Università degli Studi di Cagliari

PHD DEGREE

in Mathematics and Computer Science

Cycle XXXII

TITLE OF THE PHD THESIS

New Routing Problems with possibly correlated travel times

Scientific Disciplinary Sector(s)

MAT/09 Operational Research

Phd Student :	Federica BOMBOI
Coordinator of the PhD Programme :	Giuseppe RODRIGUEZ Michele MARCHESI
Supervisor :	Paola ZUDDAS Massimo DI FRANCESCO Christoph BUCHHEIM

Final exam. Academic Year 2018 – 2019

Thesis defence: January-February 2020 Session

To my family.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

Federica Bomboi
Final examination academic year 2018-2019
Exam session February 2020

Acknowledgements

I would like to acknowledge my tutors Prof. Dr. Massimo Di Francesco of University of Cagliari and Prof. Dr. Christoph Buchheim of TU Dortmund. I also acknowledge Jonas Prünte for his help and support.

I gratefully acknowledge Sardinia Regional Government for the financial support of her PhD scholarship (P.O.R. Sardegna F.S.E. Operational Programme of the Autonomous Region of Sardinia, European Social Fund 2007-2013 - Axis IV Human Resources, Objective l.3, Line of Activity l.3.1.) and the German Academic Exchange Service (DAAD) for the financial support of her scholarship (Research Grants - Short-Term Grants, 2018 - 57381332).

Unione europea
Fondo sociale europeo

REPVBBLICA ITALIANA

REGIONE AUTÒNOMA DE SARDIGNA
REGIONE AUTONOMA DELLA SARDEGNA

FSE 2007-2013
POR SARDEGNA
obiettivo competitività regionale e occupazione

Università degli Studi di Cagliari

Abstract

In the literature of operational research, Vehicle Routing Problems (VRP) were and still are subject of countless studies.

Under the scope of combinatorial optimization, this thesis analyses some variants of VRP both with deterministic and uncertain travel times.

The deterministic problem under study is a drayage problem with characteristics concerning service types and requirement seldom investigated all together. The formulations proposed to model this problem are: the node-arc formulation and the Set Partitioning formulation. Concerning the solution methods, two heuristics and a branch-and-price approach are presented.

The section dealing with uncertain and correlated travel times faces two classes of VRP with time windows using either single or joint chance constraints depending on whether missing a customers time window makes the entire route infeasible or not.

From a comparison between deterministic and stochastic methods, these last represent a profitable investment to guarantee the feasibility of the solution in realistic instances.

Table of contents

List of figures

List of tables

Nomenclature

The next list describes several symbols that will be later used within the body of the thesis.

Abbreviations

ESPPRC: Elementary SPPRC
(The shortest path does not pass through the same node twice.)

jcc: Joint chance constraint

PCTSP: Prize Collecting TSP
(TSP where the cost of the path minus the sum of the prizes associated with the nodes in the path is minimized)

scc: Single chance constraint

SPPRC: Shortest Path Problem with resource constraints
(Finding the shortest path from a source to a sink node satisfying a set of constraints defined over a set of resources)

SVRP: Stochastic VRP
(VRP with uncertain travel costs)

SVRPTW-BL: SVRPTW with Backhauls and Linehauls
(Customer demands can be either deliveries - linehauls - or pickups - backhauls)

SVRPTW: SVRP with Time Windows

TSP: Travelling Salesman Problem
(Minimizing the total cost of the path starting and returning to the same node visiting each one of the other nodes)

VRP: Vehicle Routing Problem
(All customers correspond to deliveries and the demands are deterministic and known)

VRPB: VRP with Backhauls
(Precedence constraints on the deliveries - linehaul customers - are assumed, whenever a route serves both types of customer)

VRPCB: VRP with clustered Backhauls
(VRPB where the cluster of delivery customers has to be served before the first pickup customer can be visited)

VRPPD: VRP with Pickup and Delivery
(A delivery and a pickup are assumed for each customer)

VRPTW: VRP with Time Windows

Data and Parameters

α_{ir}: Binary coefficient of the deterministic problem with value 1 if customer i is served in route r and 0 otherwise

ε: Threshold for feasibility of a route

μ_e: Expected travel time of arc e

σ_e: Standard deviation of travel time of arc e

$\sigma_{e,f}$: Covariance between travel times of arc e and arc f

a_e: Earliest arrival time at end node of arc e

b_e: Latest arrival time at end node of arc e

d_{ij}: Cost of arc (i, j) in the deterministic problem

d_r: Cost of route $r \in R$ in the deterministic problem

S_{max}: Maximum number of stops in a route for the deterministic problem

Sets

C: Set of customer nodes (when it it includes depot, it is specified)

E: Set of all arcs

E_r: Set of arcs contained in a route r

K: Set of vehicles

R: Set of all feasible routes

Variables

X_e: Random variable describing the travel time of arc e

x_r: Binary variable of the deterministic problem with value 1 if route r is selected and 0 otherwise

x_{ijk}: Binary variable of the deterministic problem with value 1 if arc (i, j) is traversed by vehicle k and 0 otherwise

Chapter 1

Introduction

The scope of the thesis is the 0-1 integer programming (and combinatorial) optimization applied to the routing problems. Integer and combinatorial optimization deals with problems of maximizing or minimizing a function of many variables subject to (a) inequality and equality constraints and (b) integrality restrictions on some or all of the variables [Nemhauser et al. 1988].

The 0-1 integer programming (0-1 IP) problem can be expressed as:

$$\max\{cx : Ax \leq b, x \in B_+^n\}$$

where B_+^n is the set of 0-1 n-dimensional vectors and x is the variable used to represent logical relationships. The 0-1 IP is a particular case of the most general linear mixed-integer programming model (MIP):

$$\max\{cx + hy : Ax + Gy \leq b, x \in \mathbb{Z}_+^n, y \in \mathbb{R}_+^p\}$$

In the notation above [Nemhauser et al. 1988], x and y are the variables, $\mathbb{Z}_+^n$ is the set of nonnegative integral n-dimensional vectors and $\mathbb{R}_+^p$ is the set of nonnegative real p-dimensional vector.

The 0-1 IP has many applications (e.g. 0-1 Knapsack problem, Assignment and Matching problems). Vehicle Routing Problems (VRP), in which the binary choice of traveling or not traveling a route has to be decided, can also be seen as an application of the 0-1 IP. In the literature of operational research, Vehicle Routing Problems were and still are subject of countless studies. These problems consist in finding a set of vehicle routes serving certain customer requests with the minimum total travel cost.

The more frequent models for the basic version of VRP are the *vehicle flow formulations* [Toth and Vigo 2002]. These formulations are also known as node-arc formulations because each arc (or node) in the graph is associated with an integer variable which count the number of times the arc is traversed by a vehicle. Following the notation of [Toth and Vigo 2002], let $G = (V, A)$ be a complete graph where $V = \{0, \ldots, n\}$ is the vertex set, with 0 the depot and $C = V \setminus \{0\}$ the set of customer and A the arc set, a first basic model of VRP is:

$$\min \quad \sum_{i \in V} \sum_{j \in V} d_{ij} x_{ij} \tag{1.1}$$

$$\sum_{i \in V} x_{ij} = 1 \quad \forall j \in C \tag{1.2}$$

$$\sum_{j \in V} x_{ij} = 1 \quad \forall i \in C \tag{1.3}$$

$$\sum_{i \in V} x_{i0} = K \tag{1.4}$$

$$\sum_{j \in V} x_{0j} = K \tag{1.5}$$

$$\sum_{i \notin S} \sum_{j \in S} x_{ji} \geq r(S) \quad \forall S \subseteq C, S \neq \emptyset \tag{1.6}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i, j \in V \tag{1.7}$$

The objective is to minimize the total travel cost (1.1), where d_{ij} is the cost of arc (i, j). According to constraints (1.2) and (1.3), exactly one arc enters and leaves each customer. Constraints (1.4) and (1.5) impose that K vehicles leave and enter the depot. Setting $r(S)$ as the minimum number of vehicles needed to serve set S, the connectivity of the solution and the vehicle capacity requirements is imposed by constraints (1.6). The variable x_{ij} takes value 1 if arc (i, j) is traversed by a vehicle in the optimal solution and 0 otherwise.

