travel time. Instances with up to 32 customers can be solved by the first algorithm in the general case. In the adjacent case, the second algorithm outperforms the first and can solve instances with up to 75 customers.
1 Introduction

The Vehicle Routing Problem (VRP) is one of the central problems in logistics. Since the late 1950’s, it has been the object of a considerable amount of research efforts. In the VRP, the goal is to find a set of routes serving a given set of customers at minimum cost (often, minimum distance). To each customer is associated a given demand for some homogeneous product. Each route should begin and end at a specified depot and serve a subset of customers whose total demand does not exceed the capacity of the vehicle performing this route. In general, the fleet is assumed to be homogeneous with all vehicles having capacity $Q$.

Most of the research on the VRP has focused on the deterministic version of the problem and on its extensions, in which all problem parameters are known precisely before making the routes. However, in practice, several parameters of the problem can be uncertain. One way of addressing this situation is through the use of probability distributions to describe the uncertain parameters, which thus become random variables. This gives rise to the so-called Stochastic VRPs (SVRPs). While this class of problems has received significantly less attention than deterministic VRPs, there have been nonetheless several efforts to tackle different stochastic variants of the VRP. The most commonly studied involve stochastic demands, stochastic customer presence, or stochastic travel and service times. See [4] for a comprehensive exposition.

In this talk, we focus on the Vehicle Routing Problem with Stochastic Demands (VRPSD), where only customer demands are stochastic and specified through probability distributions; all other problem parameters are assumed to be deterministic and known. As for the actual demand of each customer, it is usually assumed that it is only observed upon arrival at the customer location.

Several modeling paradigms have been proposed to formalize the problem and the way in which it is solved (see [4]). In this work, we adopt the a priori optimization approach, which was extensively...
discussed in [1]. Under this approach, the overall decision-making process is broken into two steps that correspond respectively to the planning of the routes and their execution. While the planning stage is rather obvious, it is important to understand what may happen when one tries to perform (execute) a planned route. We first remark that while one could plan routes that would be feasible for all possible realizations of the stochastic demands, in practice, this would be highly inefficient. This means that it is usual (and, on average, effective) to plan routes that cannot deal with the maximal demand of their customers. However, this also means that for some demand realizations, a route may fail. More precisely, a route failure occurs when the demand of a customer exceeds the residual capacity of the vehicle. Whenever a failure occurs, corrective actions, called recourse actions, must be taken and some associated costs, called recourse costs, need to be incurred. The objective in the VRPSD is to minimize the total expected costs (usually measured in distance), which consist of planned route costs, which are deterministic, and expected recourse costs. It is interesting to note that, under the a priori approach, the VRPSD can be cast into the framework of a two-stage stochastic integer program with recourse (see [2] for a comprehensive coverage of stochastic programming).

Several recourse policies have been proposed. The earliest and most used one, which is called the classical recourse policy, was introduced in [3]. Under this policy, the driver follows the planned route until the vehicle returns to the depot (no-failure case) or its capacity is depleted at some customer location. This depletion may occur in two ways: (i) A failure occurs, and then vehicle must perform a back-forth (BF) trip to the depot to replenish its capacity in order to complete the service; or (ii) The vehicle capacity is exactly depleted by a customer demand, in which case, the vehicle returns to the depot to restock and then continues to the next customer (except if this is the last customer).

An alternative recourse policy is the optimal restocking policy, in which vehicles can also perform preventive return (PR) trips to the depot in anticipation of costly failures. This policy was originally proposed by [9] and made famous by [8]. For a given planned route, the optimal restocking policy can easily be determined by solving a straightforward dynamic programming recursion, which allows to determine for each customer an optimal threshold on the residual capacity after serving this customer that is used to decide whether or not a PR trip should be executed. The objective of the underlying dynamic program is the minimization of the total expected costs of performing the route.

While implementing the optimal restocking policy for a given route is easy, the integration of this policy into an exact solution methodology is extremely challenging. In fact, for several years, there have been no successful attempts in this direction. Very recently, Louveaux and Salazar González [7] have proposed an exact solution approach, based on the Integer L-shaped method, to solve the VRPSD under an optimal restocking recourse policy. It should be noted that, while this paper provides bounding procedures applicable to instances in which customer demand distributions are not identical, much of this work focuses on the case where all customers have identical demand distributions and all the reported computational results cover only this case.

The purpose of this talk is to present an exact algorithm to solve the VRPSD under an optimal restocking recourse policy. An important feature of our approach is that it allows for the consideration of different demand distributions for the customers in a computationally effective way, as long as they are discrete and with finite support.

2 Model

The general formulation of our model for the VRPSD under an optimal restocking policy follows existing two-stage formulations for the VRPSD under the classical recourse policy. Namely, the first stage of the model is almost identical to the well-known 2-index formulation for the undirected Capacitated VRP, with $x_{ij}$ ($i < j$) decision variables, which denote the number of times each edge $(v_i, v_j)$ is traversed in the first stage (see, for instance, [6] for further details). The only important difference to notice is the addition of a term $Q(x)$ to denote the cost of expected recourse actions to
the objective function.

Generally speaking, in VRPSD models, the second-stage model, which corresponds to the application of the selected recourse policy, is not described mathematically, since no optimization takes place at this point: one simply has to derive an appropriate computational procedure to compute the value of the expected recourse costs for any first-stage solution $x$. In our case, we must recall that our recourse policy involves the optimization of the thresholds that govern the decisions to return preventively to the depot. We thus have an explicit second-stage model, which will be presented in detail at the conference.

3 Solution Approach and Computational Results

Our solution approach relies on the general structure of the Integer $L$-shaped algorithm, augmented by various effective bounding procedures that are used to derive lower bounding functional (LBF) cuts at integer and fractional solutions as in [6, 5]. A key element of our algorithm is the fact that we are able to derive efficiently effective bounds for the optimal restocking expected cost. These bounds will be described in detail at the conference.

The method was applied to three sets of instances, including instances with asymmetric distribution for demands. Our experiments showed that we were able to solve optimally several of the instances proposed in [7], as well as instances with different distributions for some customers. The largest instances solved have up to 60 customers and four vehicles.

References


We study a vehicle and crew (re-)scheduling problem for a public transit system which is subject to highly stochastic travel times and disruptions. Our research is motivated by the operations of the tramways system in Hong Kong, which serves hundreds of thousands of passengers per day in a densely populated area and whose operations face severe challenges because it does not run on dedicated tracks but must share the road with vehicular traffic in heavily congested areas.

We investigate how the availability of historical and real-time location and traffic information can be exploited to provide decision support to the controllers. We develop a model for the re-scheduling of vehicles and crew, so as to maximize the route frequencies in order to provide good service to passengers, and minimize the violation of staff regulations (meal-break delays and overtime) taking stochastic time-dependent travel-time uncertainties into account.

In the operations that motivated our research, re-assignment of motormen/trams to different routes is possible only upon arrival at a terminal. Therefore, our decision support system is a “look-ahead” model (solved repeatedly on a rolling horizon basis) to find the best set of re-assignments for all trams and motormen that will be arriving at some terminal within the next time period. For all trams arriving at a terminal within the planning period, we consider all possible subsequent schedules that could be assigned to the tram and evaluate the cost (in terms of overtime, demand coverage, meal-break delays, etc.). Using a matching-based model, re-assignment of routes for all trams arriving at the terminals within the planning period is optimised.

We also explored a variant of the model where we consider demand constraints not only for the current planning period, but also several time periods into the future. Another variant of the model considers crew and tram availability. Yet another variant is the incorporation of planned maintenance of the vehicles into the daily scheduling. We are also interested in exploring the robustness of the model when the frequency of re-optimisation is increased. Future research will investigate how real-time demand information (from multi-media sources) might be available and how the system can operate to be more demand-responsive dynamically.
Anticipating Emission-Sensitive Traffic Management Strategies for Dynamic Delivery Routing

Dirk C. Mattfeld, Felix Köster, Marlin W. Ulmer, and Geir Hasle

1Technische Universität Braunschweig, Germany
2SINTEF ICT, Oslo, Norway

The demand for city transportation, individual and freight, is increasing through continuous growing e-commerce and urbanization. This general increase in transportation has led to substantial challenges for both urban municipalities and logistic service providers. Both parties operate within the urban traffic environment and an information exchange suggests itself. However, a communication has not been established yet. In this research, we analyze how the exchange of information, more specific, the provision of traffic control information from the urban municipalities to the service providers leads to benefits for both parties’ objectives.

Urban municipalities need to enable effective and efficient transportation but also need to provide a livable environment for the citizens without pollutions. To this end, many German cities like Braunschweig or Potsdam install emission-sensitive online traffic management systems (TMS) to dynamically control traffic flows based on current and expected emission levels. The city’s TMS constantly monitors the pollution levels in hot-spot areas, where the emissions tend to be critical. If the emission levels exceed a threshold, the TMS changes the traffic strategy. In essence, a strategy limits access to polluted areas while traffic around and out of the hot-spots areas is accelerated by coordinated traffic light intervals. These traffic strategies are dynamically changed during the day with respect to emission levels. The emissions are subject to stochastic elements like weather conditions and congestion. Reliable predictions of future emission levels are possible only for a limited time horizon.

Major producer of urban emissions is the freight transport sector, especially Courier, Express and Parcel companies (CEPs). Recently, CEPs initiate eco-friendly programs. Nevertheless, in practice, CEPs often ignore emissions in their transport activities due to the high cost-pressure keeping in mind that last-mile delivery is responsible for more than 50% of the overall delivery costs. Hence, CEPs optimize and update their delivery routes with respect to delivery costs (Ehmke et al. 2016). One of the main costs factors are the drivers’ working times. Because TMS decisions impact the travel times within the city, these costs indirectly depend also on emissions. More specific, a traffic strategy may change an individual shortest path and/or the travel time between two customers. Each traffic strategy induces an individual travel time matrix and a change in the traffic strategy may render a current routing plan inefficient. Furthermore, a mere reaction to new information may be insufficient. Anticipation of potential future changes is necessary. There are two measures to avoid inefficient planning by anticipation. The TMS can communicate the next planned decisions to the service provider and the provider can estimate potential future decisions based on current

*Corresponding Author, d.mattfeld@tu-braunschweig.de
information by means of predictive analytics. The derived information needs then to be integrated in the planning algorithm.

In this research, we present dynamic routing policies for CEPs anticipating potential traffic strategy changes. We focus on a CEP-routing problem where the TMS provides information about the current and the near-future traffic strategy. The problem under consideration can be defined as *dynamic vehicle routing problem with stochastic changes of travel time matrices* (DVRPMC). A fleet of vehicles delivers goods to a set of customers. A set of travel time matrices is given, each representing a traffic strategy. Initially, the goods are assigned to the vehicles. While the vehicles are on the road, the traffic strategy and, therefore, the travel time matrix changes based on stochastic emission developments. At any point of time, the dispatcher has access to the current and near-future traffic strategy as well as the current emission levels. Based on this information, the dispatcher can dynamically adapt the planned routes for each vehicle of the fleet. The objective is to minimize the expected travel times for delivery.

In the according Markov decision process model of the DVRPMC, we experience several curses of dimensionality. The number of states is vast because of the exponential increase with respect to the number of customers. The information space models the potential emission changes and is therefore continuous. Finally, the action space is large since it incorporates routing decisions. To account for the large state and information space, in every decision point, we apply a heuristic policy sampling emissions and evaluating current decisions with potential future developments. To account for the large action space, our policy identifies critical areas in the matrix with potentially long travel times. We incorporate these areas in the travel time matrix in such as way that we can solve the resulting model with state of the art routing software for delivery planning. The solution determines the current routing plan and the next customer to visit.

We evaluate our method for a case study for the City of Braunschweig, Germany. Braunschweig represents the layout of a “standard” medium-sized city and is therefore often used as reference city in mobility research. We draw on historical emission observations and test the policy for instance settings varying in the number of vehicles and the TMS’s impact. We compare our policy with static routing and dynamic routing on current information. Our analysis provides two main managerial implications:

1. Our anticipatory dynamic routing method reduces travel times for the CEP on average by 6.8% and up to 16.0%. The reduction is particularly high if the number of deliveries per vehicle is large and the impact of the TMS’s decisions is high.
2. A cooperation between city’s TMS and CEP leads to an average reduction of CEP-traffic in polluted areas by 54.6%. The cooperation is therefore highly beneficial for both parties.

Our contributions are as follows. This research is the first quantifying and analyzing how a cooperation of traffic management and CEP lead to benefits for both. With the DVRPMC, we provide a new and relevant dynamic routing problem reflecting emissions and TMS in the decision making. We further provide a comprehensive Markov decision process model enabling the depiction of stochastic *correlated* travel times, a feature generally neglected in the literature. Our work is similar to the recent suggestion by Gendreau et al. (2016) to draw on a “set of suitable designed scenarios” for different travel time patterns. Our presented solution method is able to incorporate correlation and provides excellent solution quality with respect to the objectives of CEP and TMS.

**References**

The Dynamic Bicycle Rebalancing Problem

Kristine Hagen, Marte Dybendal Gleditsch,
Henrik Andersson, Kjetil Fagerholt
Department of Industrial Economics and Technology Management,
Norwegian University of Science and Technology, Trondheim, Norway
Email: henrik.andersson@ntnu.no

1 Introduction

Due to urbanization, the amount of private motorized traffic increases in cities all over the world. The results are traffic congestion and environmental pollution. In response to the growing concerns, green transportation methods, such as bike sharing, have emerged. Because of the ability to contribute to a healthier lifestyle, and increased mobility, Bike Sharing Systems (BSSs) are widely adopted. Today there are more than 1500 active systems, and another 400 under construction.

The concept of a BSS is simple: a user picks up an available bicycle at a docking station, rides it to their destination, and locks the bicycle at a nearby station with an available lock. Due to technological development, new opportunities to operate a BSS and more clever solutions have evolved. However, imbalanced systems is the major operational issue, as stations are regularly empty or full. As a result, customer demand is often unsatisfied. Hence, it is necessary to rebalance the bicycles. This task is called the bicycle rebalancing problem. The rebalancing problem is either static or dynamic, referring to a system without or with customer interaction, respectively. Vehicles, specially designed to transport bicycles, are often used as a mean to solve this problem. These vehicles are referred to as service vehicles.

The evolution of BSS the last decade has made the bicycle rebalancing problem an emerging topic in Operations Research. The bicycle rebalancing problem consists of two main objectives; finding optimal routes for the service vehicles, and finding optimal loading and unloading quantities for the service vehicles at the stations on this route. These objective form what we call a rebalancing strategy. This optimization problem is a fundamental challenge for all BSSs, as the number of possible solutions is vast and the customer demand is uncertain.
2 Problem Description

The bicycle rebalancing problem is dynamic as information concerning changes in the distribution of bicycles and service vehicles are not known beforehand, but becomes available during operation. Furthermore, the problem is stochastic as new arrivals of customers demanding locks or bicycles are assumed to be random variables with known probability distributions.

As the future distribution of the bicycles are partly dependent on a stochastic demand variable and partly by a controlled rebalancing decision, this problem can be formulated as a Markov Decision Process (MDP). A MDP consists of a set of states $S$, together with a set of decisions $X$. A probability distribution for the outcome of the stochastic variable is also known. This probability distribution may depend on the current state and the decision that is made. This problem is to be solved over a given set of decision points $t$ in the horizon $T$.

In the bike sharing problem, it is natural to have a decision point when a service vehicle arrives at a station, as this is a specific event. The dynamic problem can then be approximated by solving a series of smaller deterministic subproblems over a shorter time horizon $\bar{T}$. We call this subproblem the Deterministic Bicycle Rebalancing Subproblem (DBRS).

The initial state describes the situation when the DBRS is to be solved. This includes the number of bicycles positioned at each station, the position of the service vehicles, and the number of bicycles at each service vehicle. We assume that the stochastic information regarding demand at each station is known and follows a constant rate. This means that for each station the net demand is either positive or negative with a rate that describes how many people arrive or depart every time unit, respectively. The rates of deliveries and arrivals of bicycles depend on the time of day and on the station.

When a customer arrives at an empty station with intentions of getting a bicycle, the station is said to be starving. Likewise, when a customer arrives at a full station with intentions of delivering a bicycle, the station is said to be congested. Both of these situations are defined as a violation.

The driving times between the stations are known input parameters to the subproblem. There is a fixed parking time associated with a station visit. This includes the time it takes to park the service vehicle and get started again. In addition to this, there is a variable handling time. This time is proportional to the number of bicycles that are handled, and is constant regardless of whether bicycles are being loaded to or unloaded from the service vehicle.

The fleet of service vehicles is homogeneous with a given capacity. The stations have a given number of locks that may differ from station to station. This means that one cannot bring more bicycles to a station than there are locks available. Nor can one pick up more bicycles from a station than there are bicycles currently parked there. Each station has a
predefined number of bicycles that is calculated to be the optimal at a given time. This is defined as the optimal state of the station. Hence, the goal is to reach this number within each time horizon. The difference between the station load at the end of the time horizon and its optimal state is defined as deviation.

The objective of the DBRP is to minimize the total violation within the time horizon $T$, at the same time as we want to minimize the total deviation at time $T$. In addition to this, we want to maximize the reward given for initiating service vehicle trips where the arrival times exceed the time horizon.

3 Simulation Framework

To test how the DBRS performs in a dynamic and stochastic environment we have developed a discrete-event simulation framework. Based on probability distributions derived from historical customer demand data, the simulator draws different customer arrivals scenarios. A customer arrival is defined as one customer arriving at a station at a specific time wanting to rent or return a bicycle. A customer arrivals scenario is defined as one possible outcome of customer arrivals within a certain time period. This means that a customer arrivals scenario contains information about all the customer arrivals within a certain time period. The DBRS is solved for a given time horizon and given rates of deliveries and arrivals. After this, the solution is evaluated by checking how many customer arrivals, in a given scenario, that are violated when the service vehicles follow the rebalancing strategy determined by the subproblem. Then, the time rolls until next time a service vehicle arrives at a station and the process is repeated.

Preliminary results show that including the total deviation and giving a reward initiating service vehicle trips where the arrival times exceed the time horizon is important. This means that we improve the quality of the results by incorporating aspects that account for future effects. Compared to a situation where only violations are included, the expected percentage violation was decreased by 3.7%.
The static bike sharing rebalancing problem with forbidden temporary operations

Bruno P. Bruck(1), Fábio Cruz(2), Manuel Iori(3), Anand Subramanian(2)

(1) CIRRELT, Canada,
(2) Centro de Informática, Universidade Federal da Paraíba, Brazil,
(3) DISMI, University of Modena and Reggio Emilia, Italy

Abstract

We study the static bike rebalancing problem with forbidden temporary operations. In this problem, one aims at finding a minimum cost route in which a vehicle performs a series of pickup and delivery operations, while satisfying demand and capacity constraints. In addition, a vehicle can visit stations multiple times but cannot use them to temporary store or provide bikes. Apart from bike rebalancing, the problem also models courier service transportation and repositioning of inventory between retail stores, where temporary operations are frequently disliked because they require additional manual work and service time.