The model above is simple but according to the constraints of a problem, a disadvantage of the node-arc formulations can be the modeling phase itself, that is finding a node-arc formulation capturing all attributes of the specific problem at hand. Another important disadvantage of the node-arc formulations lies in the subtour elimination constraints (1.6), that are in general exponentially many and are solved either by separation or introducing additional variables than the ones used in (1.1) - (1.7). More variables we use, slower the problem is solved.

To remedy this disadvantage, the Set Partitioning formulation uses one binary variable for each feasible route in the problem. Assuming that the set of customers $C = V \setminus \{0\}$ is known, together with the set of feasible routes R and the corresponding cost d_r for each $r \in R$, the following Set Partitioning formulation models the problem above:

$$\min \quad \sum_{r\in R} d_r x_r \tag{1.8}$$

$$\sum_{r\in R} x_r \leq K \tag{1.9}$$

$$\sum_{r\in R} \alpha_{ir} x_r = 1 \quad \forall i \in C \tag{1.10}$$

$$x_r \in \{0,1\} \qquad \forall r \in R \tag{1.11}$$

Here we choose a cheapest subset of all routes (1.8) such that each costumer is visited exactly once by these routes (1.10) and at maximum K trucks/routes are used (1.9). The variable x_r takes value 1 if the route r is in the optimal solution and 0 otherwise (1.11).

A disadvantage of the Set Partitioning formulation is the enumeration of all the feasible routes needed, which cause a big use of memory and a long running time.

1.1 Deterministic section

In the thesis, in Chapter 2, a case of pick up and delivery problem called drayage is studied. Drayage is a popular research area in the field of intermodal transportation logistics. It concerns the distribution of empty and loaded containers between intermodal facilities (e.g. ports or railway terminals), export and import customers. Intermodal transportation is subdivided into pre-haulage, main-haulage and end-haulage. Pre-haulage denotes the route segments from customer to terminal, main-haulage indicates the route segments from terminal to terminal and end-haulage from terminal to customer [Nossack et al. 2013]. A picture of intermodal container transportation is shown in Fig. 1.1 [Nossack et al. 2013]. "In the literature, pre-haulage and end-haulage is also referred to as inland container transport, hinterland service or drayage" [Reinhardt et al. 2016]. It is generally handled by trucks and it accounts for a significant portion of the transportation cost, even though main-haulage represents the longest traveling distance carried out by rail or maritime shipment. Then, optimizing drayage problems contributes not only to reducing traveling costs but affects directly the environment reducing CO_2 emission, pollution and traffic congestion. The drayage problem presented in the thesis differs from the majority of the literature review presented in Section 2.3 for the problem characteristics and for the methodology used, as detailed in Chapter 2 below. Given the modeling complexity of the constraints, the problem is going to be analyzed and solved by the Set Partitioning formulation, although it has not been a very common formulation in the drayage area. Anyway, in order to provide more than

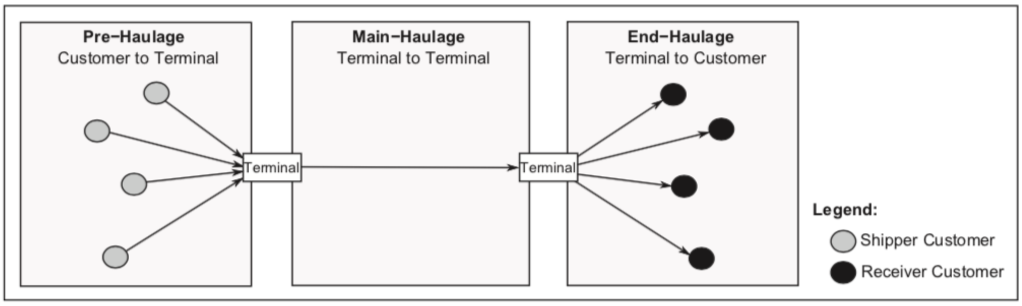


Fig. 1.1 Intermodal container transportation, [Nossack et al. 2013]

one formulation and in particular a formulation more in line with the trends in the drayage, a node-arc model is also presented. The continuous relaxation of such a formulation is usually rather weak, making it an inefficient choice to develop solution methods. Yet, the continuous relaxation of the corresponding Set Covering-type formulations is known to usually provide much stronger lower bounds. In order to reach the required integer solution, this approach has to be embedded in a branch-and-bound.

This idea leads to the branch-and-price approach that is a common approach in the field of VRP and uses column generation for solving the relaxed Set Partitioning problem. The column generation is a method for solving linear programming problems adding columns/ constraints during the pricing step of the dual simplex formulation of a linear problem. In fact, it consists in solving a restricted linear master problem in which just a subset of routes is involved identifying iteratively routes to add. The column generation is very useful for problems of exponential size, that is, with an exponential number of columns because it deals at every iteration with a problem of smaller size. In order to reach the required integer solution, the column generation approach has to be embedded in a branch-and-bound that does not branch according to the Set Partitioning formulation.

Taking again the Set Partitioning formulation into account (1.8) - (1.11), the relaxation of the formulation is obtained substituting (1.11) with (1.15), obtaining the following relaxation problem:

$$\min \quad \sum_{r\in R} d_r x_r \tag{1.12}$$

$$\sum_{r\in R} x_r \leq K \tag{1.13}$$

$$\sum_{r\in R} \alpha_{ir} x_r = 1 \quad \forall i \in C \tag{1.14}$$

$$x_r \geq 0 \qquad \forall r \in R \tag{1.15}$$

The column generation approach starts solving the primal problem (1.12) - (1.15) using a subset of feasible routes $R^{'} \subseteq R$ in the place of R. An other optimization problem provides the routes to add in the primal problem in order to reach the feasible solution. This optimization problem is called pricing. Looking at the dual formulation of the problem given by (1.12) - (1.15),

$$\bar{d}_r = d_r - \bar{\theta} - \sum_{i \in C} \bar{\lambda}_i \alpha_{ir} \tag{1.16}$$

is the reduced cost of route r, where the variable θ is associated to the constraint (1.13), the variables λ_i to the constraints (1.14). According to the column generation approach, a route is added to the subset of routes used in (1.12) - (1.15) when its reduced cost is negative, that is when $(\bar{\lambda}, \bar{\theta})$ is not feasible in the dual formulation of (1.12) - (1.15). More then one route can be added at a time. The pricing problem consists then in an optimization problem in which the objective is to find the minimum reduced cost:

$$\min_{r \in R} \quad \bar{d}_r \tag{1.17}$$

Algorithm 1 shows the scheme of the general column generation method. Before using

Algorithm 1 Column generation algorithm

Compute set of feasible routes $R^{'} \subseteq R$ that provides at least one feasible solution for (1.12) - (1.15) with $R^{'}$
while Routes with negative reduced costs can be found **do**
    Solve (1.12) - (1.15) with $R^{'}$
    Solve a heuristic Pricing Problem for finding new routes with negative reduced costs
    if No Routes with negative reduced costs can be found **then**
        Solve the Pricing Problem for finding new routes with negative reduced costs
    end if
    Add the new routes to $R^{'}$
end while
Return the solution of (1.12) - (1.15) with $R^{'}$

the exact pricing, a faster heuristic algorithm can be used to find new routes. Only if no routes with negative reduced costs can be found, the exact pricing is necessary to prove that the algorithm can stop.

For solving the drayage problem of Chapter 2, in Section 2.6 also the column generation approach and a branch-and-bound are going to be used and the solution obtained is going to be compared to the other presented solution methods.

1.2 Stochastic section

The term VRP in general refers to the deterministic problem, in which all the data are known and not subject to uncertainty. However, in practice, due to a variety of influences such as traffic jams, weather, customer availability etc., the real situation is often unpredictable, so that a solution of the deterministic problem in most cases produces solutions that are far from optimal in reality. To deal with this problem, specific approaches for addressing these uncertainties are needed. VRPs involving uncertainty and for which the probability distribution governing the uncertain data can be estimated (or it is known), are referred to as Stochastic Vehicle Routing Problems (SVRP). This class of routing problems subject to the uncertainty of the travel times is addressed in Chapter 3 of the thesis. Travel times are then random parameters.

In general terms, with respect to a *fixed* route r, we are going to discuss if the route r is feasible with a high enough probability and if so, what is the expected cost of route r. This is motivated by the fact that many exact as well as heuristic approaches for solving vehicle routing problems can be reduced to the above tasks, in particular when deciding if the route r is feasible with a high enough probability is a difficult problem in itself. An important example for this class of approaches is again the Set Partitioning approach to the VRP.

Citing [Ackooij et al. 2011] "the main difficulty of such models (with random parameters in optimization problems) is due to (optimal) decisions that have to be taken prior to the observation of random parameters. There are circumstances in which compensating decisions for balancing constraint violation do not exist or cannot be modeled by cost in any reasonable way. In such circumstances one would rather insist on decisions guaranteeing feasibility as much as possible".

In the thesis the approach of the chance constraints is used for assuring a high probability of feasibility in case of unexpected extreme events. Chance constrained programming was first introduced by [Charnes et al. 1959] in 1959 and the derived method of the chance constraint is one of the major approaches used in stochastic optimization. It is used to ensure that the probability of meeting a constraint of the uncertain problem is above a fixed threshold. Let the following be a general formulation of an optimization problem with decision variable x, objective function f, equality constraints g_i and inequality constraints h_j and subject to uncertain data ξ:

$$\begin{array}{lll} \min & f(x,\xi) & \\ \text{s.t.} & g_i(x,\xi) = a & i \in \{1,\dots,n\} \\ & h_j(x,\xi) \geq b & j \in \{1,\dots,m\} \end{array}$$

We can guarantee that each inequality constraint is satisfied for a fixed probability level p, formulating the single chance constraints as:

$$P(h_j(x,\xi) \geq b) \geq p \quad \forall j$$

where, for each $j \in \{1,\ldots,m\}$, a chance constraint is set. In this form, different probability levels p_j can also be assigned to every chance constraint. However, if we want that the constraint as a whole is satisfied to the fixed probability level p, then one joint chance constraint has to be formulated as:

$$P(h_j(x,\xi) \geq b \quad \forall j) \geq p$$

Single chance constraints are in general easier to solve then the joint chance constraints. The first can be in fact expressed as deterministic inequalities. The major challenge towards solving chance constrained optimization problems lies in the computation of the probability; often normal (Gaussian) distribution is considered as an adequate assumption for many uncertain variables in the engineering practice and only mean and variance are required, which are usually available [Li et al. 2008].

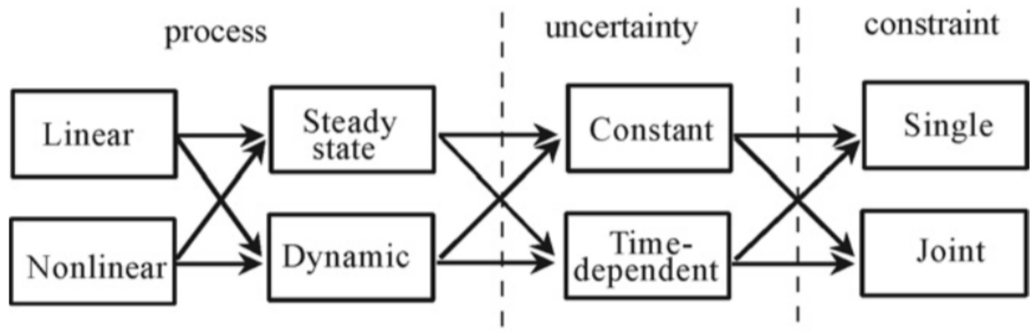


Fig. 1.2 Classification of chance constrained problems [Li et al. 2008]

Following the classification of chance constrained problems given by [Li et al. 2008], Fig. 1.2, in this thesis, in order to lead to a high probability of complying with time windows constraints for solving two variants of the VRPTW (VRP with Time Windows) subject to stochastic and normally distributed travel times, in a linear process both single and joint chance constraint approaches are used with both cases of constant and time dependent uncertainty.

1.3 Objectives

As a result of the consideration above, the objectives of the thesis are now presented.

In the drayage problem in Chapter 2, several characteristics regarding the service requests are considered. The problem has some attributes that are motivated by real cases of carrier shipment and all together are not investigated yet in the literature. The objective is to compare several solution methods to solve the problem in a reasonable time.

Concerning the SVRPTWs in Chapter 3, first the objectives are to investigate the importance of considering correlations between travel times by solving real instances and of considering travel times varying over the day by solving instances created with real data. A main objective is to give for the first time an approach for solving the single chance constrained routing problem with correlations, without and with time dependencies. Following this, we elaborate an algorithm considering correlations and time dependencies at the same time for solving single chance constrained routing problems. Another main objective is to give for the first time for a specific class of VRP like drayage an approach for solving the joint chance constrained routing problem with and without correlations. Finally, we consider an estimation of the waiting times of the vehicles at the customer locations, that can be used in penalty based approaches.

As a consequence of the study of Chapters 2 and 3, a comparison between the stochastic and the deterministic approaches is presented in order to check the costs of the investment in stochastic methodologies and to analyze the quality of the optimal solutions in terms of feasibility of the routes.

1.4 Outline

Summarizing, in Chapter 2 a new deterministic drayage problem is studied, in Chapter 3 two variants of the VRPTW subject to uncertain travel times are studied, in Chapter 4 a comparison between the deterministic and stochastic methods is given. Furthermore, two appendices follow the Bibliography for more details about the number of feasible routes of the drayage problem (Appendix A) and the computation of the updated moments for the algorithms presented in Chapter 3 (Appendix B).

Chapter 2

A deterministic drayage problem

2.1 Introduction

The objective of this chapter is solving a drayage problem with attributes concerning commodity types and service requirements which make the problem hard to solve with conventional methods. These characteristics are presented in details in the following Section 2.2. Next, the most related literature on the problem is presented and the main differences are pointed out in Section 2.3. In Section 2.4, a mathematical model for the problem statement is proposed and a node-arc formulation and a Set Partitioning formulation are presented. From an analysis at the end of Section 2.4 on how the types of request influence the number of feasible routes and therefore the running time, two heuristics are proposed in Section 2.5. These heuristics have scalable parameters to find either faster or more precise solutions, and both heuristics can be combined with each other. In Section 2.7, we perform an extensive experimental evaluation and compare the two heuristics with the "exact" solution obtained with the Set Partitioning formulation to which all the feasible routes are given. The research presented in this chapter is also collected in [Bomboi et al. 2018].