To solve the problem, we propose three exact algorithms based on different mathematical formulations. Furthermore, we also present some theoretical results on problem complexity and worst case analysis, as well as an effective feasibility check procedure.

Our three exact algorithms make all use of a branch-and-cut framework, but with different emphases. The first one is based on the use of an aggregate formulation derived from the work of Chemla et al. [2], where an integer variable expresses the number of times the vehicle passes through an arc. The solution of this formulation may be infeasible because temporary operations might be performed in stations visited twice or more. Thus, our first algorithm iteratively removes infeasible solutions, one at a time, in a branch-and-reject process. Our second algorithm is a direct extension of the one presented in Erdoğan et al. [3] and is built upon an expanded arc structure that enables one to only use binary variables. With this structure, constraints on paths can be used to directly forbid infeasibilities. Our third algorithm removes instead infeasibilities by duplicating vertices associated with stations visited multiple times as done in Salazar-González and Santos-Hernández [4]. However, it attempts duplicating as few vertices as possible, using the iterative approach from Bruck and Iori [1], called minimal extended network.
Extensive computational results on instances involving up to 60 stations show that the exact algorithm based on the minimal extended network produces the best results on average.

**Keywords:** Bike sharing; branch-and-cut; minimal extended network.

**References**


Combining optimization and simulation for designing short sea feeder networks with transshipment at sea

C. A. Benjamin Medbøen  Magnus B. Holm  Kjetil Fagerholt  Peter Schütz
Department of Industrial Economics and Technology Management
Norwegian University of Science and Technology, Trondheim, Norway
Email: peter.schuetz@iot.ntnu.no

In an attempt to improve the competitiveness of short sea shipping (i.e. maritime transportation over shorter distances), a novel maritime transportation system has been proposed: Short Sea Pioneer (SSP). The SSP concept builds upon the conventional idea of having several feeder routes served by smaller daughter ships, which are connected to a main route served by bigger mother ships. The unique and new aspect with the system is that mother and daughter ships can meet at sea to transship their cargo. Thus, expensive port and storage costs can be avoided, but requiring synchronized schedules for mother and daughter vessels. Synchronized schedules can be sensitive to delays in the system and therefore need to be robust with respect to potential sources of delays.

The problem we consider can be formulated as a liner shipping network design problem. The objective is to determine an optimal fleet of mother and daughter ships as well as a set of weekly routes and schedules, such that the sum of the charter costs and operational costs is minimized. We consider the case of a Norwegian container shipping company operating between the European continent and Western Norway, as illustrated in Figure 1. The vessels have to visit each port once a week, where continental main ports in Europe can only be visited by the mother ship, coastal main ports, i.e. larger ports in Norway, can be visited by either mother ship or daughter ship (but not both), and local main ports, i.e. smaller ports, can only be visited by the daughter vessels. Transshipment between mother and daughter ships takes place in ocean hubs, which are suitable candidate locations for transshipment at sea. We distinguish between north-going and south-going ocean hubs, where a north-going ocean hub is visited by a north-going mother ship and a south-going ocean hub is visited by a south-going mother ship.

A route sailed by a daughter ship, i.e. the daughter route, can only be sailed by one daughter vessel and must be completed within one week. The main route connecting the
Norwegian west coast with ports on the European continent must be served with either one or two mother ships. We assume there is a known and constant weekly cargo demand between the continental main port and each of the coastal ports. Shipping of cargoes between coastal ports is not considered since the vast majority of the cargoes transported by the case company are either going to or coming from the continental main port. A transshipment can only occur in an ocean hub between a mother and daughter ship. Each cargo can only be transshipped once which means that the ship receiving the cargo will also deliver it to its final destination. We assume that the mother vessel is always chosen large enough to carry all cargo, while there are capacity restrictions for the daughter vessels.

A daughter ship can meet a mother ship once or twice every week for transshipment in the ocean hubs. If a daughter ship meets a mother ship twice, one of the visits must be in a north-going mother ship and the other visit must be with a south-going mother ship. The meeting location must be in the same ocean hub. Furthermore, a north-going mother ship will only deliver cargoes to a daughter ship, whereas a south-going mother ship will only pick up cargoes from a daughter ship.

When weather uncertainty is introduced, the time usage for a ship during a round trip can deviate from the original plan, e.g. due to reduced speeds. For a synchronized transshipment to take place in an ocean hub, both the mother ship and the daughter ship have to be present at the same time. If one of the vessels is delayed, we observe a synchronization violation and, as a consequence, both ships will be delayed. Thus, a single delay can propagate throughout the system and lead to delays for the other ships. If a ship
is delayed too much, it might be unable to maintain a weekly frequency. The SSP logistics system therefore needs robust schedules, e.g. using buffer time windows to account for potential delays, and/or operational flexibility (the ability to speed up above the design speed of the vessel), to ensure the reliability and regularity of the transportation service.

The problem of finding the optimal set of robust routes for mother and daughter ships under uncertain weather conditions is solved by combining optimization and simulation. This framework is illustrated in Figure 2. The iterative process between the master (optimization) and simulation model is referred to as the solution triggered feedback approach.

Figure 2: The optimization-simulation framework with solution triggered feedback.

Based on the input data, a set of feasible mother and daughter routes is generated using a dynamic label setting algorithm. The optimization model (master) determines the optimal set of routes. The solution proposed by the master problem is then evaluated using the simulation model. Using historic observations of wave heights along the selected routes, the true sailing speed of the vessels is estimated and their arrival times at ports and ocean hubs are calculated. Any delay in the schedule of either mother or daughter ship is logged. If a delay causes the ship to exceed its maximum roundtrip duration, a penalty cost is added to the cost of the route to make the route less attractive for the optimization model. The simulated routes with updated cost information are then fed back to the optimization problem in order to determine a new optimal solution. If the new solution contains routes that have not yet been simulated, a new simulation of the selected routes is triggered. This iterative procedure continues until no new (i.e. non-simulated) routes are selected by the master problem, which guarantees that no better solutions can be found based on the trade-off between operational costs and penalty costs.

We present and discuss results for the Short Sea Pioneer concept based on real world data from the Norwegian west coast. The results show that our solution approach is able to generate more robust schedules for the proposed maritime transportation system.
Flexible and robust supply vessels routing and scheduling

Irina Gribkovskaia, Yauheni Kisialiou
Molde University College – Specialized University in Logistics, Pb. 2110, N-6402 Molde, Norway
Gilbert Laporte
HEC Montréal and CIRRELT, 3000, chemin de la Côte-Sainte-Catherine, Montreal, Canada H3T 2A7

Abstract

We solve the problem of tactical supply vessel planning arising in the upstream offshore petroleum logistics. Supply vessels deliver all the necessary materials and equipment to offshore installations from an onshore supply base. Delivery takes place according to a vessel delivery schedule called the weekly sailing plan. The charter cost of supply vessels is the largest cost contributor in the upstream offshore supply. The planning of supply vessels should therefore be done so that their number is minimized and should at the same time provide reliable flow of supplies from the base. The problem involves the determination of the fleet composition and of the vessel schedules over a given time horizon. We present an extended version of this problem involving flexible departures from the base (multiple voyage departure options every day) and the possibility of coupling vessels by swapping their voyages in the second half of the schedule. We propose an exact set partitioning algorithm applicable to small- and medium-size instances. For the solution of larger instances, we have developed an adaptive large neighborhood search (ALNS) heuristic, which yields optimal or near-optimal solutions on small- and medium-size instances within short computing times. Its performance on larger instances is significantly better than that of alternative algorithms previously developed for the same problem. Results of extensive experiments prove the pay-off of adding more flexibility in departure time options and of allowing vessel coupling. Tests on real instances provided by oil and gas operator Statoil indicate potential substantial monetary savings achievable through fleet size reduction by using flexible departures and coupled vessels, given the high supply vessel charter cost.

The execution of a weekly sailing plan is affected by weather conditions, especially in wintertime. Harsh weather conditions increase the number of vessels required to perform the operations as well as the service times at the installations, and thus disrupt the schedule, leading to additional costs and reduced service level. To cope with the influence of weather conditions, we have imposed requirements on robustness during the construction of vessel schedules by applying our ALNS heuristic, that are incorporated through the introduction of intra- and inter-voyage slacks whose positions and durations are dependent on a predefined robustness parameter and on voyage characteristics. We have developed a binary search-based methodology in which ALNS is used to generate vessel schedules with different levels of robustness. This methodology enabled the construction of vessel schedules with the highest level of robustness for fleet sizes equal to and larger than the minimal fleet size found when constructing with ALNS a cost-optimal schedule without requirements on robustness. To demonstrate the behavior of the robustness search algorithm and to assess the stability of the results we have set up several computational experiments and simulated constructed schedules for a realistic time horizon. The proposed optimization-simulation methodology can be applied in several ways. First, it can be used only to generate a schedule with a maximal robustness level for a certain fleet of vessels, omitting the simulation phase. Such an application may be relevant when the service level of the existing fleet of vessels should be maximized. Another possibility, when the approach is fully applied, is to use the output information of the simulation phase for a trade-off analysis between schedule costs and expected service level. The optimization-simulation methodology that we have proposed enables decision makers to generate supply vessel schedules for large size problems instances with required robustness for expected weather conditions for a given time period.
Global container liner shipping networks are composed of services and each service represents a roundtrip that connects a set of ports following a published schedule. The majority of services is operated at a weekly frequency, and each port on a service is visited at the same time each week. Commonly, all vessels deployed on a service are of the same capacity. Liner services are connected through common port calls that allow liner network operators to move cargo from one service to another. The movement of containers between services is called transshipment and enables large liner shipping companies to transport containers between almost any possible pair of ports around the globe.

Current state-of-the-art models and methods (e.g. Karsten et al., 2017) for liner shipping network design problems only determine the routes and sailing speeds for individual services, but approximate transshipment times by a constant. In practice, the transshipment time between two services depends on how well the schedules of the individual services are synchronized. In the network of the world’s largest overseas cargo carrier, around half of all transported containers are transshipped, and the transshipment times may significantly affect the total transit time of containers between their origin and destination port.

The Integrated Liner Shipping Network Design and Scheduling Problem (LSNDSP) extends the classic liner shipping network design problem by defining schedules for all services, and by considering the interdependency between these. The goal is to construct a network of scheduled services and to determine feasible container routes through the network such that the revenue from transporting cargo minus the cost of operating the services and handling cargo is maximized. A service is defined by the deployed vessel class and its capacity, a cyclic route and a schedule. The schedule implicitly defines the sailing speed on each sailing leg. The length of a service is required to be a multiple of a week and the number of vessels operating the service is equal to its duration to ensure a weekly frequency. A demand represents a weekly quantity of containers for a particular origin-destination pair of ports. Each demand is associated with a unit revenue. Additionally, transit time limits apply, reflecting a demand’s time-sensitivity. Cargo can be transshipped between services, but every transshipment implies additional costs.
for cargo handling at the transshipment port. The transshipment time depends on the schedules of the unloading and the loading service. If a minimum transshipment time is not met, cargo may have to wait one week for the next vessel to arrive. A limit on the number of transshipments may be defined for a demand, reflecting the shippers preference towards direct shipping routes. A demand can, but does not have to be served by the cargo carrier, and it can be fulfilled partially.

We model the problem over a directed time-space graph $G(V, A)$, with vertices $V$ representing a port at a particular time within a week (168h), and arcs $A$ representing sailings ($A^S$) or transhipments ($A^T$). Figure 1 illustrates a solution over a small time-space graph of four ports and a time discretization of 24 hours. The problem formulation over graph $G$ is a variation of service network design problems, which are generalizations of (capacitated) fixed-charge network design problems (Crainic, 2000). The LSNDSP is \( \mathcal{NP} \)-hard in the strong sense.

To solve the problem we propose a column-and-row generation (CRG) matheuristic that combines linear programming techniques with heuristics. The method can be used to construct new liner networks or to extend or improve existing liner networks. In our talk we will discuss some

<table>
<thead>
<tr>
<th>Instance</th>
<th>Graph $G(V, A)$</th>
<th>Model</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$</td>
<td>V</td>
</tr>
<tr>
<td>Baltic</td>
<td>168</td>
<td>7,812</td>
</tr>
<tr>
<td>WAF</td>
<td>280</td>
<td>42,168</td>
</tr>
<tr>
<td>Pacific</td>
<td>630</td>
<td>542,262</td>
</tr>
</tbody>
</table>

Table 1: Graph and model properties for each data instance under a time discretization of 12 hours.
Table 2: Best and average objective function values (cost in USD) per instance, comparing results obtained by the CRG matheuristic under exact and approximated (48h) transshipment times with results from Karsten et al. (2017)

<table>
<thead>
<tr>
<th>Instance</th>
<th>CRG matheuristic</th>
<th>Karsten et al. (2017)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>exact transship. time</td>
<td>48h transship. time</td>
</tr>
<tr>
<td>Baltic</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Best</td>
<td>-2.84·10^5</td>
<td>-2.08·10^5</td>
</tr>
<tr>
<td>Average</td>
<td>-2.24·10^5</td>
<td>-2.08·10^5</td>
</tr>
<tr>
<td>WAF</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Best</td>
<td>-5.92·10^6</td>
<td>-5.77·10^6</td>
</tr>
<tr>
<td>Average</td>
<td>-5.76·10^6</td>
<td>-5.77·10^6</td>
</tr>
<tr>
<td>Mediterranean</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Best</td>
<td>2.54·10^6</td>
<td>2.11·10^6</td>
</tr>
<tr>
<td>Average</td>
<td>2.73·10^6</td>
<td>2.33·10^6</td>
</tr>
<tr>
<td>Pacific</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Best</td>
<td>2.69·10^6</td>
<td>-0.33·10^6</td>
</tr>
<tr>
<td>Average</td>
<td>3.71·10^6</td>
<td>1.06·10^6</td>
</tr>
</tbody>
</table>

Table 2: Best and average objective function values (cost in USD) per instance, comparing results obtained by the CRG matheuristic under exact and approximated (48h) transshipment times with results from Karsten et al. (2017)

of the applied linear and integer programming techniques in detail.

We tested the method on data instances from the publicly available LINER-LIB benchmark suite, which was developed in collaboration with a large liner shipping company (Brouer et al., 2014). Table 1 shows the resulting graph and model size for the four considered instances.

Table 2 displays the objective function values obtained by the CRG matheuristic for the LSNDSP as well as for the LSNDSP under the simplifying assumption of constant transshipment times. The displayed values represent cost minus revenues, thus lower values reflect better solutions. In the third column, the objective function values found by the solution method for the liner shipping network design problem by Karsten et al. (2017) are reported. Under equal assumptions, the proposed CRG matheuristic consistently finds better solutions for all addressed instances.

We further observe that a 48-hours approximation of transshipment times may result in an overestimation of profits.

References


Goods Distribution with Electric Vehicles

By Gilbert Laporte  
Canada Research Chair in Distribution Management  
HEC Montreal

I will survey several aspects related to goods distribution by electric vehicles (EVs). This includes information on the purchase cost of EVs, market shares, adoption by companies, technical considerations regarding batteries, cost competitiveness, and adoption incentives. This will be followed by an overview of current research in the area of transportation science, and by research perspectives at the strategic, tactical and operational levels.
Routing and Detouring Consideration in Location of Refueling Points on a Network

Pitu Mirchandani  
School of Computing, Informatics and Decision Systems Engineering,  
Arizona State University, Tempe, Arizona 85281, United States  
Email: pitu@asu.edu

Yazhu Song  
School of Computing, Informatics and Decision Systems Engineering,  
Arizona State University, Tempe, Arizona 85281, United States  
Email: ysong70@asu.edu

Due to environmental and geopolitical reasons, many countries are embracing electric vehicles (EVs) as an alternative to gasoline powered automobiles. Other alternative-fuel vehicles (AFVs) powered by compressed gas, hydrogen or biodiesel have also been tested for replacing gasoline powered vehicles. However, since the associated refueling infrastructure of AFVs is sparse and is gradually being built, the distance between refueling points (RPs) becomes a crucial attribute for attracting drivers to use such vehicles. Optimally locating RPs will both increase demand and help in developing the refueling infrastructure.

This paper introduces a new set of location problems related to locating RPs on real lines, trees and general networks. First, the problem of feasibility is studied. A set of RPs is said to be feasible if all Origin-Destination (O-D) flows can be served. Given there are feasible locations, the optimal location problem then becomes "where should these RPs to be located such that a given fuel-related objective is minimized", for example, the objective of minimizing the maximum distance between RPs minimizes the range anxiety for drivers. Scenarios include single one-way O-D pair, multiple one-way O-D pairs, round trips, etc.
1. Background

Routing Issues

Taking a trip, especially one through sparsely populated areas, requires the driver to plan when to refuel the vehicle. Given the abundance of filling stations for gasoline powered vehicles, drivers usually consider refueling only when their fuel tanks are low. However, in the case of range-limited vehicles (RLVs), planning when to refuel becomes very important, since at least initially, RPs would be few and far between. Therefore, one needs to develop models which look for the routes that include detouring if necessary. Objectives for such models could be (a) minimizing the total detouring distances, and (b) minimizing the total number of refueling stops. Finding routes in a network considering refueling detours have been studied by, among others, Ichimori (1981), Laporte and Pascoal (2011), Smith et al. (2012), and Adler and Mirchandani (2014).

Location Issues

Suppose we wish to locate \( m \) RPs in a place where there are none currently. The problem of optimally locating such RPs has been investigated by Kuby and his collaborators, e.g., Kuby and Lim (2005), Kuby and Lim (2007), Upchurch et al. (2009), Lim and Kuby (2010), and Capar et al. (2012). Typically, they use modifications of flow “capturing” or flow interception models [Hodgson (1990), Berman et al. (1992), Rebello et al. (1995)], to locate a given number of RPs chosen from a given set of potential sites to cover as many O-D routes as possible.

Cabral et al. (2007) considered a network design problem with relays (NDPR) in the context of telecommunication network design and proposed a column generation scheme and four algorithms. Konak (2012) also studied NDPR and proposed a set covering formulation with a meta-heuristic algorithm. However, these models only choose vertices to locate relays, which normally will not be optimal.

In our research, minimization of the number of RPs used to refuel all O-D flow is considered as the first objective, since building such refueling infrastructure is costly. The objective of the problem is to locate a fixed number of stations (with no given candidate sites) to minimize weighted sum of the distance traveled by the vehicles. We start from the simplest case, locating refueling stations on real lines, then on trees, and finally we will extend to general networks.

![Figure 1 Hierarchy of locating RPs on a network](image-url)
2. Some Location Problems on a Line
A prototypical location problem that considers refueling detours is to locate RPs to minimize the total detouring distance for given discrete O-D demands, one-way and round trips. The following models are trying to locate minimum number of RPs and do not consider any constraints on the number of stops limitation.