2.2 Problem description

In this section, a drayage problem motivated by the case study of a maritime medium-sized carrier is presented. In this context, the intermodal facility is a port in which a fleet of trucks and containers are based and adopted by the carrier to provide transportation services to the customer locations nearby the port. The routes of the operating trucks start and end at the port. The fleet of trucks and transported containers are homogeneous. Each truck can carry up to two 20 feet containers, in line with the standards in many countries, and serves a single container at a time. One attribute considered in this study is motivated by the reality of some special requests. It occurs that some container loads are not allowed to be carried with other loaded containers at the same time, e.g. containers containing flammable commodities like tobacco and alcohol or commodities with too large weight. Therefore, it is important for the carrier to meet these customer requests. For the sake of clarity, in this thesis, containers which are carrying these special loads are referred to as "special" and the other containers are called "ordinary". A second attribute of this study is that the customer is allowed to choose between two ways of service, "stay-with" and "drop & pick". In many papers of the related literature, containers are left at customer locations by trucks bypassing packing and unpacking works. Since drivers do not wait for containers during loading and unloading operations, they can serve additional customers in the meantime. This type of service is known as drop & pick. In the stay-with service, drivers wait for containers during packing and unpacking operations and trucks carry the same container before and after the customer service. In Fig. 2.1 on the left, an illustration of two stay with services to an import and to an export customer with one container per truck: the first service is made by unloading the container and the second by loading the container. On the right of Fig. 2.1, it is shown the same path with the same customers asking a drop & pick service: the container is dropped at the first import customer and a new loaded container is picked up at the export customer. From the presented characteristics, this study can be seen as a generalization of the drayage

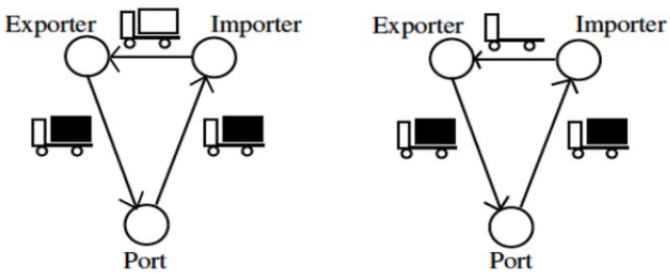


Fig. 2.1 On the left stay-with services, on the right drop & pick services

problem stated in [Ghezelsoflu et al. 2018], in which only the stay-with service is adopted and no special requests are considered. The main advance in [Ghezelsoflu et al. 2018] is testing a new distribution policy for the carrier in which exporters can also be served after importers, which differs from the street-turn methodology currently adopted by the carrier.

In Fig. 2.2 it is shown an example of two routes for [Ghezelsoflu et al. 2018] and for this study. A paper in which stay-with and drop & pick are adopted together is Funke and Kopfer (2016) [Funke et al. 2016]. In their study, the decision of the way of service is not up to the customer like in this study, but made by the algorithm. However, the methodology presented in this section is flexible to manage instances in which a subset of customers requires a fixed

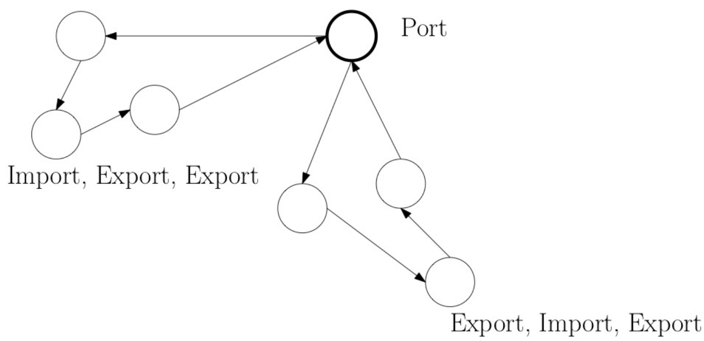


Fig. 2.2 Example of two routes

service type and for the other the algorithm can choose their service type. Several reasons can be seen by the customers to choose one of the two service types. Drop & pick services increase the flexibility of the customers because they can load or unload the container when it can be integrated optimally in their working schedule. Since drop & pick services last less time and can be integrated easier into existing routes, they can be reserved with less advance than stay-with services. Nevertheless, stay-with services need to be considered also because not all customers want to invest in a more expensive equipment that is necessary for drop & pick services. In order to meet customer transportation requests, empty containers need to be provided to and collected from customers. Two ways to provide empty containers are regarded: they are picked up in the port or directly carried from importers to exporters according to the street-turn policy, [Jula 2006] and [Deidda et al. 2008]. Concerning service times at the customer locations, the problem is formulated and solved without time windows.

2.3 Literature review

The recent papers [Song et al. 2017] and [Ghezelsoflu et al. 2018] present a detailed survey about the literature on drayage problems. The new drayage problem presented in this section can be seen as an important extension of the paper of [Ghezelsoflu et al. 2018]. They consider a drayage problem that is motivated by the case study of a real carrier. In their problem statement, the trucks are able to carry more than one container and all customers are served according to the stay-with service type. They show some improvements in the current carrier policy solving the problem in which importers do not have the priority to be served before all exporters. The two additional characteristics of the study presented here are the inclusion of drop & pick services and the introduction of special loads.

A problem which belongs to the class of VRP with clustered backhauls, in which on each route all deliveries have to be made before the pick up services, is studied by [Lai et al. 2013], with the extension of allowing more than one visit for a customer. Another extension of the VRP with backhauls, in which all deliveries have to be made before the pick up services, is made by [Vividovic et al. 2012]. They study a drayage problem with containers of different sizes, 20 ft and 40 ft and a truck can simultaneously carry one 40ft, or two 20ft containers, using an appropriate trailer type. [Dotoli et al. 2016] deal with a Full Truck load Pick-up & Delivery Problem with Clustered Backhauls.

The problem statement formulated by [Funke et al. 2016] has a lot of similarities to the one we present because they use trucks which can carry more than one container and they consider both types of services. Despite this, there are some major differences. The most relevant difference is that they assume that every customer is prepared for a drop & pick service and therefore the cargo transportation company can decide during the optimization process which service they want to use. In our problem, to reach a more realistic setting, the customers are free to decide in advance if they want a stay-with or a drop & pick service, so that all the types of request are fixed before the optimization process.

We consider homogeneous containers of 20 ft size, instead they use 20 ft and 40 ft containers, which can not be coupled with any other container. Hence, it is not possible to model 40 ft containers as special containers because special containers can be coupled with empty containers. One more difference is that they introduce one depot for the storage of empty containers when necessary. In contrast to our drop & pick rules, they assume to end this kind of service on the same day by visiting the customer twice, either with the same truck or another. They solve the problem with a node-arc model for small instances.

[Ileri et al. 2006] consider stay-with and drop & pick orders that are fixed from the beginning like in the presented problem description. They assume that drivers start and end the working day without a trailer. Additionally, they introduce trailer pools, in which infinitely many empty containers are stored. Drivers can be one of two types, namely company driver (CD) and third party (TP). This influences the objective function that is minimizing the total cost for the company. The main difference to the problem under study here is that they have only one container per truck. They use a Set Partitioning formulation and solve the relaxed problem with a column generation approach. Also [Xue et al. 2014] regard both types of service and assume that the customer decides which service to take. They have different additional depots and only one container per truck is allowed. They solve the problem with an algorithm based on window partitions. As much as regards the adoption of both stay-with and drop & pick services, the drayage problem of [Reinhardt et al. 2016] take them into account according to the model they present but their experimentations represent only the possibility of the stay-with service. Also in the Full Truck load Pick-up & Delivery Problem with Clustered Backhauls of [Dotoli et al. 2016] both types of service are possible and this results by matching a delivery with a pick up, which depends in turn on the type of request, the size of the container and their ownership (considering that some requests can be sub-contracted to other carriers when the fleet is not sufficient to satisfy the demand). The fleet of vehicles they consider is heterogeneous and each vehicle is composed by a tractor to which a trailer of different possible dimensions can be coupled.

Drop & pick service is adopted by [Namboothiri et al. 2004], [Namboothiri et al. 2008], [Xue et al. 2015] and [Braekers et al. 2013]. In fact, in [Xue et al. 2015], tractors and trailers can be separated and each customer can be visited twice, either for pick up or delivery service. Each customer node is partitioned into two task nodes, corresponding to the two visits of the tractors to the customer. The two task nodes do not have to be served by the same tractor. In [Braekers et al. 2013], when a loaded container is transported from a container terminal to a consignee, the container is unloaded and the resulting empty container should be picked up before the end of the day. Then, the destination of the empty container can be a container terminal or a shipper.