We begin by illustrating how to find feasible RPs’ locations along a road with two nodes only. Suppose that an RLV starts the round trip $O \rightarrow D \rightarrow O$ with a fully charged battery, and that two RPs are necessary and sufficient for refueling the trip. Let $r$ denote the maximum distance that our fully charged RLV can travel before refueling.

![Figure 2 Feasible Set for Locating 2 RPs](image)

To refuel the outbound trip $O \rightarrow D$, the first RP should be located within $r$ distance to node $O$ and within $2r$ distance to node $D$, and that the second RP should be located within $2r$ distance to $O$ and within $r$ distance to $D$. Similarly, to refuel the inbound trip $D \rightarrow O$, the second RP should be within $r/2$ distance to $D$ and within $2r$ distance to $O$, and the first RP should be within $3r/2$ distance to $D$ and within $r$ distance to $O$. Thus, the set of potential sites for the first RP includes all points on the line segment from point $d(= d_{OD} - 3r/2)$ to point $a(= r)$, and the set of potential sites for the second RP includes all points on the line segment from point $c(= d_{OD} - r/2)$ to point $b(= 2r)$, as indicated in Figure 2. In addition, the distance between the two RPs should be less than or equal to $r$.

Round-trip Problems

We now consider the case where RLVs go for round-trips between any two nodes along a road. Suppose that our RLV starts its trip with a full battery. Let $r$ denote the maximum distance that our fully charged RLV can travel before refueling, and let $L$ denote the length of the road between two endpoints, where $L > r$. Let triple $(i, j, i)$ represent a round-trip demand, that is, a vehicle starts at node $i$, goes to $j$ and goes back to $i$. Below are two propositions whose proofs will appear in the full paper.

![Figure 3 A Line Network with Discrete Demand Points](image)

**Proposition 1:** If round-trips $(1, n, 1)$ and $(n, 1, n)$ can be refueled, then any round-trip $(i, j, i)$ for $1 \leq i, j \leq n$ can be refueled with possible detours.

**Proposition 2:** The minimum number of RPs that are necessary and sufficient to serve all round-trips is $m = \left\lceil \frac{L}{r} \right\rceil$.

Locating stations to minimize total travel distance can be formulated as a mixed integer quadratic mathematical program, where the location variables are continuous on the line, the assignment of RPs to trips are 0-1 binary variables, and the objective is quadratic because it has terms where continuous variables are multiplied by 0-1 variables. The problem can be solved by MATLAB by using OPTI toolbox.
The existence of finite dominating set can be proved (not shown here), and we then formulate the problem as a shortest path problem on an acyclic network. The network in the formulation has several layers of nodes: It has one layer corresponding to each RP. For example, the layer $k$ has $n(k)$ nodes, $k_1, k_2, \ldots, k^{n(k)}$, where node $k^i$ signifies that the RP $k$ is located at the breakpoint $k^i$ (breakpoints are illustrated in Figure 4). The nodes $k^i$ and $(k + 1)^j$ are connected if the distance in between is less than or equal to $r$. The edge costs are defined as detouring distances. Therefore, the shortest path of the constructed network corresponds to a set of RPs which is feasible and minimizes the weighted sum of the distance traveled by the vehicles.

3. Some Location Problems on a Comb Tree

Consider a prototypical location problem on a small tree (Figure 4). In the illustrative example, by a small tree it is meant that one single RP is sufficient to serve all O-D demands. Consider the tree as shown: there are six ordered O-D pairs in total, and consider one-way problem for these O-D pairs.

Figure 4 A Small Tree with Breakpoints $A^1$, $A^2$, $B^1$, $B^2$, $C^1$ and $C^2$

The breakpoints $A^1$, $A^2$, $B^1$, $B^2$, $C^1$ and $C^2$ are such that the distances $AA^1$, $AA^2$, $BB^1$, $BB^2$, $CC^1$ and $CC^2$ are all equal to $r$. The bold segments, which constitute a subtree, represent the intersection of all paths between these breakpoints. Note that the RP must be located on this subtree, and all O-D demand pairs can be served with possible detouring. Specifically, if our objective is to minimize the total detouring for one-way trips, then the junction $J$ is the only optimal location to achieve zero detouring.

We consider a more general comb graph, where each leaf node serves as both origin and destination. We have a comb graph $G = (V, E)$, where $V$ is the set of all nodes and $E$ is the set of edges. The node set $V$ is further partitioned into two subsets: $N = \{0, 1, \ldots, n\}$ is the set of origin and destination demand nodes and $J = \{n + 1, \ldots, 2n - 1\}$ is the set of junction nodes; and the edge set $E = \{(0, n + 1), (2n - 1, n)\} \cup \{(i, n + i): 1 \leq i \leq n - 1\} \cup \{(n + i, n + i + 1): 1 \leq i \leq n - 2\}$.

Figure 5 A Comb Graph with Discrete Demand Points

To determine the minimum number of RPs needed, we perform a two-step algorithm: (1) Trimming, for each comb tooth $(i, n + i)$ with a length of $r$ or more, starting from node $i$, we place RPs $r$ distance apart until we reach the junction $n + i$. Hence, after trimming, we get a new comb graph, each end node may or may not be equipped with a RP. (2) A greedy method (to be shown in full paper) has been developed, which not only determines the number of RPs needed but also a location interval for each RP.
Similarly, the existence of finite dominating set can be proved, and the problem can be formulated and solved as a shortest path problem on an acyclic network.

4. Conclusions and Future Research

This abstract introduces some new problems of locating RPs on a line and comb-tree networks. In special cases, after locating optimally, no detouring is necessary. However, in general, detouring is necessary when one needs to consider round trips between the O’s and the D’s, and when there is an underlying tree structure. In the full paper to be given at the conference, complexity results for problems on optimally locating on a tree will provided and exact and algorithms for these problems will be evaluated.

References


Routing a Mix of Conventional, Plug-in Hybrid, and Electric Vehicles

Gerhard Hiermann, nuwings OG, Austria
Richard F. Hartl, Department of Business Administration, University of Vienna, Austria
Jakob Puchinger, LGI CentraleSupélec, Université Paris-Saclay and IRT SystemX, Paris-Saclay, France
Thibaut Vidal, PUC-Rio, Pontifical Catholic University of Rio de Janeiro, Brazil

In recent years, the production of battery electric vehicles (BEVs) for large markets has increased, and considerable efforts have been made to reduce vehicle and battery costs. However, the relatively limited capacity of batteries considerably reduces the operational range of BEVs, such that time-consuming visits to recharging stations must be considered in the planning phase.

Hybrid vehicles with features of both BEVs and internal combustion engine vehicles (ICEVs) have been developed to reduce the necessary stops and the infrastructure dependence. The so-called plug-in hybrid electric vehicles (PHEVs) have two engines - an internal combustion engine (ICE) and a pure electric engine - that can be easily switched, permitting the use of electric mode on selected route segments. PHEVs do not have the operational range restriction of BEVs, and they can recharge en route to reduce the use of fossil fuel on other trip segments, which can be beneficial in terms of cost and emissions. However, two engines also mean a heavier base load, which in turn leads to a higher consumption of both electricity and fuel. Each technology has its merits, and with the variety of subtypes available, choosing the best fleet mix for a given transport demand is a very difficult task.

In this talk, we introduce a hybrid heterogeneous electric fleet routing problem with time windows and recharging stations (H²E-FTW). This problem considers three different vehicle classes - ICEV, BEV, and PHEV---as well as multiple vehicle types for each class, differing in capacity, battery size, and electric energy and/or fuel consumption per mile. The batteries of BEVs and PHEVs can be charged at recharging stations. To retain a simple problem formulation and focus on the fleet-composition decisions, we assume that the recharging time is proportional to the amount of energy charged. For PHEVs, the engine type used can be switched at any given time by the driver or an on-board unit. The routing cost consists of the electric energy and the fuel consumption weighted by their respective costs. The number of vehicle classes and types used in the final mix is not limited, but a fixed cost per usage balances the fleet cost with the consumption-based variable cost. By integrating tour planning and vehicle selection within this problem, we aim to achieve better overall plans compared to those obtained by sequential decisions. To this date, integrated approaches have not been often considered in the literature, especially for ICEVs, BEVs, and PHEVs of different types.

The three vehicle classes differ in the need to recharge and to choose the engine. To account for these decisions, we propose a systematic route evaluation approach using decision layers, hence progressing towards a unified view of the problem. These layers separate the vehicle-specific knowledge from the higher-level solution procedure, and this generic design allows us to apply either existing or new metaheuristic approaches and operators. Based on the characteristics of the problem, the vehicle routing problem with time windows (VRPTW) is viewed as the highest unifying layer. In the lower layers of the algorithm, we use labeling algorithms for the placement of recharging stations and greedy policies for the engine use. Finally, we rely on an efficient hybrid genetic algorithm (HGA) with a local search (LS) for solution intensification, along with state-of-the-art population management techniques for diversification. Two additional components are also included: a set partitioning component, which is regularly used to create a new solution based on routes discovered in the past search; and a large neighborhood search (LNS) which acts as a mutation operator.
Through extensive numerical studies, we evaluate the performance of this metaheuristic on a variety of benchmark instances and problem variants. For the special case of the heterogeneous fleet size and mix problem considering only BEVs (E-FSMFTW; Hiermann et al., 2016; Montoya, 2016), our approach finds solutions of better quality than previous algorithms in a similar time. Additional comparative analyses, on the electric VRPTW with recharging stations (E-VRPTW; Schneider et al., 2014) and partial recharging (E-VRPTWPR; Keskin & Catay, 2016) further demonstrate the competitiveness of our approach. Finally, to gain additional managerial insights, we performed sensitivity analysis to evaluate the merits of a mixed vehicle fleet, and measure the impact of fuel and electricity costs on fleet composition.

References


The Most Reliable Flight Itinerary Problem

Jan Ehmke (University of Magdeburg), Michael Redmond, Ann Campbell (The University of Iowa)

Travel itineraries between many origin-destination (OD) pairs can require multiple legs, such as several trains, shared rides or flights, to arrive at the final destination. Travelers expect transparent reliability information to help improve decision-making for intermodal and multi-leg itineraries. We focus on airline travel and making a priori decisions about flight itineraries based on the reliability of arriving at the destination within the travel time budget. We model the reliability of intermodal multi-leg itineraries and, given publicly available data, create probability distributions of flight arrival and departure times. We use these values in our reliability calculations for the entire itinerary. We implement a stochastic network search algorithm that finds the most reliable flight itinerary (MRFI) considering intermodal transportation services on the first mile from the origin to the airport and the last mile from the airport to the destination. We also implement several ideas to improve the efficiency of this network search. Computational experiments help identify characteristics of the MRFI for a diverse set of OD pairs.
The value of integration in supply chain planning

Martin P. Kidd\textsuperscript{a}, Maryam Darvish\textsuperscript{b}, Leandro C. Coelho\textsuperscript{b}

\textsuperscript{a}Division of Management Science, Technical University of Denmark
\textsuperscript{b}Canada Research Chair in Integrated Logistics, CIRRELT and Faculté des sciences de l’administration, Université Laval, Canada

As supply chains become more complex and globalized, logistic efficiency is essential for a firm to maintain its competitive strategy. It is well known that a lack of coordination and information sharing among facilities with conflicting objectives is a major cause of large logistic inefficiencies in a supply chain. Therefore, the literature emphasizes that a shift from the traditional approach of localized planning to a global, integrated solution is one way in which systemwide costs can be greatly reduced (Darvish et al., 2016; Guastaroba et al., 2017). Subsequently, a large body of research exists on integrated optimization problems in logistics and supply chain management (e.g. Coelho et al. (2014); Schmid et al. (2013)), and Table 1 summarizes some of the well-known problems that integrate up to three decisions of production, inventory, distribution and facility location.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Production</th>
<th>Inventory</th>
<th>Distribution</th>
<th>Location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lot Sizing Problem (LSP)</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inventory-Routing Problem (IRP)</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Production-Routing Problem (PRP)</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Location-Routing Problem (LRP)</td>
<td>✓</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Location-Inventory Problem (LIP)</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Overview of some of the well-known integrated supply chain planning problems (adapted from Adulyasak et al. (2015)).

Theoretically speaking, the feasible region of an integrated approach contains the feasible regions of the localized problems, and therefore always yields a superior solution. In reality, however, integration significantly increases the complexity of the entire decision making process as more coordination and information sharing is needed. In addition, moving from a localized planning approach to an integrated one might require an upgrade in organizational and information infrastructure that impose additional costs. The question naturally arises: how can we quantify the value of integration in order to support the strategic decision to shift from localized planning to integrated planning? The concept of the value of integration has been considered before, for example by Darvish and Coelho (2017), who highlight the benefits of integration over sequential/hierarchal approaches to supply chain planning. In this work we continue along this line of research.

Crucial to our research question is that it is not only a matter of whether or not to integrate, but also what to integrate. Figure 1 shows two approaches to solving the planning problem from Table 1, where the subproblems of location, production, inventory and distribution have to be solved. The model on the left first solves the location problem and then fixes its solution. It then moves on to making and fixing the production decisions and then solving the IRP. The model on the right, however, integrates all decisions in a single optimization problem. The model on the right is “more integrated” than the one on the left, and so the potential for improved performance is higher. Furthermore, from a complexity point of view, the left one might be
preferable, depending on the case. Here is where the value of integration will play an important role.

![Diagram showing integration in supply chain planning]

Figure 1: Examples of two different levels of integrated supply chain planning.

In this talk we will present some initial results on data from different industries. In order to compare several possible levels of integration (such as the ones shown in Figure 1), computational tests will be conducted by using mixed integer programming (MIP) formulations of the subproblems, depending on how the integration is done. We will address the question of whether or not there is a relationship between a good way of integrating decisions and certain features of the supply chain.

References


The last mile delivery problem

Yuan Yuan, Diego Cattaruzza, Maxime Ogier, Frédéric Semet
Univ. Lille, CNRS, Centrale Lille, Inria,
UMR 9189 - CRIStAL Lille, France
Email: frederic.semet@centralelille.fr

1 Introduction and problem definition

E-commerce is a thriving market around the world and suits very well the busy lifestyle of today’s customers. An annual survey by analytics firm comScore and UPS revealed that consumers in US were purchasing more things online than in stores in 2016\(^1\). According to the Ecommerce foundation, 1.4 billion people purchased goods and/or services online at least once in 2015. They spent $2,272.7 billion online, which results in an average spending per e-shopper of $1,582. It is obvious that the e-commerce growth poses a huge challenge for transportation companies, especially in the last mile delivery. According to [1], the last mile parcel delivery cost often reaches or even exceeds 50% of the total transportation cost, making it a top concern for many companies.

Nowadays, the most common last mile delivery service is home delivery. Customers wait at home to get their orders. Besides home delivery, companies like Amazon and FedEx, develop locker and pick-up&go delivery services. When customers shop online, they can choose a nearby locker or a store offering a pick-up&go counter. In the past two years, a new concept called trunk delivery, has been proposed. Here, customers’ orders can be delivered to the trunk of their cars. Volvo launched its in-car delivery service in Sweden in 2016. The courier has a one-time digital code to get access to the car.

Trunk delivery is different from home delivery and locker delivery since the car moves during the day and can be in different locations during the planning horizon. We study an efficient last mile delivery system that combines all these delivery services: home, locker, pick-up&go location and car trunk.

In this presentation, we address the routing problem for one vehicle. The problem is modeled on a directed graph \(G = (V, A)\). The set of vertices \(V = \{0, 1, ..., N\}\) is partitioned into \(C_0 = \{0\}, C_1, ..., C_K\) clusters. Cluster \(C_0\) contains only the depot. Each other cluster \(C_k, k > 0\) represents the set of alternative locations on which client \(k\) can

\(^1\)http://fortune.com/2016/06/08/online-shopping-increases/
be delivered. Each vertex is associated with a time-window $[E_i, L_i], i \in \{0, 1, ..., N\}$ with $[E_0, L_0] = [0, T]$. A visit can only be made to a vertex during its time-window and an early arrival leads to waiting time while a late arrival causes infeasibility. In a cluster, the time-windows associated with the vertices may overlap. Arcs are only defined between vertices belonging to different clusters, that is, $\mathcal{A} = \{(i, j) : i \in \mathcal{C}_k, j \in \mathcal{C}_l, k \neq l\}$. Each arc $(i, j) \in \mathcal{A}$ is associated with a traveling cost $C_{ij}$ and time $T_{ij}$. A tour starts from and ends at the depot.

The objective is to find a minimum cost tour visiting each customer at one location within the associated time-window. The problem that arises is called the Generalized Traveling Salesman Problem with Time Windows (GTSPTW). We assume that this problem is static and deterministic, namely all customer locations and the associated time-windows are known with certainty in advance. The GTSPTW can be modeled as follows.

$$\min \sum_{(i,j) \in \mathcal{A}} C_{ij} x_{ij}$$

s.t. $\sum_{i \in \mathcal{C}_k} y_i = 1$ $k \in \{0, 1, ..., K\}$

$$\sum_{(i,j) \in \delta^+(i)} x_{ij} = \sum_{(j,i) \in \delta^-(i)} x_{ji} = y_i \forall i \in \mathcal{V},$$

$$\sum_{i \in \mathcal{C}_k} E_i y_i \leq t_k \leq \sum_{i \in \mathcal{C}_k} L_i y_i$$

$$t_k - t_{\ell} + T_{ij} x_{ij} \leq \sum_{u \in \mathcal{C}_k} L_u y_u - \sum_{v \in \mathcal{C}_\ell} E_v y_v - (L_i - E_j) x_{ij} \forall (i, j) \in \mathcal{A}, i \in \mathcal{C}_k, j \in \mathcal{C}_\ell,$$

$$y_i \in \{0, 1\} \forall i \in \mathcal{V},$$

$$x_{ij} \in \{0, 1\} \forall (i, j) \in \mathcal{A}.$$
Generalized Travelling Salesman Problem (Fischetti et al. [2]). For the multi-vehicle case, the problem is named Generalized Vehicle Routing Problem with TW (Moccia et al. [4]). The special case where time-windows on clients do not overlap has been recently considered by Reyes et al. [3]. The problem is called the Vehicle Routing Problem (VRP) with Roaming Deliveries and models the case when only trunk deliveries are considered.

2 A branch-and-cut scheme

We propose several valid inequalities for GTSPTW. Some of them are derived from valid inequalities for the asymmetric traveling salesman problem with time-windows, others are specific. The main constraints are the followings:

- Feasible path inequality. 
  \[ x_{ij} + \sum_{h \in S_{ij}} y_h \geq y_i + y_j - 1 \quad i \in C_k, j \in C_\ell, \]
  where 
  \[ S_{ij} = \{ h \in V \setminus (C_k \cup C_\ell) | E_i + T_{ih} \leq L_h, E_h + T_{hj} \leq L_j, E_i + T_{ih} + T_{hj} \leq L_j \} \]
  for \( i \in C_k \) and \( j \in C_\ell \).

Two vertices are visited either directly or there exists a connection between them.