In the drayage problem of [Nossack et al. 2013], customers are served with stay-with services, that is, containers are loaded and unloaded at customer locations but containers are entirely dropped and picked up at the terminals. Stay-with service is also provided by [Shiri et al. 2016], [Lai et al. 2013] and [Srour et al. 2010].

The possibility of splitting the commodity is considered by [Reinhardt et al. 2016] and [Dotoli et al. 2016]. In particular, [Reinhardt et al. 2016] allow that multiple customers share a container ordering the goods in vertical layers corresponding to each customer. This case has an implication on the order in which customers are visited. In the Full Truck load Pick-up & Delivery Problem with Clustered Backhauls and Split Delivery of [Dotoli et al. 2016], from the split delivery assumption derives that multiple visits at the same customer are allowed for terminating the complete shipment to the customer.

Another attribute which characterizes routing problems is the number of containers each truck can move at a time. In [Namboothiri et al. 2004], [Namboothiri et al. 2008], [Shiri et al. 2016], [Reinhardt et al. 2016] and [Braekers et al. 2013] the capacity is of a single container. In [Lai et al. 2013], trucks can carry one or two containers.

As regards the number of terminal and depots, [Nossack et al. 2013] solve a drayage problem whose setting is with multiple terminals and multiple depots. Depots are used as truck parking space and empty container storage. [Shiri et al. 2016] consider to have multiple truck depots where a limited number of trucks is located, and where these trucks must start at and return to. Empty containers are stored in a empty container depot. [Reinhardt et al. 2016] consider a drayage problem with time windows, several terminals and depots. [Braekers et al. 2013] study a drayage problem with one depot and more terminals, where empty containers are available and can be stored. Routes start and end at the depot.

Some papers deal also with the important problem of truck congestion at the intermodal terminals. [Namboothiri et al. 2008] provide an overview of US seaports terminals characterized by truck congestion at the ports and very high levels of demand for trucking service. In particular, they study the management of a fleet of trucks providing container pickup and delivery service (drayage) to a port with an appointment-based access control system in order to improve the scheduling of truck arrival to and departures from the port. The port operates an access control system where its operating hours are divided into a set of equal-duration time slots. The drayage firm is limited by the slot capacity (an upper bound on the number of truck accesses to the port) during each time slot. They are the first to address routing problems with constraints on the number of times vehicles may access a specific location (or locations) during different time windows. In the drayage problem of [Namboothiri et al. 2004], to model congested access to the port, they delay each tractor arrival at the entry gate using a a time- dependent, exogenously-defined congestion function. [Shiri et al. 2016] analyze a drayage problem in which the intermodal terminal requires trucks to have an appointment. This appointment system defines the number of trucks allowed at the terminal at each time period. [Srour et al. 2010] study a drayage problem inspired by a Dutch logistic service provider. The problem consists of time windows for the customers and for the terminal operating time.

As much as regards formulations and methods, in [Namboothiri et al. 2004], to fulfill all container movement requests within the required time windows and with minimum total transportation cost for the drayage company, their model starts with a Set Partitioning formulation and it is based on the column generation approach with shortest path pricing. [Namboothiri et al. 2008] develop a drayage operations planning approach based on an integer programming heuristic based on column generation that explicitly models a port access control system. The goal of the MIP [Srour et al. 2010] present in an node edge form is to minimize the total routing costs, which consist of time traveling empty and penalties for rejected jobs. [Vividovic et al. 2012] propose two MIP formulations, a multiple assignment problem with time windows and the VRP with simultaneous pickups and deliveries. In the drayage of [Braekers et al. 2013], the objective consists in minimizing the number of vehicles used and the total distance travelled. They assume that waiting is allowed in case a vehicle arrives early at a customer location and it has no costs. The problem is solved with a sequential approach and an integrated approach. In the sequential one, empty container allocations are determined before vehicle routes are created. In the integrated one, empty container allocations are made simultaneously with vehicle routing decisions.

[Lai et al. 2013] propose a metaheuristic based on several local search phases implementing node movements and truck swaps for solving the VRP with clustered backhauls. In the drayage problem with hard time windows constraints of [Nossack et al. 2013] a node-arc based mixed integer programming problem is formulated. They develop a two stage heuristic approach in which, in a first stage, an initial solution is constructed and, in a second stage, improved by an ejection chain heuristic. [Escudero-Santana et al. 2015] solve a drayage problem using the viral system metaheuristic method, a bioinspired approach. In the local container drayage problem of [Xue et al. 2015], the transportation service is performed in a local area near the terminal and it is assumed that empty containers are unlimited. An Ant Colony Optimization algorithm is developed to solve the problem. To deal with a Full Truck load Pick-up & Delivery Problem with Clustered Backhauls, Split Delivery, and hard Time Windows, [Dotoli et al. 2016] used a node arc model. [Reinhardt et al. 2016] modeled their drayage problem with time windows with a set covering formulation. In the modeling they consider constraints for selecting empty container depots and limiting imbalances of empty containers. [Shiri et al. 2016] formulated their drayage problem as a multiple traveling salesman problem with time windows and solve it with an algorithm based on reactive Tabu Search.

In conclusion, to the best of our knowledge, the presented problem statement is innovative and not directly comparable with the other problem statements in the literature. It integrates many realistic characteristics from different papers and is therefore worth to be regarded. In particular, the presented Set Partitioning formulation is very flexible, it can handle a lot of different constraints and can be used for various problem statements. Especially for this problem statement, it is more effective than other methods presented in the literature.

2.4 Modelling

In the following subsection, the sets of customer requests are defined and the feasible routes are illustrated. Next, the Set Partitioning model is presented and a discussion on the size of the realistic problem instances is provided.

2.4.1 Feasibility of routes

The set of customer service types can be defined as follows:

- PE: set of requests in which an empty container must be picked up from an importer during drop & pick service;

- DE: set of requests in which an empty container must be dropped to an exporter during drop & pick service.
- PLO: set of requests in which an ordinary loaded container must be picked up from an exporter during drop & pick service;
- PLS: set of requests in which a special loaded container must be picked up from an exporter during drop & pick service;
- DLO: set of requests in which an ordinary loaded container must be delivered to an importer during drop & pick service;
- DLS: set of requests in which a special loaded container must be delivered to an importer during drop & pick service.
- $PLSWO$: set of requests of exporters with stay-with services and ordinary containers;
- $PLSWS$: set of requests of exporters with stay-with services and special containers;
- $DLSWO$: set of requests of importers with stay-with services and ordinary containers;
- $DLSWS$: set of requests of importers with stay-with services and special containers.

In order to define the feasible sequences of visits among these types of requests, we describe the state of the truck slots after each type of customer request and the types of requests that can be served for each possible state. Let $s_{rj}(i)$ be the state of truck slot j after visiting customer request i in route r. Since trucks carry up to two containers, we have $j \in \{1,2\}$. The state of any container slot is one of the following entries:
$\{Absent, ClosedO, ClosedS, Empty, InitialO, InitialS\}$.
The possible states are defined hereafter:

- $s_{rj}(i) = Absent$ means that no container is placed in slot j. It can be an initial state of the truck slot or the state after serving customer request $i \in DE \cup DLS \cup DLO$. Next, truck slot j can be used to serve a request in $PE \cup PLO \cup PLS$ in route r;
- $s_{rj}(i) = ClosedO$ means that an ordinary loaded container is put in slot j after serving an export request $i \in PLSWO \cup PLO$. In this case, this slot cannot change the state any longer and cannot be used to serve any additional customer request in route r;
- $s_{rj}(i) = ClosedS$ means that a special loaded container is put in slot j after serving an export request $i \in PLSWS \cup PLS$. Even in this case, this slot cannot change the state

any longer along the route and cannot be used to serve any additional customer request. Compared to *ClosedO*, this case constraints the status of the other slot of the truck, that cannot be used for any kind of loaded container;

- $s_{rj}(i) = Empty$ means that the container in slot j is empty. It can be an initial state of the truck slot or the state after serving a customer request $i \in DLSWO \cup DLSWS \cup PE$. Next, this truck slot j can be used to serve a request of set $DE \cup PLSWO \cup PLSWS$ in route r;

- $s_{rj}(i) = InitialO$ means that an ordinary container in slot j is loaded to satisfy an importer request in set $DLSWO \cup DLO$ in route r. Next, truck slot j will be denoted by the state $Empty$ or $Absent$;

- $s_{rj}(i) = InitialS$ means that a special container in slot j is loaded to satisfy an importer request in set $DLSWS \cup DLS$. Next, truck slot j will be denoted by the state $Empty$ or $Absent$.