- Infeasible path elimination inequality. Several infeasible path elimination constraints can be defined. The simplest ones are the followings:
  \[ x_{ij} + \sum_{h \in C_{ij}^k} y_h \leq 1, \quad k \in \{1, \ldots, K\}, i, j \in V \setminus C_k, i \neq j \]
  where 
  \[ C_{ij}^k = \{ h \in C_k | E_i + SP_{hi} + T_{ij} > L_j \text{ or } E_h + SP_{hi} > L_i \text{ or } E_i + T_{ij} + SP_{jh} > L_h \text{ or } E_j + SP_{jh} > L_h, i, j \in V \setminus C_k \}. \]

\( SP_{ij} \) represents the shortest traveling time from vertex \( i \) to vertex \( j \). When the triangle inequality is not satisfied, the shortest path from \( i \) to \( j \) can include the visit of other vertices. These constraints detect if a vertex and an arc cannot be simultaneously selected in a feasible solution due to time-windows.

- Generalized subtour elimination inequalities (GSECs).

- Clique inequality. 
  \[ \sum_{i \in S} y_i \leq |S| - 1. \]

If no feasible path passing through all the vertices of a set \( S \subset V \) exists (due to time-windows), then the number of vertices of \( S \) that can be visited in all the feasible solutions are less than the size of \( S \).

We develop a branch-and-cut algorithm for the GTSPTW. We include at the root node of the branch-and-bound tree all polynomial sets of inequalities while GSEC inequalities and clique inequalities are separated in the course of the algorithm. An initial solution is provided to the algorithm thanks to a heuristic which combines the generation of sequences of clusters and the solution of shortest path problems with resources constraints.

The algorithm is implemented in C++ using Cplex 12.6 and the Concert technology. Preliminary results are obtained on instances generated from GTSP instances. When
creating an instance, we guarantee that a feasible solution exists. We set the CPU time limit to 1 hour. Instances with up to 22 clusters and 107 vertices are solved to optimality. The average solution time for instances with less than 20 clusters is 19.4 seconds while it is equal to 441.6 seconds for instances with 20 and 22 clusters.

References


1 Introduction

We study a hub location problem that involves two levels of decision makers acting non-cooperatively. The upper level decision maker (leader) locates \(q\) hubs and provides transportation service between the hubs at a price to be determined. Each of the lower level decision makers (followers) aims to send its commodity from a source to a destination with minimum cost, either routing their commodities using the existing infrastructure or using the arc hubs. More precisely, the arc hubs are used only if the transportation cost is cheaper by a factor \(\gamma\) than the cost of using the original infrastructure. The leader aims to maximise its revenue, finding the optimal location of the hubs and determining prices for all arcs connecting hubs.

Hub location problems have been studied extensively, with many variants based on the number of hubs to transport commodities, hub capacities, and objective function type (minisum or minimax). We refer the interested reader to the book of Contreras [1] for a comprehensive review. Joint hub location and pricing models, on the other hand, received less attention. Lüer-Villagra and Marianov [2] have studied a competitive hub location and pricing problem, where two hub networks compete to maximize their profits, with the existing hub network using mill pricing and the entrant using arc-based pricing. The most relevant study is [3], where the authors focus on the problem of deciding which arcs of a given network to invest on and subsequently apply pricing to.

2 Formulation

Consider a complete directed network \(G = (V,A)\) with vertex set \(V\) and arc set \(A\), and a set of commodities \(K\) to be transported. For each \(i \in V\), we define \(\delta_i^+\) as the set of outgoing arcs from node \(i\) and \(\delta_i^-\) as the set of incoming arcs. Furthermore, we write \(t_k\) to denote the total amount of
commodity $k \in K$ to be routed from the origin $o(k) \in V$ to the destination $d(k) \in V$. We define $c_{ij}$ as the cost of using the arc $(i, j) \in A$ for any follower, and assume that these costs satisfy the triangle inequality. For the sake of brevity, we write $d_{ij}^k$ to denote the cost of using arc $(i, j) \in A$ to transport commodity $k \in K$ without using the hub network, where $d_{ij}^k = \delta \times c_{ij}$ if $i = o(k)$ and $j = d(k)$, and $d_{ij}^k = c_{ij}$ otherwise. In addition, we define $b_k^i = 1$ if $i = o(k)$, $b_k^i = -1$ if $i = d(k)$ and $b_k^i = 0$ otherwise.

For any $i \in V$, let $x_i$ be equal to 1 if a hub is located at node $i$, and 0 otherwise. In addition, for any $(i, j) \in A$ and $k \in K$, let $y_{ij}^k$ be equal to 1 if commodity $k$ is transported on arc $(i, j)$ without using the hub network, and 0 otherwise. Similarly, for any $(i, j) \in A$ and $k \in K$, let $w_{ij}^k$ be equal to 1 if commodity $k$ is transported on arc $(i, j)$ using the hub network, and 0 otherwise. The transportation variables are stated as $y_{ij}^k$, $w_{ij}^k$ for the followers. Finally, let $p_{ij}$ be the price of using the hub network for crossing the arc $(i, j) \in A$. A formulation of our problem can then be stated as:

$$\text{maximize} \quad \sum_{k \in K} \sum_{(i, j) \in A} t_k p_{ij} w_{ij}^k$$

subject to

$$\sum_{j \in V} x_i \leq q$$

$$p_{ij} \geq c_{ij}(1 - x_i) \quad \forall (i, j) \in A,$$  

$$p_{ij} \geq c_{ij}(1 - x_j) \quad \forall (i, j) \in A,$$  

$$\sum_{j \in \delta_i^+} y_{ij}^k \leq x_i \quad \forall i \in V, k \in K,$$  

$$\sum_{j \in \delta_i^-} w_{ij}^k \leq x_j \quad \forall i \in V, k \in K,$$  

$$\sum_{j \in \delta_i^+} (y_{ij}^k + w_{ij}^k) - \sum_{j \in \delta_i^-} (y_{ji}^k + w_{ji}^k) = b_k^i \quad \forall i \in V, k \in K,$$  

$$x_i \in \{0, 1\} \quad \forall i \in V,$$  

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

$$\sum_{(i, j) \in A} (d_{ij}^k y_{ij}^k + p_{ij} w_{ij}^k) =$$

$$\text{minimize} \quad \sum_{(i, j) \in A} (d_{ij}^k y_{ij}^k + p_{ij} w_{ij}^k)$$

subject to

$$\sum_{j \in \delta_i^+} (y_{ij}^k + w_{ij}^k) - \sum_{j \in \delta_i^-} (y_{ji}^k + w_{ji}^k) = b_k^i \quad \forall i \in V,$$  

$$\forall (i, j) \in A.$$  

The objective function (1) maximizes the profit of the leader. Constraint (2) sets the maximum number of hubs to be located. Constraint sets (3) and (4) state that the price of an arc cannot be less than the cost of direct transportation unless both ends of the arc are hubs. Constraint sets (5) and (6) enforce both vertices at the ends of an arc to be hubs if a commodity is transported on the arc using the hub network. Constraint set (7) states the flow conservation for the commodities. Constraint sets (8) and (9) require the hub and transportation decisions to be binary. Finally, constraint set (10) ensures that each commodity is transported with minimum cost.
The formulation presented above is nonlinear due to the objective function (1) as well as the bilevel programming constraints (10). To linearize this formulation, we define \( z_{kij}^k = p_{ij}w_{ij}^k, \forall(i, j) \in A, k \in K \). We also define the dual variables \( u^k_i \), associated with constraint \( i \in V \) for commodity \( k \in K \) within (10). The resulting linearized formulation is then:

\[
\begin{align*}
\text{maximize} & \quad \sum_{k \in K} \sum_{(i,j) \in A} t_k z_{ij}^k \\
\text{subject to} & \quad (2)–(9), \\
& \quad z_{ij}^k \leq p_{ij} \quad \forall(i, j) \in A, k \in K, \\
& \quad z_{ij}^k \leq d_{ij} w_{ij}^k \quad \forall(i, j) \in A, k \in K, \\
& \quad z_{ij}^k \geq p_{ij} - (1 - w_{ij}^k) c_{ij} \quad \forall(i, j) \in A, k \in K, \\
& \quad \sum_{(i,j) \in A} (d_{ij}^k y_{ij}^k + z_{ij}^k) = \sum_{i \in V} b_i^k u^k_i \quad \forall k \in K, \\
& \quad u^k_i - u^k_j \leq d_{ij}^k \quad \forall(i, j) \in A, k \in K, \\
& \quad u^k_i - u^k_j \leq p_{ij} \quad \forall(i, j) \in A, k \in K, \\
& \quad u^k_i \text{ unrestricted} \quad \forall i \in V, k \in K.
\end{align*}
\]

Constraint sets (12), (13), and (14) linearize the profit made by the leader. Constraint set (15) replaces (10) to ensure that each commodity is transported with minimum cost. Constraint sets (16), (17), and (18) stem from the dual of the followers’ problems.

We strengthen the formulation by means of valid inequalities and variable reductions. The strengthened formulation is capable of solving randomly generated instances of nontrivial size.

The solutions obtained are compared with the corresponding mill pricing solutions. In mill pricing, all hub arcs prices are set by discounting the cost of the corresponding infrastructure arc by a factor \( \alpha \). A formulation and a heuristic algorithm are proposed to solve the mill pricing version of our problem, in which the optimal location of the hubs and the optimal value of \( \alpha \) have to be determined. Arc pricing solutions are observed to be considerably more profitable than mill pricing solutions.

References


Heuristics for the traveling salesman problem with release dates and completion time minimization

Claudia Archetti∗, Dominique Feillet†, Andrea Mor‡, M. Grazia Speranza§

1 Introduction

A common trait of the classical Vehicle Routing Problems is the assumption that the goods to be distributed are available at the depot when the distribution starts. To consider a more dynamic organization of the distribution, [2] introduces the arrival time of the goods at the depot as part of the optimization problem. This poses the additional problem of whether it is better to wait for additional goods to arrive and have a better loaded vehicle, or to start a route of the vehicle with the currently available goods. The arrival time at the depot of the goods to be delivered to a customer is called its release date. The study of this problem is motivated by the growing interest in topics such as city logistics, where goods are consolidated at a distribution center and distributed to customers typically with smaller and lighter vehicles. In this case, goods are delivered to the distribution center throughout the day, when the final distribution to customers is already taking place. Another application arises in same day delivery problems related to e-commerce logistics.

In this work we consider the TSP-rd(time) introduced in [1] defined on a general graph. A single uncapacitated vehicle is allowed to perform several trips during the time horizon (say, the day), one after the other. We seek the minimization of the service ending time, that is the sum of the traveling time and waiting time at the depot. We present a heuristic approach based on a large neighborhood search and a local search. Two alternative insertion operators, one simple and fast and the other based on a mathematical programming model, are proposed for the large neighborhood search, which give rise to two heuristics. The proposed mathematical programming formulation is used to find the optimal solution on instances with up to 20 customers built from benchmark instances for the classical TSP. On all but one instances the heuristic with the simple insertion operator finds the optimal solution.

2 Definition and notation

The TSP-rd(time) is defined as follows. Let $G = (V, A)$ be a complete graph. A traveling time and a traveling distance are associated with each arc $(i, j) \in A$. These two values are assumed identical and are denoted by $t_{ij}$. It is also assumed that the triangle inequality is satisfied. The set of vertices $V$ is composed by vertex 0 which identifies the depot and the set $N$ of customers, with $|N| = n$. The release date for customer $i \in N$ is denoted by $r_i$, $r_i \geq 0$. This means that the goods for customer $i$ can either arrive at time $r_i$, then $r_i > 0$, or be at the depot at the beginning of the distribution, e.g., because they arrived overnight, then $r_i = 0$. A single vehicle is allowed to perform a sequence of trips. Capacity constraints are not considered. The objective is to minimize the completion time, that is, the total traveling

∗University of Brescia, Department of Economics and Management, Brescia, Italy, claudia.archetti@unibs.it
†Ecole des Mines de Saint-Étienne and LIMOS UMR CNRS 6158, CMP Georges Charpak, Gardanne, F-13541 France, feillet@emse.fr
‡Corresponding author, University of Brescia, Department of Economics and Management, Brescia, Italy, andrea.mor@unibs.it
§University of Brescia, Department of Economics and Management, Brescia, Italy, grazia.speranza@unibs.it
time plus the waiting time at the depot, and serve all customers. Given a solution to the TSP-rd(time), we call route a trip starting and ending at the depot and not visiting the depot in between.

3 Heuristics for the TSP-rd(time)

The idea of the heuristic approach is to iteratively find new solutions through a large neighborhood search (LNS), and to apply a local search (LS) on the solution generated by the LNS. The initial solution is built by solving the TSP on the complete graph $G$ with the Lin-Kernighan (LK) heuristic. The starting time of this unique route is then set to $r_n$. This solution is feasible and has a completion time greater than or equal to $r_n + d_{TSP}$.

3.1 LNS

A new solution is found with a large neighborhood search. Given a solution $S$ and a parameter $\alpha$, the LNS chooses a set $\tilde{N}$ of $\alpha$ customers that are removed from $S$. An insertion operator is used to reassign customers from $\tilde{N}$ to one of the routes in $S$. A final step optimizes each route in $S$.

In order to choose the customers that are removed, we first rank all customers on the basis of one of the following criteria: the starting time saving obtained by removing a customer from the route and the classical detour saving, without accounting for changes in the starting time of the route. At each iteration of the LNS, one of these two criteria is chosen at random. Then, $\alpha$ customers are removed as follows: customers are considered sequentially in the order of the ranking and each customer is removed with a probability 50% until $\alpha$ customers are removed. The assignment of the nodes from $\tilde{N}$ to the routes of $S$ is accomplished by an insertion operator. Two different operators are used which give rise to two heuristics: the first one is based on a mathematical formulation, called MIP hereafter, while the second one is a random insertion approach.

**MIP insertion**: given $\tilde{N}$, a mixed integer linear program is solved where non removed nodes are forced to be visited by the same route as in $S$ and the integrality constraints on the arc variables are relaxed. A maximum time limit is given to the solution of the modified MIP. The solution of the modified MIP provides a clustering for the customers. Routes are computed by applying the LK heuristic on each cluster. A solution is then obtained by computing the earliest possible starting time for each route and by scheduling the routes in the increasing order of these values.

**Random insertion**: this insertion operator randomly assigns each removed node to a cluster. A solution is then built by applying the LK heuristic on each cluster, sorting the obtained routes by the largest release date and computing the earliest possible starting time for each route. The aim of the random insertion is to quickly find a feasible solution.

3.2 Local Search

After each successful LNS move, a local search is applied. The proposed local search is defined by seeking the first improvement in a neighborhood of four possible moves: the relocation of a node to a different route, the merging of two consecutive routes, the split of a route in two separate ones, and the anticipation or postponement of a depot visit. The nodes are randomly sorted and considered sequentially: if the node is not the depot the node relocation
move is considered, otherwise the remaining three moves are evaluated. The first improving move is implemented and the process is repeated until no improving move is found. We call MathTSPrd the algorithm originating when the MIP insertion is used, HeuTSPrd the one where the heuristic insertion operator is used.

4 Computational experiments

The performance of the heuristics has been compared to the optimal solution on instances with up to 20 customers, as those are the largest for which a solution is found by CPLEX within one hour. The HeuTSPrd shows a slight advantage, finding the optimal solution in all but one of the 24 instances of size \( n = 15 \) and \( n = 20 \). The MathTSPrd finds the optimal solution in 21 instances of size \( n = 15 \) and in 22 instances of size \( n = 20 \). Overall, no percentage gap above 1% has been found. A comparison of the two heuristics has also been carried out on larger instances. In the instances of size \( n = 50 \) the average percentage gap of the MathTSPrd from the HeuTSPrd is 0.94%. The best solutions of the MathTSPrd are found in 258 seconds on average, while the HeuTSPrd takes an average time of 14 seconds to find a solution at least as good. For the instances of size \( n = 100 \) the average percentage gap of the MathTSPrd from the HeuTSPrd is 3.4%. The MathTSPrd finds the best solution in 265 seconds on average, while the HeuTSPrd takes an average time of 57 seconds to match these results.

The results obtained indicate that the time required by the operator adopted in the MathTSPrd limits the number of iterations performed in the given computational time compared to those performed when the random insertion is adopted in the HeuTSPrd. This implies that exploring a broader portion of the solution space is more effective than spending more time to find better solutions to feed the LS.

5 Conclusions

The computational results show that the HeuTSPrd outperforms the MathTSPrd. Heuristics that make use of MILP models, the so called matheuristics, have been shown to be effective in several cases when compared to other heuristics. However, in our study, using MILP models as insertion operators in an iterative framework has not turned out to be beneficial for the solution of the TSP-rd(time).

Several research directions remain open. An interesting direction is related to investigating the situations where the use of a MILP model is effective in a heuristic framework. It would also be interesting to study the extension of the TSP-rd(time) problem to the case where a fleet of vehicles is available and a dynamic version of the problem.

References


Heuristics for vehicle routing problems: Sequence or set optimization?

Túlio A. M. Toffolo¹,², Thibaut Vidal³, Tony Wauters¹

¹ KU Leuven, Department of Computer Science, CODeS & imec-ITEC - Belgium
² Federal University of Ouro Preto, Department of Computing - Brazil
³ Pontifícia Universidade Católica do Rio de Janeiro, Computer Science Department - Brazil

1 Introduction

The Capacitated Vehicle Routing Problem (CVRP) is classically described as a combination of a Traveling Salesman Problem (TSP) with an additional capacity constraint which lends a Bin Packing (BP) substructure to the problem (Toth and Vigo 2014). It can also be viewed as a Set Packing (SP) problem in which the cost of each set corresponds to the distance of the associated optimal TSP tour. These problem representations emphasize the two decision sets at play: customer-to-vehicle ASSIGNMENTS, and SEQUENCING choices for each route (Vidal et al. 2013), a duality which has left long-standing impressions in the literature, from the early developments of route-first cluster-second (Bodin and Berman 1979, Beasley 1983) and cluster-first route-second constructive methods (Fisher 1981), all the way to the set-covering-based exact methods and matheuristics which are currently gaining popularity.

Examining the recent progress on metaheuristics for the CVRP, little has changed in recent years concerning intra-route neighborhood search: Relocate, Swap and 2-opt neighborhoods and their immediate generalizations are employed, and these neighborhoods alone are sufficient to guarantee that most solutions resulting from a local search contain TSP-optimal routes. This is generally because classical CVRP instances involve short routes with up to 15 or 20 visits. For such small problems, even simple neighborhood search methods for the TSP tend to produce optimal tours.

Based on this observation, a larger effort dedicated to TSP tour optimization, as a stand-alone neighborhood, is unlikely to result in further improvements. For this reason, it is very uncommon to observe the use of larger intra-route neighborhoods (e.g., 3-Opt or beyond) in recent state-of-the-art metaheuristics. Nevertheless, does this mean that SEQUENCING optimization should be abandoned in favor of more extensive search concerning ASSIGNMENT choices? Certainly not. Indeed, even if local minima exhibit optimal TSP tours, inter-route moves frequently lead to TSP-suboptimal tours which are rejected due to their higher cost, but would be accepted otherwise if the tours were optimized. Such solution improvements would then not arise from separate ASSIGNMENT or SEQUENCING optimizations, but from a careful combination of both.