According to this notation, the possible initializations of the two container slots in a truck leaving the port are:

- InitialO & InitialO
- InitialO & Empty
- InitialO & Absent
- InitialS & Empty
- InitialS & Absent
- Empty & Empty
- Empty & Absent
- Absent & Absent

In the list of possible truck initializations it is taken into account that the special loaded containers can be carried alone in a truck or be coupled only with empty containers. Table 2.1 summarizes the acceptable sequences of visit for each type of request, which is reported in the central column and denoted by *Type of request*. Column *Previous* shows which types of request can be visited before the considered one and the slot state that can be used to meet the considered type of request. Column *Next* shows which types of requests can be visited after the considered one or shows the slot state after serving the considered type of request. For example, the first row shows the possible sequences $\{Absent, PLO, ClosedO\}$, $\{DLO, PLO, ClosedO\}$, $\{DLS, PLO, ClosedO\}$, $\{DE, PLO, ClosedO\}$ for PLO.

Previous	*Type of request*	*Next*
Absent DLO DLS DE	**PLO** (**PLS**)	ClosedO (ClosedS)
InitialO (InitialS)	**DLO** (**DLS**)	PLO PLS PE Absent
Empty DLSWO DLSWS PE	**PLSWO** (**PLSWS**)	ClosedO (ClosedS)
InitialO (InitialS)	**DLSWO** (**DLSWS**)	DE PLSWO PLSWS
Absent DLO DLS DE	**PE**	PLSWO PLSWS DE
Empty DLSWO DLSWS PE	**DE**	PLO PLS PE Absent

Table 2.1 Possible sequences of visits for each truck slot

To determine feasible routes for two-containers trucks, both container slots must be taken into account. Table 2.2 shows in column *Slot pre-condition* the required state of a slot in order to serve the type of request reported in the first column *Type*. For example, the state of a slot must be "Absent" in order to pick an empty container (i.e. serving a request of set PE) and put the container in that slot. In the column *Pre-condition other slot* the status of the other slot in the truck is indicated in order to accept the type of request reported in the first column. The string "Irrelevant" shows that the ability to serve a request of set PE is independent from the state of the other slot. Column *Slot Post-condition* indicates the state of the used slot after the service of a customer request reported in the corresponding line. For example, the state of a slot must be "Empty" after picking up an empty container (i.e. serving a request of set PE). To clarify, one can serve a request of set PLO by a truck slot if the state of the other slot is different from $ClosedS$ and $InitialS$.

The set of feasible routes is obtained by connecting acceptable sequences of customer requests according to Table 2.1 and Table 2.2. Therefore a dynamic program over the number

Table 2.2 Changes in slot states

Type	Slot		Other slot
	Pre-condition	Post-condition	Pre-condition
PE	Absent	Empty	Irrelevant
DE	Empty	Absent	Irrelevant
PLO	Absent	ClosedO	not ClosedS & not InitialS
PLS	Absent	ClosedS	not ClosedO & not ClosedS & not InitialO & not InitialS
DLO	InitialO	Absent	Irrelevant
DLS	InitialS	Absent	Irrelevant
PLSWO	Empty	ClosedO	not ClosedS & not InitialS
PLSWS	Empty	ClosedS	not ClosedO & not ClosedS & not InitialO & not InitialS
DLSWO	InitialO	Empty	Irrelevant
DLSWS	InitialS	Empty	Irrelevant

of customers in a route is used. To obtain routes with $k+1$ customers, all routes with k customers are regarded and for every customer it is checked if adding this customer is feasible. This has to be done for all possible truck initializations because there exist routes that are only feasible for one of the initializations.

2.4.2 Node-arc formulation

In line with the majority of the existing modeling ways for drayage problems, we also looked for a node-arc formulation. Starting from a basic formulation, other constraints are gradually added to model the problem requirements.

Following the notation,

N: Set of nodes;

A: Set of arcs;

K: Set of trucks;

$S =$ $\{0, \dots, S_{max}+1\}$ set of stops (positions of the nodes/customers) in a route;

d_{ij}: Cost of the arc (i, j);

$\Delta^+(i)$: Forward star of node i;

$\Delta^-(i)$: backward star of node i;

x_{ijk}: Decision variable equal to 1 if arc (i, j) is covered by vehicle k, 0 otherwise;

$y_{i,s,k}$: Decision variable equal to 1 if customer i is served by truck k at stop s, 0 otherwise.

a basic node-arc formulation is given:

$$\min \sum_{k\in K}\sum_{(i,j)\in A} d_{ij}x_{ijk} \tag{2.1}$$

$$\sum_{k\in K}\sum_{j\in\Delta^+(i)} x_{ijk} = 1 \quad \forall i \in N\setminus\{0\} \tag{2.2}$$

$$\sum_{i\in\Delta^-(j)} x_{ijk} - \sum_{i\in\Delta^+(j)} x_{jik} = 0 \quad \forall k \in K, \forall j \in N \tag{2.3}$$

$$\sum_{j\in\Delta^+(0)} x_{0jk} \le 1 \quad \forall k \in K \tag{2.4}$$

$$\sum_{s\in S} y_{i,s,k} - \sum_{j\in\Delta^+(i)} x_{ijk} = 0 \quad \forall i \in N\setminus\{0\}, \forall k \in K \tag{2.5}$$

$$\sum_{(i,j)\in A} x_{ijk} \le S_{max}+1 \quad \forall k \in K \tag{2.6}$$

$$y_{i,0,k} = 0 \quad \forall i \in N\setminus\{0\}, \forall k \in K \tag{2.7}$$

$$y_{i,S_{max}+1,k} = 0 \quad \forall i \in N\setminus\{0\}, \forall k \in K \tag{2.8}$$

$$y_{j,s+1,k} - y_{i,s,k} \ge x_{ijk} - 1 \qquad \forall i \in N\setminus\{0\}, \forall j \in \Delta^+(i), \forall k \in K, \forall s \in S \tag{2.9}$$

$$x_{ijk} \in \{0,1\} \qquad \forall k \in K, (i,j) \in A \tag{2.10}$$

$$y_{i,s,k} \in \{0,1\} \qquad \forall i \in N, \forall s \in S, \forall k \in K \tag{2.11}$$

The minimum total cost has to be found (2.1). Each customer is served by exactly one vehicle (2.2). Flow on the route followed by truck k is characterized by (2.3). Constraints (2.4) ensure that every vehicle makes at most one route. The sub-tour elimination constraints are set as (2.5). In fact, fixing $i = \tilde{i}$ and $k = \tilde{k}$ and assuming that there is at least a cycle, then at least the two s_u and s_v exist such that $y_{\tilde{i},s_u,\tilde{k}} = 1$ and $y_{\tilde{i},s_v,\tilde{k}} = 1$. This means that $\sum_{j\in\Delta^+(i)} x_{ijk}$ is at least 2, breaking the constraints (2.2). Constraints (2.2) together with (2.5) ensure that each customer is served at exactly one stop and there is a stop at a customer just when there is an arc going to that customer. In the formulation we need to know at which point the route can be stopped, fixing the maximum number of stops S_{max} by (2.6). An example of S_{max} can be the total number of customers in the problem, which is also used in the implementation of the presented drayage problem in Section 2.7. Constraints (2.7) and (2.8) impose that, at stop

0 and after the maximum number of stops, trucks are at the depot and no customer is served. In particular, constraints (2.8) are very important because (2.5) alone do not guarantee cycles after the maximum number of stops S_{max}. Forcing the vehicle to terminate the route coming back to the port after S_{max}, (2.8) avoids the formation of additional cycles. Using (2.5) and (2.8) in this model, it follows that subtours are not possible. The binary decision variables, in (2.10)-(2.11), are connected by constraints (2.9). In fact, if arc (i, j) is traversed by vehicle k then $x_{ijk} = 1$. This means there is a stop s at which customer i is served by vehicle k ($y_{i,s,k} = 1$) and j follows ($y_{j,s+1,k} = 1$) and (2.9) are valid. If $x_{ijk} = 0$, (2.9) are also valid.

The variable $y_{i,s,k}$ is in general not necessary for modeling the majority of the VRPs. In this case it has been used to introduce the following constraints required by the drayage problem under study.

Given the additional notation,

$a_{i,k}$: number of Absent at time i on truck k;

$b_{i,k}$: number of Empty at time i on truck k;

$c_{i,k}$: number of InitialO at time i on truck k;

$d_{i,k}$: number of InitialS at time i on truck k;

$e_{i,k}$: number of ClosedO at time i on truck k;

$f_{i,k}$: number of ClosedS at time i on truck k.

the following constraints guarantee the fulfillment of the service by truck k at customer c at time s in accordance with the status of the slots updated at time $s-1$, refer to Table 2.2.

We will refer to these constraints as "pre-condition constraints".

$$y_{c,s,k} \leq a_{s-1,k} \quad \forall c \in \{PE, PLO, PLS\}, \forall k, \forall s \geq 1 \tag{2.12}$$
$$y_{c,s,k} \leq b_{s-1,k} \quad \forall c \in \{DE, PLSWO, PLSWS\}, \forall k, \forall s \geq 1 \tag{2.13}$$
$$y_{c,s,k} \leq c_{s-1,k} \quad \forall c \in \{DLO, DLSWO\}, \forall k, \forall s \geq 1 \tag{2.14}$$
$$y_{c,s,k} \leq d_{s-1,k} \quad \forall c \in \{DLS, DLSWS\}, \forall k, \forall s \geq 1 \tag{2.15}$$
$$y_{c,s,k} \leq 1 - f_{s-1,k} \quad \forall c \in \{PLO, PLS, PLSWO, PLSWS\}, \forall k, \forall s \geq 1 \tag{2.16}$$
$$y_{c,s,k} \leq 1 - d_{s-1,k} \quad \forall c \in \{PLO, PLS, PLSWO, PLSWS\}, \forall k, \forall s \geq 1 \tag{2.17}$$
$$2y_{c,s,k} \leq 2 - c_{s-1,k} \quad \forall c \in \{PLS, PLSWS\}, \forall k, \forall s \geq 1 \tag{2.18}$$
$$2y_{c,s,k} \leq 2 - e_{s-1,k} \quad \forall c \in \{PLS, PLSWS\}, \forall k, \forall s \geq 1 \tag{2.19}$$

In (2.12) for example, if customer c has to be served at time s by the truck k, then at least one status slot of truck k has to be Absent in order to satisfy one of the request types PE or PLO or PLS.

The following constraints are referred to as "Update Status constraints", refer to Table 2.1.

$$a_{s,k} = a_{s-1,k} - \sum_{c \in \{PE,PLO,PLS\}} y_{c,s,k} + \sum_{c \in \{DE,DLO,DLS\}} y_{c,s,k} \quad \forall k, \forall s \geq 1, \tag{2.20}$$
$$b_{s,k} = b_{s-1,k} - \sum_{c \in \{DE,PLSWO,PLSWS\}} y_{c,s,k} + \sum_{c \in \{PE,DLSWO,DLSWS\}} y_{c,s,k} \quad \forall k, \forall s \geq 1 \tag{2.21}$$
$$c_{s,k} = c_{s-1,k} - \sum_{c \in \{DLO,DLSWO\}} y_{c,s,k} \quad \forall k, \forall s \geq 1 \tag{2.22}$$
$$d_{s,k} = d_{s-1,k} - \sum_{c \in \{DLS,DLSWS\}} y_{c,s,k} \quad \forall k, \forall s \geq 1 \tag{2.23}$$
$$e_{s,k} = e_{s-1,k} + \sum_{c \in \{PLO,PLSWO\}} y_{c,s,k} \quad \forall k, \forall s \geq 1 \tag{2.24}$$
$$f_{s,k} = f_{s-1,k} + \sum_{c \in \{PLS,PLSWS\}} y_{c,s,k} \quad \forall k, \forall s \geq 1 \tag{2.25}$$

Constraint (2.20) imposes that the number of absent containers at time stop s in the truck k is equal to the number of absent containers at the previous time stop $s-1$ minus the number of occupied slots caused by services like PE, PLO, PLS, plus the number of new absent caused by services like DE, DLO, DLS. The same idea is applied to $b_{s,k}, c_{s,k}, d_{s,k}, e_{s,k}, f_{s,k}$ in order to update at every time stop the status of all trucks.

The following constraints are referred to as "Status initialization constraints", refer to the discussion at Section 2.4.1.

$$a_{0,k} + b_{0,k} + c_{0,k} + d_{0,k} = 2 \quad \forall k \in K \tag{2.26}$$

$$c_{0,k} + 2d_{0,k} \leq 2 \quad \forall k \in K \tag{2.27}$$

$$e_{0,k} = f_{0,k} = 0 \quad \forall k \in K \tag{2.28}$$

Constraints (2.26) sets the possible initial status of the two slots of every truck to Absent, Empty, InitialO and InitialS. With (2.27), two InitialO are possible, but InitialS cannot be coupled neither with InitialO nor InitialS. ClosedO and ClosedS are not status initialization (2.28).

Following on from the disadvantages of the node-arc formulations stated in Chapter 1, in the presented formulation the number of additional variables needed to model the drayage problem under study increases the running time and clearly constitutes a disadvantage for the implementation, as later can be seen from the experimentation in Table 2.5. The subtour elimination constraints obtained with (2.5) and (2.8) do not represent a problem instead.

The additional variables needed in the formulation translate in the case of the Set Partitioning into a high number of constraints from which it can benefit. We prefer then the Set Partitioning model over the node-arc formulation for the presented problem statement because the problem is highly constrained and with a growing number of constraints the running time of the route generation and of the Set Partition model does not increase in general, as opposed to node-arc formulations that become harder to solve.

2.4.3 Set Partitioning formulation

Given the previous set of feasible routes, we present a Set Partitioning model using the following notation:

R: Set of all feasible routes

C: Set of customer requests $= PE \cup DE \cup PLS \cup PLO \cup DLS \cup DLO \cup PLSWS \cup PLSWO \cup DLSWS \cup DLSWO$

K: Maximum number of trucks;

d_r: Cost of route $r \in R$;

α_{ir}: Coefficient with value 1 if route $r \in R$ covers $i \in C$, 0 otherwise;

x_r : Binary variable which is 1 if route $r \in R$ is selected, 0 otherwise.