Figure 1 schematically represents the solution set of the CVRP, whose decision variables are split into SEQUENCING decisions (x-axis) and ASSIGNMENT decisions (y-axis). The y-axis also
represents the solutions in terms of their Assignment decisions solely, ignoring Sequencing choices. These partial solutions can be viewed as a projection (Geoffrion 1970) of the original solutions \( S \) on the space \( S^A \) defined by a single decision subset (Assignment). Moreover, from a solution represented in terms of Assignment decisions, it is possible to find the best associated complete solution by solving each TSP associated with the routes.

![Diagram of two alternative search spaces for the CVRP](image)

Figure 1: Two alternative search spaces for the CVRP

With this picture in mind, it is tempting to conduct a search in the space \( S^A \) rather than \( S \). After all, the size of \( S^A \) is exponentially smaller than that of \( S \), the solutions of \( S^A \) are in average of better quality, and the average size of a path in \( S^A \) is smaller, such that fewer LS iterations are expected for convergence. However, the obvious drawback is that each move evaluation in \( S^A \) requires solving one or several small TSPs to optimality, leading to a significant computational effort. Still, note that considerable progress has been made in the past 30 years with regard to the efficient resolution of TSPs, and small problems with approximately 20 customers are solvable in a few milliseconds. Based on these observations, this work takes a fresh look at heuristic searches for the CVRP to answer two essential questions about the search space \( S^A \):

1. Is it practical and worthwhile to search in the space \( S^A \) rather than \( S \)?
2. If searching in \( S^A \) requires an excessive effort, can we define a search space which maintains most of the key properties of \( S^A \) but can be more efficiently explored?

As will be demonstrated in this presentation, our experiments led us to answer the first question negatively: even with non-trivial memory and speedup techniques (hashtables and move filters) the computational overhead related to the exact resolution of TSPs during each move evaluation, for a complete search in \( S^A \), does not appear worth the gain in terms of solution quality.

By contrast, our answer to the second question is positive. Rather than requiring a complete exact resolution of each TSP, the dynamic programming approach of Balas and Simonetti (2001), hereafter referred to as B&S, can be employed to perform a systematic route optimization during move evaluations. Such a technique has been previously proposed by Balas and Simonetti (2001) and Irnich (2013), but never explored thoroughly. Given a range parameter \( k \) and an initial tour, the B&S algorithm finds, in \( O(k^22^k-2n) \) operations, the vertex sequence with minimum cost
such that no vertex is displaced by more than $k$ positions. This allows us to define a search space $S^B_k$ such that $S^B_0 = S$ and $\lim_{k \to \infty} S^B_k = S^A$. Moreover, even for a fixed $k$, we propose tunneling techniques that exploit the memory of past solutions to dynamically reshape the search space, in such a way that $S^B_k$ converges towards $S^A$ as the search progresses.

To evaluate experimentally the potential of the new search spaces, we conduct experiments with a simple multi-start local search (MS-LS), and with the unified hybrid genetic search (UHGS) of Vidal et al. (2012, 2014). The use of $S^B_k$ for $k \in \{1, \ldots, 3\}$ appears to lead to solutions of higher quality on the new instances from Uchoa et al. (2017). New best solutions were also found for surprisingly small instances with as few as 242 or 256 customers. These solutions had not been attained up to now with classic neighborhoods. Overall, this research allows to better evaluate the respective impact of SEQUENCING and ASSIGNMENT optimization, proposing new ways to combine the optimization of these two decision sets, and leading to new state-of-the-art algorithms for the CVRP.

References


Real-world picker routing problems in mixed shelves warehouses of online retailers

Maximilian Schiffer\textsuperscript{1}, Nils Boysen\textsuperscript{2}, Gilbert Laporte\textsuperscript{3}

\textsuperscript{1} RWTH Aachen University, Chair of Operations Management, Aachen, Germany. maximilian.schiffer@om.rwth-aachen.de
\textsuperscript{2} Friedrich Schiller Universität Jena, Chair of Operations Management, Jena, Germany. nils.boysen@uni-jena.de
\textsuperscript{3} HEC Montréal, CIRRELT and Canada Research Chair in Distribution Management, Montréal, Canada. gilbert.laporte@cirrelt.ca

Mots-clés: picker routing, scattered storage, retailer warehouses, dynamic programming

Increasing e-commerce leads to rising trade volumes in the business-to-customer (B2C) segment of online retailers. Besides increasing competition between major players, increasing customer requirements call for an efficient operation of an online retailer’s warehouse in the B2C segment. However, large assortments and volatile demands of final customers make predictive analysis of customer behavior or customer orders nearly impossible. Nowadays, warehouses of large retailers are often organized according to the mixed-shelves paradigm. In such a warehouse, incoming unit loads of stock keeping units (SKUs) are purposefully broken down into single items. These items are then stored using a scattered storage principle, i.e., items are scattered all around the shelves (cf. Weidinger and Boysen, 2017). This storage strategy increases the probability that items which are ordered together are stored close to each other. Thus, the basic intention of mixed shelves storages is to considerably reduce the unproductive picker walking time, which in traditional warehouses often consumes up to 50\% or more of the pickers working hours (cf. de Koster et al., 2007; Tompkins et al., 2003). However, breaking down the SKUs and spreading the items randomly over the shelves leads to additional handling effort. If multiple items of the same SKU arise in one order, a picker has to access multiple shelves at various storage positions to assemble all items. Thus, a scattered storage only helps to reduce unproductive picker times if a picker can assemble SKUs to fulfill an order list efficiently.

Herein, optimization can help to increase the efficiency of pickers in such warehouses by designing optimal route plans to collect SKUs from a scattered storage to fulfill an order list. Up to now, exact solution approaches are often limited to small warehouse layouts that do not resemble practical problem sizes or do not consider all necessary flexibility options (e.g., cf. Ratliff and Rosenthal, 1983). So far, only heuristic approaches are capable to address the large-sized layouts that arise in practice (e.g., cf. Theys et al., 2010). Additionally, all publications so far focus on single picking modes or policies.

Against this background, we present the first generic exact algorithm that accounts for different picking modes and different warehouse layouts. In more detail, we account for:

- different warehouse layouts, i.e., single and multiple depots at which completed orders can be dropped and new order lists can be taken.
- different picking modes, i.e., a picker that is accompanied by his cart or a picker that is allowed to leave his cart for a limited subtour.
- parallel processing of multiple order lists.

We present an exact algorithm based on dynamic programming and problem specific reduction techniques. With this algorithm, we are capable of solving real-sized instances and real-world...
warehouse layouts to optimality in only a few seconds. In addition, we show how the optimal sequencing of order lists can be integrated into this algorithm.

With this generic algorithm, we compare the picking policies of different online retailers. Namely, we compare the picking policies of i) Hermes group, Germany’s second largest postal service provider, ii) Amazon Europe, Europe’s biggest online retailer, and iii) Zalando, Germany’s leading online fashion retailer. Using our generic algorithmic framework, we benchmark and analyze the different picking policies and warehouse settings. Besides, we also evaluate additional possible picking policies. Based on these results, we derive managerial insights for practitioners that have to setup a warehouse and that want to plan picking processes.

References


Picker routing in AGV-assisted order picking systems

Michael Schneider
schneider@dpo.rwth-aachen.de
Deutsche Post Chair – Optimization of Distribution Networks
RWTH Aachen University
Germany

Maximilian Löffler
loeffler@dpo.rwth-aachen.de
Deutsche Post Chair – Optimization of Distribution Networks
RWTH Aachen University
Germany

Nils Boysen
nils.boySEN@uni-jena.de
Lehrstuhl für Operations Management
Friedrich-Schiller-Universität Jena
Germany

Christoph Glock
glock@pscm.tu-darmstadt.de
Fachgebiet Produktion und Supply Chain Management
Technische Universität Darmstadt
Germany

AGV-assisted order picking is applied to support human order pickers, e.g., if heavy and bulky items are to be collected from the racks of a warehouse. In the scenario considered in this paper, automated guided vehicles (AGVs) accompany human order pickers during the order picking process, who put the picked items onto the AGV. Once a picking order has been completed, the AGV autonomously drives towards the shipping area and a new AGV for the next order is requested. Thus, the picker does not have to return to a central depot after completing a picking order but continuously picks order after order. This paper addresses the routing of the AGV-assisted pickers through the warehouse as well as the sequencing of incoming orders. Computational complexity is proven and efficient solution procedures are developed. As a main contribution, we provide an exact polynomial time algorithm for the case the order processing sequence is given. Furthermore, we explore the difference of the picker's walking distance with and without AGV support, such that practitioners having to choose among both options receive some decision support.

Keywords: warehousing, automated guided vehicles, AGV, order picking, routing
Tramp ship routing and scheduling with voyage separation requirements

Jesper Larsen, Richard Lusby and Charlotte Vilhelmsen

Department of Management Engineering, Technical University of Denmark.

This presentation addresses a tramp routing and scheduling problem. Tramp ships operate like taxies by following the available demand, as opposed to liner ships that operate like busses on a fixed route network according to a published timetable. Tramp operators determine some of the demand in advance by ensuring long-term contracts. The rest of the demand comes from optional voyages found in the spot market. Routing and scheduling a tramp feet to best utilize feet capacity according to the current demand is therefore an ongoing and complicated problem. We add further complexity by incorporating voyage separation requirements that enforce a minimum time spread between some voyages. We developed a new and exact Branch-and-Price procedure for this problem. A dynamic programming algorithm generates columns, while a novel time window branching scheme is used to enforce the voyage separation requirements. Computational results show that the algorithm finds optimal solutions very quickly for the vast majority of test instances. We compare the results with two earlier published methods and show that our Branch-and-Price approach outperforms both an a priori path generation method and an Adaptive Large Neighbourhood Search heuristic.
Ship routing and speed optimization with heterogeneous fuel consumption profiles

Gabriel Homsi¹, Rafael Martinelli², Thibaut Vidal¹, Kjetil Fagerholt³

¹ Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Informática, Brazil
ghomsi,vidalt@inf.puc-rio.br

² Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Industrial, Brazil
martinelli@puc-rio.br

³ Norwegian University of Science and Technology, Department of Industrial Economics and Technology Management, Norway
kjetil.fagerholt@ntnu.no

Keywords: Ship routing, Speed optimization, Metaheuristics, Resource allocation problem, Nested constraints

1 Introduction and Problem Definition

We study an Industrial and Tramp Ship Routing and Scheduling Problem (ITSRSP) where the speed on each sailing leg (route segment) is a decision variable, and fuel consumption is a convex function of speed and payload [4]. Routing models often assume a constant speed, resulting in several limitations. For example, different speeds can significantly change the travel cost and time-window feasibility of a solution. Therefore, without speed as a decision variable, it is not possible to efficiently explore the trade-off between routing decisions and fuel consumption. This is more evident in ship routing scenarios, as shipping operators may engage in slow steaming practices to reduce fuel costs [5]. The influence of payload on fuel consumption extends previous work [3], and results in leg-dependent fuel consumption functions, as the load on board of the ship changes at every port visit. We refer to the studied problem as the ITSRSP with Speed Optimization (ITSRSPSO). It can be viewed as an extension of the Pickup and Delivery Problem with Time Windows with a heterogeneous fleet, ship-cargo compatibility constraints, different ship starting points and starting times, service flexibility with penalties, and speed optimization.

We now define the ITSRSPSO. A tramp or industrial shipping operator has a fleet of $m$ ships. Let $G = (N, A)$ be a complete directed graph where $N$ is the union of a set of pickup nodes $P = \{1, \ldots, n\}$, delivery nodes $D = \{n+1, \ldots, 2n\}$, and starting points $\{0_1, \ldots, 0_m\}$. There are $n$ cargoes available. Each cargo $i \in \{1, \ldots, n\}$ consists of transporting a load of size $q_i$ from a pickup $i \in P$ to a corresponding delivery location $n+i \in D$. Every node $i \in P \cup D$ is associated with a time window of allowable visit times $[a_i, b_i]$. A ship $k \in \{1, \ldots, m\}$ is initially located at $0_k$, has capacity $Q_k$, and is first available at time $s_{0_k}^0$. It can traverse an arc $(i, j)^k \in A$ at any speed $v \in [v_{\text{min}}^k, v_{\text{max}}^k]$ with a cost $f_{ij}^k(v, u) = \frac{1}{24} \mu_k v^2 t_{ij}^k (0.8 + 0.2u)$, where $\mu_k$ is a ship-specific constant, $t_{ij}^k$ is the arc length, and $u$ is the proportion of maximum load on board the ship before arriving node $j$. For every combination of ship $k$ and node $i \in P \cup D$, there is an associated service (port) cost $s_{ik}^0$ and duration $s_{ik}^0$ for visiting $i$. There might be incompatibilities between ships and cargoes. A penalty $p_i$ is paid if a cargo $i$ is not transported.
by the fleet. The objective of the ITSRSPSO is to minimize total travel cost plus the associated penalties in the case where some cargoes are not transported.

2 Solution Methodology

To solve the ITSRSPSO, we propose a Hybrid Genetic Search (HGS), a non-trivial extension of the Unified Hybrid Genetic Search of [6]. The proposed HGS includes a set-partitioning-based large neighborhood and uses problem-tailored crossover and local search operators to cover multiple ITSRSPSO attributes which were not included in the original framework. Being a hybrid metheuristic, the HGS combines the exploration capabilities of genetic algorithms with efficient local search improvement procedures. The search space of the HGS considers penalized infeasible solutions to enhance the search towards high-quality feasible solutions, penalizing the maximum load violation, time-window infeasibility (using “returns in time” for constant-time evaluations of infeasible routes [7]), and number of incompatible cargoes. Several components of the HGS ensure a balance between solution quality and population diversity, and the local search procedure also employs granular search techniques to reduce the size of the neighborhoods.

The speed of each leg is jointly optimized on every local search move evaluation. If a neighbor solution is time-window infeasible, the move is evaluated in \(O(1)\) using preprocessed information and route concatenation techniques. For this scenario, travel cost is obtained from the fuel consumption functions assuming maximum ship speed and full load. Otherwise, if a neighbor solution is time-window feasible, the optimal speed for each leg is found by solving a Resource Allocation Problem with Nested Constraints (RAP-NC) using the \(O(n \log m \log \frac{nm}{e})\) Monotonic Decomposition Algorithm (MDA) of [8]. Ships have a minimum and maximum cruising speed. Therefore, they may wait on an early arrival due to a combination of speed bounds and time windows. This leads to fuel consumption functions that are not strictly-convex (if \(v < v_{\min}^k\), \(f_{ij}^k(v, u) = f_{ij}^k(v_{\min}^k, u)\)) in the RAP-NC formulation. Fortunately, the MDA does not require strict convexity and therefore is able to account for waiting times.

3 Computational Experiments and Conclusions

We implemented the HGS in C++ and conducted our experiments on an i7-3960X CPU. We compared a version of HGS without speed optimization (HGS-W) with several Adaptive Large Neighborhood Search (ALNS) variants from the literature on a set of 240 ITSRSP benchmark instances based on real-life scenarios [2, 1]. Then, we adapted these instances to evaluate the impact of joint speed optimization. We compared our HGS with joint speed optimization (HGS-J) with a version of HGS that assumes a fixed speed and optimizes speed once at the end (HGS-O). Comparison with previous literature is shown in Table 1 and Table 2, while speed optimization results are shown in Table 2. Table 1 provides summary results over a subset of the instances evaluated by [2], and the results in Table 2 are grouped according to problem topology (short sea, deep sea) and cargo type (mixed load, full load). Average values for gap (in percentage) to previously best known solution and CPU time (in minutes) are respectively given by “Gap” and “T”.

The HGS-W significantly outperforms all ALNS variants from previous literature, becoming the new state-of-the-art heuristic for the ITSRSP. The average solution quality of HGS-J is largely better than HGS-O, with a moderate increase (5.7\times) of computational effort. Given the complexity of a tight integration of speed optimization and routing decisions in a metaheuristic,
the observed CPU time is very satisfying. Overall, this work demonstrates that a joint speed and routing optimization is highly profitable from an operational cost perspective, and is made possible without too much overhead when using an efficient linearithmic algorithm such as MDA.

References


ABSTRACT

1. Introduction

The literature on liner shipping (see Meng et al. (2014) for a recent survey) is rich with many models on the problem of optimizing containership speed, frequently combined with fleet deployment, fleet size and mix, network design and other related facets of the problem. Many of these models assume a fixed service frequency, typically one call per week. In this paper we consider flexible service frequencies that belong to a broader set than the standard assumption of one call per week. This may conceivably yield better solutions. Models also typically assume a fixed revenue for the ship operator and usually minimize costs. This treatment ignores the freight rate’s influence upon the model’s results. In that sense, such models do not capture a fundamental aspect of shipping market behavior, that ships tend to speed up in periods of high freight rates and slow down in depressed market conditions. This paper develops a simple model for a fixed route scenario which, among other things, incorporates the influence of freight rates, along with that of fuel prices and cargo inventory costs into the overall decision process. The objective to be maximized is the operator’s average daily profit. The impact of the line’s decisions on CO₂ emissions is also examined and illustrative runs of the model are made on three existing services.

2. Methodology

The model assumes without loss of generality a fleet of $N$ identical containerships deployed on a given fixed route. The objective to be maximized is the average per day profit of the carrier.

Inputs to the problem include:

- The route geometry, represented by a set of ports and a set of legs representing the route.
- The lengths of each leg of the route.
- The freight rate of transporting a TEU from a port on the route to another port on the route, for all relevant port pairs.
- The demand in TEUs from a port on the route to another port on the route, for all relevant port pairs.
- The bunker price.
- The daily operating costs of each vessel, other than fuel.
- The daily at sea fuel consumption function as a function of ship speed.
- The daily at port fuel consumption.

---

1 Corresponding author. hnpas@dtu.dk
The average monetary value of ship cargo on each leg of the route.
- The operator’s annual cost of capital.
- The time spent at each port.
- The cargo handling cost per TEU.
- The capacity of each vessel.
- The minimum and maximum allowable ship speeds.

The problem’s main decision variables are:
- The ship speeds along each leg of the route.
- The service frequency.
- The number of ships $N$ deployed on the route.

We formulate a non-linear optimization problem maximizing the operator’s average daily profit. Note that, in contrast to most liner shipping optimization problem formulations, our objective function is the maximization of a ratio, that of total route profit divided by the total duration of the route. Both numerator and denominator of the ratio are nonlinear functions of speed, and of course so is the ratio itself.

The above non-linear optimization problem has three main decision variables, as defined above. A constrained version of the above problem is if the schedule frequency is fixed, that is, is considered an exogenous input and cannot vary freely. In fact for liner services it is typically expected that the service period $t_0$, defined as the reciprocal of the service frequency, can take on only prespecified values, the most common of which is 7 days for a weekly service. In theory, other values of $t_0$ can also be considered (eg $t_0=14$ corresponds to a biweekly service, and $t_0=3.5$ is a service twice a week), but this is not very common. Almost unheard of is the case that $t_0$ may take on other values, such as for instance 6, 8, 9, or even fractional values. However, as liner services schedules are published in each carrier’s web site and other media well in advance, there is really nothing fundamental that prevents a carrier from setting up a service with $t_0$ equal to any prescribed value, if these ‘unconventional’ service frequencies happen to achieve better results for the carrier. Whatever it is, it is obvious that the constrained version of the problem ($t_0$ fixed and equal to 7) will not achieve better results vis-à-vis the case in which $t_0$ is allowed to vary freely, or is restricted to a wider range or set of values. In that sense, a fixed $t_0$ will generally come at a price.

The non-linear optimization problem was solved by linearizing the objective function, coding the model in MATLAB and using an Excel spreadsheet solver. Details can be found in Giovannini and Psaraftis (2018).

3. Application

We examined the following three actual liner routes:

- **AE2** - North Europe and Asia: such service links Asia to North Europe and is provided by Maersk. The same service is also provided by MSC under the name SWAN. Indeed, both Maersk’s ships and MSC’s ships are deployed along this route.
- **TP1** - North America (West Coast) and Asia: the route connects Asia to the West Coast of North America. Maersk offers this service, however the same service is also provided by
MSC and it is called EAGLE. As for the AE2 service, along the TP1 route are deployed Maersk’s vessels as well as MSC’s vessels.

- **NEUATL1** - North Europe and North America (East Coast): the NEUATL1 lane links North Europe to the US East Coast. The service is furnished by MSC or similarly by Maersk under the name TA1.

For these routes we analyzed three different cases:

**First case:** the service frequency is constant and the number of ships is variable. Therefore the main decision variables in such case are two, the speeds and the number of deployed vessels.

**Second case:** the number of ships is constant and the frequency is variable. Hence the main decision variables are again two, the speeds and the service frequency.

**Third case:** both the frequency and the number of ships are variable, in which case the main decision variables are three. However, in this case the number of ships is bounded from above. This bound is imposed because otherwise the optimal number of ships may reach unrealistic values.

Hereby we only show a limited set of the results concerning the second of the above cases, namely the case where the number of ships is constant but the frequency is variable. The number of ships concerning the “base scenarios” is the actual number of ships employed on the route involved in the examined routes; these are 10 ships for the AE2 route and 5 ships for each of the TP1 and NEUATL1 routes.

Figure 1 depicts the service frequency’s trend and the average speed’s trend at 8 different freight rate values for the route TP1. It can be seen that if the freight rate is low enough, a service period of 8 days is better, whereas for higher rates a service period of 6 days is better.

![Figure 1: Fixed number of ships scenario, optimal service period and optimal average speed at different average freight rates (route TP1)](image)

The above means that if we force \( t_0 = 7 \), the solution will be suboptimal in 7 out of the 8 instances. We can compute the fleet-level difference \( \Delta \) in the objective function between the
optimal solution and the solution in which \( t_0 \) is forced to be equal to 7. This is shown in Table 1:

**Table 1: Cost of forcing a weekly schedule in a fixed number of ships scenario.**

<table>
<thead>
<tr>
<th>Instance</th>
<th>Optimal ( t_0 ) (days)</th>
<th>( \Delta ) (USD/day)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td>4,132</td>
</tr>
<tr>
<td>2</td>
<td>7</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>15,717</td>
</tr>
<tr>
<td>4</td>
<td>6</td>
<td>35,029</td>
</tr>
<tr>
<td>5</td>
<td>6</td>
<td>54,341</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>73,653</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>92,965</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>131,590</td>
</tr>
</tbody>
</table>

At the low end of the freight rate spectrum (instance 1), the model chooses an 8-day service period as optimal and a low corresponding average speed, 15.02 knots. If one forces a higher frequency (and specifically a call every 7 days) and the number of ships is constant, this would only be achievable if the average speed increases to 17.63 knots. The higher frequency would increase the amount of cargo transported and associated revenue, but as the freight rate is low the additional revenue cannot match the increased cost due to the higher speed, hence daily profit for the fleet is lower by 4,132 USD/day.

The situation at the high end of the freight rate spectrum (instance 8) is the opposite, but its effect the same. At the last instance, the high average rate of 1,001 USD/TEU suggests a 6-day service period as optimal and a high corresponding average speed, 21.32 knots. If one forces a lower frequency (a call every 7 days) and the number of ships is constant, this would only be achievable by a lower ship speed, again 17.63 knots. The lower frequency would decrease cargo transported, but given the freight rate is high the associated loss of revenue would be greater than the savings in fuel cost due to the lower speed, hence again a lower daily profit (in this instance lower by 131,590 USD/day for the entire fleet). The situation in instances 3 to 7 is similar.

More results will be presented in the talk and can be found in Giovannini and Psaraftis (2018).

**REFERENCES**


Demand responsive public transport scheduling: Dial-a-ride versus Pick up and Delivery with Time Windows

Evelien van der Hurk
Management Science, DTU Management Engineering
Technological University of Denmark
Tel.: +45 45 25 48 21
evdh@dtu.dk
December 22, 2017

Demand responsive transport services are currently employed in the context of rural areas to compliment general public transport, as a service to elderly, to persons with a disability, or to persons with an illness. Dynamic versions of these problems could be envisioned to play a more central role in a future that includes shared mobility through e.g. mobility as a service concepts; as well as automatic driving vehicles.

This class of problems generally relate to one- to-one pick up and delivery problems either with time window constraints on pick up and delivery (PDPTW), or with additional constraints on the driving time per customer or commodity (dial-a-ride-problem, DARP) (Berbeglia et al., 2007). The past decades have produced several fast algorithms that provide high quality solutions to both of these problems. Cordeau and Laporte (2007) note that for the static DARP many excellent algorithms exist; the more recent overview in (Molenbruch et al., 2017) supports this view and remarks that the impact of different assumptions on the solution speed, and on the quality of solutions has received little to no attention.

This research focusses on the impact the model formulation of time window constraints has on solution quality and computation time for a real life demand responsive public transport service. Specifically, it investigates the impact on including maximum driving times as time window constraints only (PDPTW); or with the addition of maximum drive time constraints (DARP). Solutions are evaluated in terms of operating costs; in-vehicle riding time; and journey time as the time between latest arrival time (earliest pick-up time) and pick-up (drop off) for inbound trips (outbound trips)

The research is motivated by the real life case of a Swedish demand responsive public transport provider that contracts a mixed fleet consisting of different types of mini busses and taxis. All requests are made at least a day in advance. Over 90% of their customers are schoolkids who require transport from and to school either because no general public transport is available close to their homes, or because due to personal circumstances they cannot use regular public transport. About 6% of the requests requires a special vehicle type (e.g. for a wheelchair), and about 3% of customers have special needs: e.g. a need for a specific vehicle type, a need to be traveling always on the direct route and be dropped off first, a need to travel alone, or a need to travel together with a companion. On a typical day over 3500 requests are scheduled.
Their problem is close to Jaw et al. (1986) with the difference that rather than time windows, deadlines are specified in the form of a latest arrival time for the outbound trip (school start), and an earliest pick-up time for the inbound trip (school end). Furthermore, journey time is limited through an age-dependent maximum. This allows to translate the problem directly to a PDPTW. Note that this maximum travel time restriction is independent of the minimal travel time between origin and destination for each trip. Therefore, this formulation may lead to "unfair" solutions were children who need to travel a short distance may spend more additional minutes in the vehicle than children that live far away. Also, the formulation limits the number of customers that can be combined especially on long trips, where it may be most beneficial to cluster customers together. Therefore, a dial-a-ride formulation with a restriction on maximum travel time specified as an additional budget on top of the minimal travel time could increase fairness of solutions and may even reduce operating cost.

The operator is a non-profit organization that on behalf of the government aims to provide a good customer service cost-efficiently. To this end, they would be interested in not only efficient algorithms that can improve their current schedules, but also changes in policy that could improve either customer service level or cost-efficiency. The formulation of maximum travel times as either time windows or drive times is therefore of interest. Formulations for both DARP and PDPTW will be solved to compare the influences of the difference in formulation.

References


Routing of small parcels in a crowd-sourced network

Raviv Tal, Tenzer Z. Eyal
Department of Industrial Engineering, Tel Aviv University, Israel
Email: talraviv@eng.tau.ac.il
December 15, 2017

1 Introduction and problem definition

The rapid growth of e-commerce has caused a significant increase in the volume that passes through the parcel delivery industry. Recent advances in the mobile computing technology created new opportunities to redesign the delivery process to make it more efficient and sustainable. We envision a logistic process for delivering small parcels by people who subscribe as occasional curriers (OCs). A network of automatic service points (SPs) is deployed. The SPs are used as locations where parcels can be dropped-off by their senders and picked up by their recipients. Also, the SPs can serve as intermediate transfer points.

Similar schemes were presented by [1] who proposed using occasional drivers to deliver parcels when there is a good match between his itinerary and the origin and destination of the driver. [2] introduced a scheme based on journeys made by taxi drivers, where a parcel is picked before and dropped-off after passengers trips, in nearby locations.

The operation of such a logistic network raises many challenges: On-line decisions about which OC should be called for service and which parcels she should be asked to carry along her journey should be made. This abstract focuses on finding an optimal policy for these decisions. The policy optimization problem is formulated as a stochastic dynamic problem (SDP) and an efficient algorithm to solve this SDP is presented.

We consider a system with \( n \) SPs. A set of OC journeys is given. Each journey is a sequence of SPs. The travel time between a pair of SPs is denoted by \( t_{ij} \). There is a known Poisson process that governs the arrival of OCs to the beginning of each such journey. \( C_i \) denotes the set of SP sequences that pass but not end at \( SP_i \). Each member of this set represents the sequence of SPs made by OCs after visiting \( SP_i \) (including \( SP_i \)). \( C^k_i \) denotes the \( k \)th such sequence. Note that \( C^k_i \) is a result of the unification of all the journeys that pass via \( SP_i \) and continue along the same sequence of SPs. The rate of OCs that follows the sequence \( C^k_i \) is denoted by \( \lambda^k_i \). Finally \( \lambda_i = \sum_k \lambda^k_i \) is the arrival rate of OCs at \( SP_i \). Each parcel in the system is characterized by its origin and destination SPs.

Since the goal is to deliver the parcels as quickly as possible, the system is charged for any unit of time in which a parcel is not yet at its destination. We refer to this cost as “holding cost” and denote it by \( h \). In addition, the system is charged for each operation made by the OCs. We denote the payment made to the OC for transferring a parcel from \( SP_i \) to \( SP_j \) (in a single leg) by \( r_{ij} \). We refer to this component as “handling cost”. The objective is to minimize the total expected cost by assigning parcels to OCs and deciding where they should drop them.

We make the following assumptions: 1) An OC can carry an arbitrary number of parcels and stop at an arbitrary number of SPs along her journey. 2) the capacity of the SPs is not binding. Under these assumptions, the decisions described above can be made separately for each parcel based on its current location and destination only.
2 Solution method

The policy optimization problem described above can be formulated as a Markov Decision Process for each possible destination and solved using standard methods such as policy iteration. However, the convergence of such a procedure is slow. In this abstract, we present a Dijkstra-like solution approach that is based on stochastic dynamic programming and solve the problem accurately and much more efficiently.

The state space consists of two types of states. The first represents situations in which a parcel is in $SP_i$ and no OC passes by it. The second represent situations in which the parcel is in an SP that is visited by an OC with a known journey. We denote the states of the first type by $S_i^0$ and of the second by $S_i^k$ where the index $i$ refer to the current location of the parcel ($SP_i$) and the index $k$ to the index of the sequence. The decision space is a singleton for the $S_i^0$ states (just wait for an OC to arrive) and $C_i^k$ for the $S_i^k$ states, which is to select for which SP the parcel should be carried. Recall that $SP_i \in C_i^k$, which means that it is always possible not to send the parcel with the OC. The Bellman equations are

$$ v(S_i^k) = \min_{j \in C_i^k} \{ht_{ij} + r_{i,j} + v(S_j^0)\}, \quad (1) $$

and

$$ v(S_i^0) = \min_{K \subset \{1, \ldots, |C_i|\}} \left\{ \frac{h}{\sum_{k \in K} \lambda_i^k} + \sum_{k \in K} \frac{\lambda_i^k}{\sum_{l \in K} \lambda_i^l} v'(S_i^k) \right\}. \quad (2) $$

Assuming, without loss of generality, that the destination of the parcel is $SP_i$, the boundary condition can be stated as $v(S_i^0) = 0$. This represents the fact that there is no cost that is associated with a parcel that is already located at its destination.

The challenges in calculating the above Bellman equation are twofold. First, it is not clear in what order the equitations can be evaluated, such that all the relevant values of other states are available when needed. Second, the evaluation of (2) requires optimizing over the exponentially large collection of all the subsets of $\{1, \ldots, |C_i|\}$. We overcome the first challenge by keeping a set of SPs with the smallest values and gradually adding new SPs in increasing order of value. Thus, at each iteration there is at least one $S_i^0$ state that can be evaluated. The set is initialized with the destination, whose value is zero. At each iteration, the optimal policy is tentatively calculated for all the yet unknown SPs and their corresponding decisions nodes, based on the current values. The SP with the smallest calculated value must be a one from which, under an optimal policy, parcels are transferred only to SPs in the known set. Thus its value is calculated for all the yet unknown SPs and their corresponding decisions nodes, based on the current values. The SP with the smallest calculated value must be a one from which, under an optimal policy, parcels are transferred only to SPs in the known set. Thus its value is calculated for all the yet unknown SPs and their corresponding decisions nodes, based on the current values. The SP with the smallest calculated value must be a one from which, under an optimal policy, parcels are transferred only to SPs in the known set. Thus its value is calculated for all the yet unknown SPs and their corresponding decisions nodes, based on the current values. The SP with the smallest calculated value must be a one from which, under an optimal policy, parcels are transferred only to SPs in the known set. Thus its value is calculated for all the yet unknown SPs and their corresponding decisions nodes, based on the current values. The SP with the smallest calculated value must be a one from which, under an optimal policy, parcels are transferred only to SPs in the known set. Thus its value is calculated for all the yet unknown SPs and their corresponding decisions nodes, based on the current values. The SP with the smallest calculated value must be a one from which, under an optimal policy, parcels are transferred only to SPs in the known set. Thus its value is calculated for all the yet unknown SPs and their corresponding decisions nodes, based on the current values.
Let $v(S^n_0) = 0$;

for $i = 1, \ldots, n - 1$ do Let $v(S^0_i) = \infty$;

Let $Q = \{1, \ldots, n - 1\}$;

while $Q \neq \emptyset$ do

for $i \in Q, k = 1, \ldots, |C_i|$ do $v(S^k_i) = \min_{j \in C^k_i} \{ht_{ij} + r_{ij} + v(S^0_j)\}$;

for $i \in Q$ do

Sort the sequences in $C_i$ increasing order of $v(S^k_i)$;

Let $v(S^0_i) = \min_{k' \in \{1, \ldots, |C_i|\}} \left\{ \frac{h}{\sum_{k=1}^{k'} \lambda^k_i} + \sum_{k=1}^{k'} \frac{\lambda^k_i}{\sum_{l=1}^{k'} \lambda^l_i} v(S^{[k]}_i) \right\}$;

end

Let $Q = Q \setminus \arg\min_{i \in Q} v(S^0_i)$;

end

Algorithm 1: Dijkstra like algorithm for the parcel routing problem

3 Results and conclusions

The parcel routing policy derived from the above dynamic program was tested in simulation based on realistic data about movements of people and parcels with 50 SPs located in Tel Aviv area and up 8000 OCs and 16000 parcels a day. The results demonstrate that such a scheme can be used to provide a reliable next day delivery service at modest cost (stems from the payments to the OCs).

In an ongoing study, we are checking the sensitivity of the scheme to our simplifying assumptions about the capacity constraints and the willingness of the OCs to serve all the requests. In particular, we run an extensive simulation study to test a reward mechanism that includes compensation both for handling each parcel (as in the DP model) and for each stop made by the OC. The capacity constraint and the more complex OC reward scheme require optimizing request made to the OCs to allow better consolidation and a smaller number of stops. Our preliminary simulation study indicates that it is possible to meet a daily demand for 16000 parcels with daily 8000 OC journeys, which represent a tiny fraction of the number of journey made in the city. Assuming that the vehicle capacity of each OC is 50 parcels and the capacity of the SPs is 400 parcels our simulation demonstrate the capability of our logistic model to provide a reliable same day delivery service in a fraction of the cost currently charged in the market. This small cost is achieved while rewarding the OCs for their effort and loss of time with incentives that are more than five times the average hourly wage. Therefore we suggest that such a crowd-sourced parcel delivery scheme has a potential to economize and reduce the ecological footprint of the small parcel delivery industry.

References


Emergency Medical Services – A practical application of the Vehicle Routing Problem

Martin Wahl Andersen
LOGIS A/S

Abstract: Emergency Medical Services (EMS) have traditionally focused on response time performance for high acuity emergency calls, which basically only relies on a shortest path algorithm to identify the closest vehicle. However in recent years, more attention has been given to the fact that only a very few patients have an outcome dependent on the response time. Instead focus is shifting to increase utilization of resources for lower acuity calls to increase overall coverage of the service area, or the service preparedness. Additionally there is a trend globally for EMS operators to start expanding the number of outpatient services provided to also include mobile integrated health and patient inflow management.

The expansion of EMS services into preemptive medicine as well as being the gatekeeper for the patient inflow makes the use of Operations Research more important than ever, since the number of scheduled tasks within the individual EMS providers grows rapidly while there is a constant focus on utilizing available resources optimally to keep the growing health care costs down. It is important to note, that EMS in itself carries a very small cost in health care compared to the operation of hospitals etc., but a well performing EMS system can have a great impact in the hospital system itself by improving hospital ability to manage beds and waiting lists.

Examples of services provided by progressive EMS systems today are:
- Ambulance dispatch for emergency calls
- Stretcher based non emergency patient transfers
- Wheelchair/sedan non emergency patient transfers
- Home visits by physicians for patients who are not able to see their private physician
- Social and psychiatric responses
- Mobile X-Ray and Blood tests
- Organ transfers
- Preemptive follow up visits for discharged patients
- Phone triage and counseling
- Scheduling ED patient inflow

Naturally some of these services require crew and fleet to have a diverse set of skill levels and capacities, but many of the services can also be performed across these skill levels, presenting a unique opportunity to optimize the use of resources across the services provided. As much of the work is scheduled between 1 hour and sometimes days in advance, the vehicle routing problem is probably one of the most important optimization problems to address within EMS. However in a high performance setting like an EMS system, there are an incredible number of constraints that must be taken into account to provide a feasible solution, such as:
- Soft Time Windows for either pickup or drop off
- Maximum time in transit for ensuring patients are not picked up or dropped
- Operating areas allowing resources to service a geography or to keep resources close to home
- Credentials allowing either physical units and/or crew to either pickup and/drop off patients in various areas
- “Wrapped units” where certain units are preferred for certain facilities
- End of shift rules allowing overtime for certain types of calls
- Required Level of Service
- Unit capacity with capacity substitution
- Creating an “obvious” solution that looks logical
Besides the constraints on the problem the challenge can also be the overall size, where systems spanning all of Europe or all of USA can encompass millions of calls a year and several thousand active resources to be optimized on relatively limited hardware resources. For a high performance system like EMS to work these optimization problems must be solved in milliseconds as results are presented to users in real time.

This talk will focus on the implementation aspect of using Operations Research in a high performance environment, where a multitude of soft and hard constraints play a large role in dictating possible solutions. In addition an introduction will be made to the complimentary problems that coexist with the vehicle routing problem within EMS. Deployment strategies often differ based on the acuity of the call and the level of service provided by resources. This poses an interesting challenge when determining the tradeoff between optimal utilization of resources and maximum geographical coverage for certain levels of service. An example is an EMS system where the same resources can respond to both non emergency scheduled calls and emergency calls, where it is important to limit the utilization of resource for non emergency to preserve room for sudden emergency calls. In other words, where to make room on the resources depending on the geographical distribution of future expected emergency demand.

Logis Solutions is one of the top tier providers of Computer Aided Dispatch software globally with customers in Australia, New Zealand, Europe and USA within both the public and private sector. Rene Munk Joergensen has a Ph.D in demand responsive transportation from DTU, and was an associate professor in ITS at DTU prior to joining Logis Solutions as one of 3 partners in 2006.

Med venlig hilsen
Time-Dependent Vehicle Routing Problem with Time Windows and road network information

Hamza Ben Ticha$^1$, Nabil Absi$^1$, Dominique Feillet$^1$, Alain Quilliot$^2$, Tom van Woensel$^3$

$^1$ École des Mines de Saint-Etienne and UMR CNRS 6158 LIMOS CMP F-13541 Gardanne, France
{hamza.ben-ticha,absi,feillet}@emse.fr

$^2$ LIMOS Institut Supérieur d’Informatique de Modélisation et leurs Applications, Aubière, France
{alain.quilliot}@isima.fr

$^3$ École de génie industriel, Université de technologie d’Eindhoven, Eindhoven, Pays-Bas
T.v.Woensel@tue.nl

Mots-clés : TDVRP, road network, branch-and-price

1 Introduction

Vehicle routing problems (VRPs) have been widely studied in the operations research literature. Many variants have been formulated in order to correctly model real-life applications. One of these variants is the Time-Dependent Vehicle Routing Problem (TDVRP) where travel times vary over the time. These variations may be due to traffic congestion, weather conditions and vehicle breakdowns.

Typically, vehicle routing problems, including time-dependent variants, are tackled using a complete graph representation, called customer-based graph. In this representation, a node is introduced for every point of interest (depot and customer locations) and arcs represent the best paths linking these points. This modelling approach relies on the assumption that the best path between two points in the original road network can be easily defined. In many situations, this assumption is seldom valid. One of these situations is when several attributes are defined on road segments. In this case, alternative paths with different compromises could exist between two points in the road network. Discarding these alternatives when tackling the vehicle routing problem could have a negative impact on the solution quality [3].

In the literature, an increasing number of papers investigate the limits of the customer-based graph representation for non-time-dependent vehicle routing problems. It has been shown that transforming the original road network into a customer-based graph could result on losing optimality [3]. Two approaches are proposed to handle this issue. The first approach consists in representing the road network with a multigraph. In this representation, all alternative paths between two points of interest are considered and maintained when solving the problem [3]. The second approach consists in tackling the problem on a graph that mimics the original road network [2], so-called road-network graph.

In this paper, we propose to investigate the impact of the traditional complete graph representation on the solution quality for the time-dependent vehicle routing problems (TDVRPs). Due the time-dependency, it is difficult or even intractable to represent the road network with a multigraph. Indeed, one would have to compute the set of non-dominated Pareto-optimal paths for each pair of points of interest at each possible departure time, which induces the solution of many NP-hard problems. For this reason, we propose to solve the problem directly on the road network graph. We consider the Time-Dependent Vehicle Routing Problem with Time Windows (TDVRPTW) as test-bed problem and we develop a branch-and-price algorithm that works on the road-network graph.
2 Branch-and-price algorithm for the TDVRPTW

2.1 Column generation

In the road network based column generation for the TDVRPTW, the master problem consists in a set covering formulation as for the standard problem. The pricing problem aims at finding new routes offering better ways to serve customers, i.e., with negative reduced costs. The pricing problem can be reduced to a Time Dependent Shortest Path Problem with Resource Constraints (TDSPPRC) in the road network. It is important to emphasize that, in the road-network graph, the elementary path condition has to be modified. Indeed, in an optimal solution an arc of the road network can be traversed several times and a customer node can be visited several times. To solve the TDSPPRC, we develop a time-dependent labeling algorithm which is a modification of the labeling algorithm proposed in [1]. Basically, this algorithm performs as follows. A subpath from the depot to a node \( i \) is represented using a label. At each iteration, all labels at a certain node \( i \) are extended along arcs \((i, j)\) in the road-network graph. When the destination node \( j \) is a customer node, two extensions are processed. In the first extension, the service at the customer \( j \) is performed, if it is feasible. In the second extension, the node is only visited without servicing the associated customer.

2.2 Branching scheme

The branching rule aims at eliminating the current fractional solution by adding constraints and partitioning the solution space. Using the road network setting, the standard branching rule used for vehicle routing problems is not suitable. The reason is that in customer-based graph every arc is traversed at most once in a feasible solution. This property does not hold anymore in a road-network graph. To handle this issue, we propose a specific branching scheme that performs as follows. When the flow is fractional, we branch on an arc flow. But when the flow is integer on all arcs and the routing solution is fractional, we derive two branches. In the first branch, we enumerate all feasible routes in the sub-graph induced by the flow, then, we solve a set partitioning problem based on the obtained set of routes. In the second branch, we eliminate the current solution by enforcing the use of an arc that is not in the sub-graph supported by the flow.

3 Computational experiments

In order to achieve comprehensive conclusions, we compare solutions obtained in the road network to those obtained on two customer-based graphs : a min-cost graph where an arc represents the shortest path between two points of interest and a min-time graph where paths are selected according to the travel time. Computational experiments carried out on a large set of instances show the negative effects of the customer-based graphs against a complete consideration of road-network information. Attractive savings in terms of the solution cost are obtained when tackling the problem using the road network approach.

Références

Routing problems, multi-commodity fixed-charge network design and variable splitting

Jens Vinther Clausen, Richard Lusby, Stefan Ropke (speaker)

December 22, 2017

The multi-commodity fixed-charge network design problem (MCFCND) can be defined on a graph \( G = (V, A) \) with vertices \( V \) and arcs \( A \). Let \( K \) be a set of commodities that have to be transported and \( d_k \) be amount of commodity \( k \) that that needs to be transported. We assume that each commodity has a single origin node and a single destination node denoted \( s(k) \) and \( e(k) \), respectively. For each node \( i \in V \) and commodity \( k \in K \) we define

\[
d(i, k) = \begin{cases} d_k & \text{if } i = s(k) \\ -d_k & \text{if } i = e(k) \\ 0 & \text{otherwise} \end{cases}
\]

For each vertex \( i \in V \) we define the sets \( \delta^+(i) = \{(j, j') \in A : j = i\} \) and \( \delta^-(i) = \{(j, j') \in A : j' = i\} \). The model uses decision variables \( x_{ak} \) that specify the amount of commodity \( k \in K \) that ows through arc \( a \in A \). Flow is only possible if the arc is “open”. Variable \( y_a \in \{0, 1\} \) is one if we choose to open arc \( a \). An open arc \( a \in A \) has capacity \( u_a \). There is a cost \( f_a \) associated with opening an arc \( a \) and a cost \( c_{ak} \) for sending one unit of commodity \( k \) along arc \( a \). The model for the MCFCND is

\[
\text{(M1)} \quad \min \sum_{k \in K} \sum_{a \in A} c_{ak} x_{ak} + \sum_{a \in A} f_a y_a
\]

subject to

\[
\sum_{a \in \delta^+(i)} x_{ak} - \sum_{a \in \delta^-(i)} x_{ak} = d(i, k) \quad \forall i \in V, k \in K
\]

\[
\sum_{k \in K} x_{ak} \leq u_a y_a \quad \forall a \in A
\]

\[
x_{ak} \geq 0 \quad \forall a \in A, k \in K
\]

\[
y_a \in \{0, 1\} \quad \forall a \in A
\]

The MCFCND is a versatile model and many vehicle routing and traveling salesman problems can be modelled as a MCFCND with extra constraints. If we for example wish to model the asymmetric traveling salesman problem (ATSP) with \( n \) cities we add the constraints

\[
\sum_{a \in \delta^+(i)} y_a = 1 \quad \forall i \in V
\]

\[
\sum_{a \in \delta^-(i)} y_a = 1 \quad \forall i \in V
\]

and set

\[
K = \{1\}, d(i, k) = \begin{cases} n - 1 & \text{if } i = 1 \\ -1 & \text{if } i \in V \setminus \{1\} \end{cases}
\]

\[\]
and \(c_{a1} = 0\) for all \(a \in A\). We set \(f_a\) to be the cost of traversing arc \(a \in A\) and \(u_a = n - 1\). In the 
precedence constrained ATSP (PCATSP) we are given a start node (say node 1) and set of precedence 
relations \(P \subseteq V \times V\) where each element \((i, j)\) indicates that node \(i\) has to precede node \(j\). The ATSP 
model presented above can be extended to the PCATSP by adding the constraints

\[
\sum_{a \in \delta^-(i)} x_{a1} \geq \sum_{a \in \delta^-(j)} x_{a1} \quad \forall (i, j) \in P
\]

Consider the capacitated vehicle routing problem (CVRP) where \(V = \{0, 1, \ldots, n\}\) is the set of nodes, 
node 0 is the depot and the remaining nodes are customers. We let \(Q\) indicate the vehicle capacity, \(q_i\) the 
demand of customer \(i \in V \setminus \{0\}\), and we assume that the number of vehicles can be freely chosen. 
With this we can model the problem as a MCFCND by adding constraints

\[
\sum_{a \in \delta^-(i)} y_a = 1 \quad \forall i \in V \setminus \{0\}
\]

\[
\sum_{a \in \delta^-(i)} y_a = 1 \quad \forall i \in V \setminus \{0\}
\]

and we set

\[
K = \{1\}, d(i, k) = \begin{cases} \sum_{i=1}^{n} q_i & \text{if } i = 0 \\ -q_i & \text{if } i \in V \setminus \{0\} \end{cases}
\]

and set \(c_{a1} = 0\) for all \(a \in A\) and we set \(f_a\) to be the cost of traversing arc \(a \in A\) and \(u_a = Q\). In 
the vehicle routing problem with time windows (VRPTW) where both capacity and time windows 
constraints are modeled we let \(K = \{1, 2\}\) since we both need a demand and a time commodity. For 
the time commodity we can add upper and lower bounds for each arc in order to model time windows.

Going back to the basic MCFCND model M1 we can obtain an alternative formulation by duplicating 
each variable, linking each pair of duplicate variables and applying Dantzig Wolfe decomposition 
with the linking constraints in the master problem. This technique is known as variable splitting 
(Fisher et al. [1997]) or Lagrangian decomposition (Guignard and Kim [1987]). In the resulting model 
each variable \(\gamma_p\) represents a “pattern” for a certain vertex \(i \in V\). We do not include an \(i\) superscript 
on \(\gamma_p\) (to keep notation simple), but defines a function \(v(p)\) that maps a pattern \(p\) to its corresponding 
vertex \(i \in V\). A pattern indicates which of the arcs adjacent to vertex \(i\) that are open and how much 
flow they carry. For an arc \(a \in \delta^+(v(p))\) and a commodity \(k \in K\) the coefficients \(x_{akp}^+ \geq 0\) denotes the 
flow of commodity \(k\) on arc \(a\) in pattern \(p\) and \(y_{akp}^+ \in \{0, 1\}\) denotes if arc \(a\) is open in pattern \(p\). For all 
archs \(a \in A \setminus \delta^+(v(p))\) and all \(k \in K\) we have \(x_{akp}^- = y_{akp}^- = 0\). The definition of \(x_{akp}^-\) and \(y_{akp}^-\) are similar 
but for arcs \(a \in \delta^-(v(p))\). Patterns should be consistent, i.e. an arc can only accommodate flow if it 
has been opened, the flow on each arc should respect the capacity limit and the flow out minus flow in 
of commodity \(k\) should equal \(d(i, k)\). Intuitively speaking, a pattern \(p\) indicates a feasible set of values 
of the variables corresponding to arcs adjacent to node \(v(p)\). The set of all patterns is denoted \(\Omega\) and 
the set of patterns for node \(i\) is denoted \(\Omega(i)\). With these definitions the model becomes

\[
(M2) \quad \min \sum_{k \in K} \sum_{a \in A} c_{ak} \left( \sum_{p \in \Omega} x_{akp}^+ \gamma_p \right) + \sum_{a \in A} f_a \left( \sum_{p \in \Omega} y_{akp}^+ \gamma_p \right)
\]
subject to

$$\sum_{p \in \Omega} x_{akp}^+ \gamma_p - \sum_{p \in \Omega} x_{akp}^- \gamma_p = 0 \quad \forall a \in A, k \in K$$ (7)

$$\sum_{p \in \Omega} y_{ap}^+ \gamma_p - \sum_{p \in \Omega} y_{ap}^- \gamma_p = 0 \quad \forall a \in A$$ (8)

$$\sum_{p \in \Omega(i)} \gamma_p = 1 \quad \forall i \in V$$ (9)

$$\sum_{p \in \Omega(i)} y_{ap}^+ \gamma_p \in \{0, 1\} \quad \forall a \in A$$ (10)

$$\gamma_p \geq 0 \quad \forall p \in \Omega$$ (11)

Here (7) and (8) are the linking constraints that ensures that the selection of patterns are compatible both with respect to $x$ and $y$ variables. Constraint (9) forces a pattern to be selected for each vertex and constraint (10) forces the original $y_a$ variables to be binary.

Model M2 is inspired by a similar model for the Fixed Charge Transportation Problem presented in Mingozzi and Roberti [2017]. A third model, M3, can be obtained by generalizing the *arc quantity* constraints proposed by Mingozzi and Roberti [2017] to the MCF-CND. Define $W^+(i, a, k, q) = \{p \in \Omega(i) : x_{akp}^+ = q\}$ to be the set of patterns for vertex $i$ for which arc $a$ is leaving vertex $i$ with $q$ commodities of type $k$. Similar we define $W^-(i, a, k, q) = \{p \in \Omega(i) : x_{akp}^- = q\}$. With these definitions the arc quantity constraints are

$$\sum_{p \in W^+(i, a, k, q)} \gamma_p - \sum_{p \in W^-(j, a, k, q)} \gamma_p = 0 \quad \forall a = (i, j) \in A, k \in K, q = 0, \ldots, u_a$$ (12)

We use M3 to denote model M2 plus constraint (12). Let LPM1, LPM2 and LPM3 denote the LP relaxation of model M1, M2 and M3, respectively. For a model $X$ we let $v(X)$ denote its objective value. We have that $v(LPM1) \leq v(LPM2) \leq v(LPM3)$. Model LPM2 and LPM3 can grow big and it can be advantageous to use column generation (and row generation) to solve these models.

Using models M2 and M3 we can derive new models and relaxations of the routing problems presented earlier (and many similar routing problems). The LP relaxations can in some cases be strengthened further by including known valid inequalities for the specific routing problems. Furthermore, alternative MCF-CND representations for a specific routing problem can lead to different models with better relaxations. All together this results in a framework for developing lower bounds (and exact methods) for many routing problems. Preliminary results indicate that the approach is most promising for TSP problems with side constraints.

**References**


A Benders Decomposition for the Inventory Routing Problem

Fabián Penaranda*, Marcus Poggi*, Rafael Martinelli**
*Departamento de Informática, **Departamento de Engenharia Industrial
Pontifícia Universidade Católica do Rio de Janeiro, Brazil
Email: {fpenaranda, poggi}@inf.puc-rio.br, martinelli@puc-rio.br

1 Introduction

Critical goods inventory is, most often, managed by the supplier under a service level agreement with its clients. This is the case for many chemicals industries, hospitals and other activities where the consequences of having a stock falling short is very expensive, if not unmeasurable. Under these conditions and when customers have with suppliers a joint managing of its stocks, the total cost of the logistic activity of maintaining a good available at all customers can be managed by a centralized entity. This entity, most likely the supplier, will then be responsible for determining when and how to deliver this good in such away that it fulfills the customer demands at all times while keeping the stocks within the desired ranges, at all customers and at its own warehouse. The inventory-routing problem (IRP) captures then the planning decisions when this entity also controls the routing of the delivery. The aim is therefore the minimization of total operating cost, inventory and routing costs. The IRP is sensitive to the stock replenishment policy. Two policies are tackled in the literature. The OU policy where deliveries must always fill the inventory to its maximum capacity; and the ML policy where any quantity can be delivered. We consider the ML policy in this work, for which, currently, the best results are obtained by [3], improving the results in [2].

In the IRP, a single supplier, denoted 0, produces a known quantity of a single commodity at each period $t$ of a finite planning horizon $H$. Using a homogeneous fleet of vehicles $K$, each with capacity $C_k$, the supplier serves a set $N$ of customers where each customer $i$ demands a known quantity $q_{it}$ in each period $t \in H$. Given $V = N \cup \{0\}$, each element $i \in V$ has an inventory capacity $C_i$, an initial inventory $I_i^0 \leq C_i$, and a unit holding cost $h_i$. The IRP consists of building feasible vehicle routes in each period such that no stockout occurs, while respecting inventory capacities and the load satisfies vehicle capacities. Furthermore, each customer can be serviced at most once per period.
2 A Benders Decomposition

We propose a Benders Decomposition for the IRP based on the MIP formulation of [2] on which we keep the inventory management in the first level, moving the vehicle routing decisions to the second level, solved for each period separately. The formulations on both levels of the decomposition are integer programs. With this structure we are not limited to use the resulting routing formulation from the decomposition, any known vehicle routing formulation can be used. Clearly, the tighter the formulation, the stronger are the bounds we expect to obtain. We choose to use the Two-Index Formulation [5]. Regular Benders cuts obtained from solving the linear relaxation of the second level may be separated during the execution of the algorithm. However to reach the integer optimal solution of the first level, we must separate Benders cuts from an integer solution of the second level. This is done by tailored Benders combinatorial cuts (or integrality cuts as in [4]).

**Benders Master Problem.** The first level of the decomposition contains the inventory management decisions. The routing contribution to the objective function is represented by variable $\alpha$ as in (1), and its value is coupled to variables $\alpha_t$ in (3) which in turn are associated to benders cut added for each period $t \in H$.

\[
\min \sum_{i \in V} \sum_{t \in H} h_i I_i^t + \alpha \quad \text{(inventory constraints)} \tag{1}
\]

\[
\alpha - \sum_{t \in H} \alpha_t \geq 0 \quad \text{(3)}
\]

**Benders Optimality Cuts.** Benders optimality cuts (4) are separated by solving the linear relaxation of the second level VRP associated to each master demand solution. Integer variables $w_s^t$ are associated to the right-hand side of the rounded capacity cuts in the sub-problem and constraints (5) state that their minimum value in the master problem corresponds to the number of vehicles needed to serve demands of all customers in $S \subset V$. Dual values $\beta^t$ and $\pi_i^t$ are associated to the degree constraints related to the depot and customers respectively, and dual values $\theta_s^t$ to the capacity constraints.

\[
\alpha_t \geq \sum_{i \in V} 2.\pi_i^t.y_i^t + \sum_{s \in S} 2.\theta_s^t.w_s^t + 2.K.\beta^t \quad \forall t \in H \tag{4}
\]

\[
C_k.w_s^t \geq \sum_{i \in s} Q_i^t \quad \forall t \in H, \forall S \subset V \tag{5}
\]

\[
w_s^t \in \mathbb{Z}^+ \quad \forall t \in H, \forall S \subset V \tag{6}
\]

**Benders Feasibility Cuts.** Benders feasibility cuts (7) are separated by solving the linear relaxation of a Bin Packing sub-problem, which is used only to check if a demand
obtained from the master problem would have a feasible routing solution, i.e. if it fits the vehicles (bin) capacities without splitting.

\[
\sum_{i \in V} \pi^t_i Q^t_i + \sum_{i \in V} \phi^t_i y^t_i + \sum_{k \in K} \beta^t_k \leq 0 \quad \forall t \in H
\]  

(7)

**Combinatorial Benders Optimality Cuts.** Combinatorial Benders optimality cuts (8)-(10) are used to close the integrality gap in the master problem and are separated by solving the integer VRP subproblem. We call them route based cuts, because the exact routing cost for a given demand \( r \) imposes a valid bound on the routing cost of any other demand \( r' \) if all feasible routes of \( r' \) are a subset of the feasible routes of \( r \). The set of infeasible sets \( IS_r \) contains minimal customer subsets whose demands do not fit a single vehicle. Each of these subsets, along with zero demand customers, represent the infeasible routes associated with demand \( r \). Binary variables \( \alpha^t_{IS_r} \) have an associated cost of \( Z^*_{VRP_r} \) that corresponds to the optimal routing cost of \( r \). Constraints (9) forces the value of these variables to 1 for each demand \( r' \) that meets the condition above, in which case, the \( Z^*_{VRP_r} \) routing bound is imposed to \( r' \) by constraints (8). Binary variables \( m^r_{s,t} \) are set to 1 by constraints (10) if each infeasible customer subset \( s \subset IS_r \) is still infeasible for demand \( r' \).

Terms \( V_1 \) and \( V_2 \) represent customers with positive and zero demand, respectively.

\[
\alpha^t \geq Z^*_{VRP(r)} \cdot \alpha^t_{IS_r} \quad \forall t \in H
\]

(8)

\[
\alpha^t_{IS_r} \geq 1 - |IS_r| - n + \sum_{s \in IS_r} m^r_{s,t} + \sum_{i \in V_1} y^t_i + \sum_{i \in V_2} (1 - y^t_i) \quad \forall t \in H
\]

(9)

\[
\sum_{i \in s} Q^t_i \leq K.C_k.m^r_{s,t} + C_k \quad \forall t \in H, \forall s \in IS
\]

(10)

#### 3 Computational Experiments

We tested the methodology on some of the classical instances from the IRP literature \[1\], considering multiple vehicles in the same manner as \[3\]. Our approach was developed in Julia and the experiments were conducted on an Intel Core i7 6700HQ, with 32GB of RAM. For each test, we allowed a time limit of 3600 seconds and we report the results obtained for three vehicles in Table 1. Column **Instance** presents the name of the instance, Column **Iter** presents the executed iterations, Columns **LB** and **UB** present the lower and upper bounds, respectively, Column **T(s)** the time in seconds and Columns **RBC** and **CBC** the number of regular and combinatorial Benders Cuts, respectively.

Experiments show optimal solution obtained only for instances up to 10 customers for 3 periods and up to 5 for 6 periods. The bottleneck of the current approach appears to be the resolution of the master problem. The integer extra variables to deal with the rounding of the Benders optimality cuts seem turn the MIP problem hard. Handling a
<table>
<thead>
<tr>
<th>Instance</th>
<th>Iter</th>
<th>LB</th>
<th>UB</th>
<th>RBC</th>
<th>CBC</th>
<th>T(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>absH3high_1n10</td>
<td>6</td>
<td>5506.09</td>
<td>5506.09</td>
<td>27</td>
<td>27</td>
<td>20.21</td>
</tr>
<tr>
<td>absH3high_2n10</td>
<td>15</td>
<td>5743.25</td>
<td>5743.25</td>
<td>66</td>
<td>66</td>
<td>62.87</td>
</tr>
<tr>
<td>absH3high_3n10</td>
<td>29</td>
<td>4808.06</td>
<td>4808.06</td>
<td>123</td>
<td>123</td>
<td>103.27</td>
</tr>
<tr>
<td>absH3high_4n10</td>
<td>57</td>
<td>5335.29</td>
<td>5335.29</td>
<td>72</td>
<td>195</td>
<td>1373.1</td>
</tr>
<tr>
<td>absH3high_5n10</td>
<td>19</td>
<td>5224.45</td>
<td>5224.45</td>
<td>93</td>
<td>93</td>
<td>43.91</td>
</tr>
<tr>
<td>absH3low_1n10</td>
<td>7</td>
<td>2732.61</td>
<td>2732.61</td>
<td>33</td>
<td>33</td>
<td>19.37</td>
</tr>
<tr>
<td>absH3low_2n10</td>
<td>24</td>
<td>3470.15</td>
<td>3470.15</td>
<td>114</td>
<td>114</td>
<td>58.79</td>
</tr>
<tr>
<td>absH3low_3n10</td>
<td>37</td>
<td>2649.26</td>
<td>2649.26</td>
<td>171</td>
<td>171</td>
<td>94.68</td>
</tr>
<tr>
<td>absH3low_4n10</td>
<td>32</td>
<td>3183.36</td>
<td>3183.36</td>
<td>93</td>
<td>135</td>
<td>174.95</td>
</tr>
<tr>
<td>absH3low_5n10</td>
<td>28</td>
<td>2461.12</td>
<td>2461.11</td>
<td>129</td>
<td>129</td>
<td>75.27</td>
</tr>
<tr>
<td>absH6high_1n5</td>
<td>29</td>
<td>7259.05</td>
<td>7259.05</td>
<td>294</td>
<td>294</td>
<td>479.79</td>
</tr>
<tr>
<td>absH6high_2n5</td>
<td>26</td>
<td>6530.35</td>
<td>6530.35</td>
<td>306</td>
<td>306</td>
<td>841.55</td>
</tr>
<tr>
<td>absH6high_3n5</td>
<td>5</td>
<td>9862.89</td>
<td>9862.89</td>
<td>72</td>
<td>72</td>
<td>56.89</td>
</tr>
<tr>
<td>absH6high_4n5</td>
<td>10</td>
<td>6405.74</td>
<td>6405.74</td>
<td>150</td>
<td>150</td>
<td>429.64</td>
</tr>
<tr>
<td>absH6high_5n5</td>
<td>20</td>
<td>5972.78</td>
<td>5972.78</td>
<td>234</td>
<td>234</td>
<td>808.44</td>
</tr>
<tr>
<td>absH6low_1n5</td>
<td>29</td>
<td>4657.15</td>
<td>4657.15</td>
<td>252</td>
<td>252</td>
<td>453.99</td>
</tr>
<tr>
<td>absH6low_2n5</td>
<td>29</td>
<td>4221.00</td>
<td>4221.00</td>
<td>336</td>
<td>336</td>
<td>1252.38</td>
</tr>
<tr>
<td>absH6low_3n5</td>
<td>6</td>
<td>7703.53</td>
<td>7703.53</td>
<td>78</td>
<td>78</td>
<td>114.51</td>
</tr>
<tr>
<td>absH6low_4n5</td>
<td>10</td>
<td>4495.31</td>
<td>4495.31</td>
<td>150</td>
<td>150</td>
<td>398.34</td>
</tr>
<tr>
<td>absH6low_5n5</td>
<td>8</td>
<td>3879.09</td>
<td>3879.09</td>
<td>102</td>
<td>102</td>
<td>397.61</td>
</tr>
</tbody>
</table>

Table 1: Results for selected IRP instances. K=3

Large number of combinatorial cuts represents also a challenge for the MIP solvers tested. Current research is now focused on solving the master efficiently.

References


Cycles, Pricing, and Pivots

* Jacques Desrosiers
Departement of Decision Sciences
GERAD & HEC Montréal, Canada
Email: jacques.desrosiers@hec.ca

Jean Bertrand Gauthier
Chair of Logistics Management
Gutenberg School of Management and Economics, Mainz, Germany

Within the realm of linear programming, iterative algorithms that maintain feasibility throughout the solution process all identify a direction and then move along the latter with some non-negative step-size. We call the oracle used to identify a direction the pricing problem. Since this oracle maintains its form across the various algorithms, it is a common denominator whose canonical form is first observed in the *minimum mean cycle-canceling algorithm* (Goldberg and Tarjan 1989), the average cost of a cycle being taken over the number of arcs. In this respect, the network flow nomenclature is heavily borrowed thus contributing to the intuitive understanding of the pricing problem.

It is well known that all directed cycles necessary to reach an optimal minimum cost flow solution can be observed on the residual network. Furthermore, each of these can individually accommodate some strictly positive flow. In optimization terms, each of these directed cycles, or combination of, forms a direction. A degenerate pivot is therefore induced when the selected cycle does not actually exist on the residual network. The concepts of paths and cycles along with some network flow properties can be transferred to linear programs and alternative necessary and sufficient optimality conditions expressed on the so-called residual problem are obtained in the process. We propose a family of algorithms with non-degenerate pivots and also show that the local search heuristics for vehicle routing problems, such as 2-opt, 3-opt, swap, relocate, . . . are indeed directed cycles on the residual network.

References


Collaborative Vehicle Routing: Challenges and Limitations of Auction-based Mechanisms

Margaretha Gansterer, Richard F. Hartl¹, and Rudolf Vetschera
Department of Business Administration,
University of Vienna, Vienna, Austria Richard.Hartl@univie.ac.at

1 Introduction
Carrier collaboration is one of the big trends in transportation, as identified by Speranza (2018). Since the percentage of empty trucks contributing to traffic, pollution, and accidents is between 15% and 30%, exchanging transportation requests between carriers can reduce inefficiencies and generate economic and ecologic benefits for the partners as well as for the society. In horizontal logistics collaboration, several companies exchange some of their transportation requests in order to execute them more efficiently; see e.g. Cruijssen et al. (2007). The results of these initiatives are impressive: efficiency improvements of up to 30% have been reported, as recently surveyed by Gansterer and Hartl (2017).

Horizontal cooperation of carriers, or more precisely, the forming of coalitions is commonly realized by using auction-based exchange mechanisms. In combinatorial auctions, requests are not traded individually but are combined into packages, called bundles. The main reason for this is that a bundle might have a different (typically higher) value to the partners than the sum of the individual requests. For instance a single rather remote request might not be attractive for a carrier, while two such requests in the same area might be. Berger and Bierwirth (2010) identify 5 typical phases in combinatorial transportation auctions: First, carriers select request that they want to auction-off (request selection), then the auctioneer generates bundles of requests and offers them to the participants (bundling). These give their bids on offered bundles by calculating the marginal profit of each bundle (bidding). Finally, requests are allocated to carriers according to their bids, so that the total profit for the coalition is maximized (winner determination). Gained profits are distributed among the carriers (profit sharing).

Each of these stages has been intensively researched in the literature; see e.g. the survey by Gansterer and Hartl (2016). The idea is to achieve an efficient relocation of requests without revealing (too much) information. More or less all studies so far have assumed that the partners act truthfully and do not show strategic behavior. For instance, they place bids on requests equal to their real marginal costs. Game theoretical aspects have been widely ignored so far. This contribution tries to fill this gap by presenting a first step towards analyzing the impact of strategic behavior and finding mechanisms to avoid strategic behavior.

2 Proposed mechanisms
The exchange mechanism should have some desired properties: incentive compatibility (IC,
reporting true costs should be a Nash equilibrium), efficiency (EF, maximum value creation from exchange), budget balance (BB, no loss for auctioneer), and individual rationality (IR, no bidder is worse off when participating). These can be fulfilled for simple auctions, where carriers either buy or sell transportation requests; see e.g. Xu, Huang, and Cheng (2016). However, we show that these desired properties cannot be fulfilled at the same time in double sided auctions, where carriers exchange packages of requests, having synergies among them. In particular, IC and EF means that BB is impossible, i.e. a subsidy or a participation fee is needed.

Focusing on carriers’ strategic behavior, it is known that an IC pricing mechanism (that induces reporting true costs) can be designed based on the well-known Vickrey-Clarke-Groves (VCG) mechanism, i.e. using the concept of second price sealed bid auctions. However, we show that the interpretation of a second price is not straightforward in case of double-sided auctions. In fact, computing “true costs” is problematic here. Note that the marginal costs of inserting new requests are not independent of the existing ones, i.e. of the requests that can be handed over to partners or that must be kept after redistribution.

Carriers have to bid on bundles that potentially include their own requests, and therefore have to give buying and selling prices, simultaneously. Determining second prices by dropping a carrier (as it is supposed to be done in VCG) would also forbid all other exchanges where this carrier is involved. This suggests using a new approach and combining multiple bidders into a ”team bidder”. We follow the logic of the VCG mechanisms, which requires that bidders should not be able to influence the price they are paying via their bids. The proposed mechanism is therefore IC and EF.

3 Computational study

We numerically analyze the proposed mechanisms by applying them to available test instances. We consider carriers having less than truckload requests. The instances (O1, O2, O3) cover different scenarios in terms of (i) degree of customer overlaps and (ii) distance of requests to the carriers’ depots. Since both mechanisms can be proven to be IC, and EF, we focus on the properties of BB and IR. While VCG is IR, this property does not hold for the proposed team bidder mechanism. Thus, we first quantify how often IR is violated. None of the mechanisms is BB, and VCG will per definition lead to a higher loss for the auctioneer. This loss can be compensated by a participation fee paid by the carriers. We assume that the auctioneer’s loss is distributed equally to the carriers. Obviously, such a fee might lead to payments that violate individual rationality. We therefore evaluate numerically, whether IR holds, if carriers are charged with a participation fee. Results are summarized in the table below. While VCG is 100% IR if carriers do not have to pay a participation fee, this IR is often violated if the auctioneer’s loss has to be compensated (IR ex post in the table below).

In the table, AL denotes the auctioneer’s loss. When distributing the loss equally among participants, no individual IR can be reached in 52%, 42%, and 35% of the instances, respectively. However, the participation fee has a smaller impact on IR in the team bidder mechanism. Note that a violation of IR means that some partner is worse off, while on average all partners are better off.
If an occasional loss is compensated by improvements in other periods, IR violations might be less problematic.

<table>
<thead>
<tr>
<th>Instance</th>
<th>VCG</th>
<th></th>
<th>Team Bidder</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>IR</td>
<td>AL</td>
<td>IR ex post</td>
<td>IR</td>
</tr>
<tr>
<td>O1</td>
<td>0%</td>
<td>72.96</td>
<td>52%</td>
<td>5%</td>
</tr>
<tr>
<td>O2</td>
<td>0%</td>
<td>303.76</td>
<td>42%</td>
<td>35%</td>
</tr>
<tr>
<td>O3</td>
<td>0%</td>
<td>740.48</td>
<td>35%</td>
<td>35%</td>
</tr>
</tbody>
</table>

Our computational results show that both of the proposed mechanisms come with some limitations regarding of the four desired properties. VCG is EF, and IC. But it is not BB, and not IR as soon as carriers have to compensate auctioneers’ losses. The team bidder mechanism is less costly for the auctioneer, but IR cannot be guaranteed. Even if carriers do not pay for the loss of the auctioneer, there might be participants who by participating are worse-off.

### 4 Conclusions

Our study shows that strategic behavior in auction-based carrier collaborations can be avoided, but might violate IR of participants. It cannot be guaranteed that all carriers benefit from participation. Vice versa, if carriers insist on proven IR, untruthful behavior might become the dominant strategy. However, the identification of a profitable untruthful bidding strategy is not straightforward. Due to the synergies among transportation requests, and the fact that carriers sell and buy requests simultaneously, the outcome of strategic bidding is hard to predict. We emphasize this by additional computational results, where we observe that a natural approach of untruthful bidding is not the dominant strategy.

### References


Nested Branch-and-Price-and-Cut for Vehicle Routing Problems with Multiple Resource Interdependencies

Christian Tilk (a), Michael Drexl (b), Stefan Irnich (a)

(a) Chair of Logistics Management, Gutenberg School of Management and Economics, Johannes Gutenberg University Mainz, Jakob-Welder-Weg 9, D-55128 Mainz, Germany.

(b) Faculty of Applied Natural Sciences and Industrial Engineering, Deggendorf Institute of Technology, D-94469 Deggendorf, Germany.

Abstract:

This presentation considers vehicle routing problems (VRPs) with multiple resource interdependencies and addresses the development and computational evaluation of an exact branch-and-price-and-cut algorithm for their exact solution. An interdependency between two resources means that the two resource consumptions influence one another in such a way that a tradeoff exists between them. This impacts the feasibility and/or the cost of a solution.

The subproblem in branch-and-price-and-cut procedures for VRPs is very often a variant of the shortest-path problem with resource constraints (SPPRC). For the exact solution of many SPPRC variants, dynamic-programming based labeling algorithms are predominant. The tradeoffs in problems with multiple resource interdependencies, however, render the application of labeling algorithms unpromising. This is because complex data structures for managing the tradeoff curves are necessary and only weak dominance criteria are possible, so that the labeling algorithm becomes almost a pure enumeration. Therefore, we propose to solve also the SPPRC subproblem with branch-and-price-and-cut. This results in a three-level, nested branch-and-price-and-cut algorithm. We analyze different variants of the algorithm to enable the exchange of columns and also rows between the different levels.

To demonstrate the computational viability of our approach, we perform computational experiments on a prototypical VRP with time windows, minimal and maximal delivery quantities for each customer, a customer-dependent profit paid for each demand unit delivered, and temporal synchronization constraints between some pairs of customers. In this problem, tradeoffs exist between cost and load and between cost and time.