The Set Partitioning formulation of the model is:

$$\min \quad \sum_{r \in R} d_r x_r \tag{2.29}$$

$$\sum_{r \in R} x_r \leq K \tag{2.30}$$

$$\sum_{r \in R} \alpha_{ir} x_r = 1 \quad \forall i \in C \tag{2.31}$$

$$x_r \in \{0,1\} \qquad \forall r \in R \tag{2.32}$$

In (2.29) the costs of the selected routes are minimized. Constraint (2.30) fixes the limit K of the number of trucks to use. Constraints (2.31) ensure that each container requests is met by exactly one route. Finally, (2.32) defines the domain of the decision variable.

2.4.4 Parameters affecting the number of feasible routes

The ability of solving the previous model by a standard mixed-integer programming solver depends on the number of feasible routes. A general formula is derived to determine $|R|$ as function of the types of requests, but it is too large to be reported in the thesis, see Appendix A. In the basic problem statement it is possible to compute the exact number of routes in less than 1 second. For the problem with 8 hours working day constraints and other special constraints deriving from the heuristics (Section 2.5), we estimate the number of routes using sampling. The knowledge of the number of routes for a given instance is important to decide which algorithm to apply. More details about these formulas are provided in Appendix A. Nevertheless, the maximization of this function shows that, for a fixed number of customers, the number of feasible routes is maximum when half of the customers requests belongs to the set PE and the other half belongs to DE. Indeed, a problem with a huge number of customers and less PE and DE requests could have a lower number of feasible routes than a problem with less customers but more PE and DE requests. Table 2.3 shows the number of the feasible routes for four instances with special constraints on the types of requests, reported in the column *Type of request*, in the case of up to 8 customers per route. In the first column the number of customers $|C|$ for every instance is indicated. The column $|R|_{max}$ represents the maximum number of feasible routes. In the first and third instances, the maximum is reached

$\lvert C\rvert$	*Type of request*	$\lvert R\rvert_{max}$
20	without *PE* - *DE*	20440
20	half *PE* half *DE*	1,4 billion
100	without *PE* - *DE*	12,5 million
100	half *PE* half *DE*	50 trillion

Table 2.3 Example for number of feasible routes

when half of the customers requests are *PLO* and the other half are *DLO*. In the second and fourth instances, the maximum is reached when half of the customers requests are *PE* and the other half are *DE*. This relevant role of *PE* and *DE* requests is also confirmed by Table 2.1. It shows that *PE* and *DE* requests result in a higher number of acceptable sequences of visits as opposed to the other types of requests.

2.5 Heuristics

In this section, two heuristics are proposed to select a subset of routes from the overall set R of feasible routes. This subset of routes will replace R in the Set Partitioning model of Section 2.4.3 and result in a restricted Set Partitioning problem. Therefore, these heuristics can be seen as rules to delete routes from R which are not likely to appear in the optimal solution. The heuristics are described by the example in Table 2.4, where only 4 customer requests are considered. They are denoted by $C1$, $C2$, $C3$ and $C4$.

Name	*Type of request*	*Coordinates*
Port	-	$(0,0)$
C1	*PE*	$(0,1)$
C2	*DE*	$(1,1)$
C3	*DE*	$(1,0)$
C4	*PLS*	$(2,0)$

Table 2.4 Example for the heuristics

- *Heuristic 1.* Since customers of type PE and DE highly increase the total number of routes, the first heuristic requires a limit lim on the number of customers of these types in each route. For instance, if $lim = 2$, only routes with at most 2 customers which have type PE or DE are regarded. Following this idea, one eliminated route is $\{Port, C1, C2, C3, Port\}$ because this route contains three customers with type *PE* or *DE*, which is more than the allowed limit lim. A larger value of lim leads to more precision but to a higher running time.

- *Heuristic 2*. For every customer request $i \in C$, we make a list of the closest $m \leq |C| - 2$ requests. Then, in the routes generation process, a candidate can only be added to the route only if it is in the neighbor list of the previous customer in the route. This is motivated by the fact that a high number of long distance routes, which are not likely in the optimal solution, are not regarded. The choice of adding a new customer to a route looking at the number m of neighbors of the previous customer in the route and not at the customers placed inside a fixed distance from the previous customer in the route is because we can control the number of candidates in the list and avoiding extreme cases of very few or too many customers in the lists. To ensure feasibility, the port can always be added to a route and from the port it is allowed to reach every customer. In our example, see Table 2.4, if we set $m = 2$, then one eliminated route is $\{Port, C1, C4, Port\}$ because $C4$ is not one of the m closest neighbors of $C1$. A larger value of m leads to more precision but to a higher running time.

- *Combination of Heuristic 1 and Heuristic 2*. The rules described for Heuristic 1 and Heuristic 2 are applied together.

2.6 Column generation approach

In this section the column generation approach is used for solving the presented drayage problem. In Section 2.6.1, an adapted labeling algorithm for solving the pricing represented by the shortest path problem with resource constraints is presented. The lower bound provided by the column generation algorithm is then used in a branch-and-bound algorithm to compute the integer solution. The branch-and-price (column generation embedded in a branch-and-bound) has been tested and the final solutions are presented in Section 2.7.

2.6.1 Pricing problem

For the VRPs, the pricing problem is in general solved by a shortest path problem with resource constraints (SPPRC) or an elementary shortest path problem with resource constraints (ESPPRC). These problems consist in finding the shortest path from a source to a sink node satisfying a set of constraints defined over a set of resources, for example, time-windows constraints, drayage constraints etc. In particular, in the elementary SPPRC, the shortest path does not pass through the same node twice, so that cycles are not allowed. Considering the drayage case and the form of the reduced cost (1.16) in Chapter 1, the pricing under study consists in finding the path from the port to the port itself in which each node/customer has a prize/weight λ_i associated with it and the cost of the path minus the sum of the prizes counted for the customers served, has to be minimized. The description of this optimization problem is consistent with the goal of the Prize Collecting Traveling Salesman Problem (PCTSP) and it can be also seen as a shortest path problem with resource constraints (SPPRC). This can be done replacing each arc cost d_{ij} with $d_{ij} - \lambda_j$, that is, subtracting to the cost of the arc (i, j) the prize λ_j for each customer j added to the path.

To solve these problems avoiding a complete enumeration of all the feasible routes, a labeling algorithm with dominance rules with which non-optimal paths are eliminated can be used. Dominance rules can be described for both SPPRC and ESPPRC [Irnich et al. 2005] and dominated rules can be deleted.

More precisely, in the labeling algorithms, multi-dimensional resource vectors called labels represent the partial paths starting from the port. In our case of drayage problem the components are the reduced costs of the partial path and the status of the two slots of the vehicles at the customer to which the partial paths end. For every partial path we have to check if it is feasible and if it dominates other routes or if it is dominated by other routes.

Algorithm 2 shows the labeling algorithm steps. In the algorithm the port from which the routes start is denoted by 0 and the same port to which the routes end as d. The aim of the algorithm is to find the shortest path starting from the port 0 and finishing to the port itself d. Algorithm 2 follows the notation of the algorithm presented in the lecture of [Desaulniers 2018] and it is adapted for the drayage problem under study. In Algorithm 2, the components of the label L are in order: the reduced cost, the status of the first slot, the status of the second slot at the last node of the path associated with label L. The set U_0 is set following the truck initialization presented in Section 2.4.1. In line 12 of Algorithm 2, the dominance rules are used.

To explain how the dominance rules work, let us consider two paths, p_1 and p_2. A path p_1 dominates a path p_2 if both are ending at the same node and the reduced costs of p_1 are lower or equal than the reduced costs of p_2 and if all paths to which p_2 can be extended can also arise from p_1. If p_1 dominates p_2, p_2 can be discarded. In particular, if p_1 dominates p_2 and p_2 dominates p_1, one of them can be discarded.

For a time window constraint with possible waiting times, p_1 dominates p_2 if they end at the same node, the reduced cost of p_1 is lower or equal than the reduced cost of p_2 and the arriving time at the last customer of p_1 is lower or equal than the arriving time at the last customer of p_2.

In the drayage problem here presented, we do not consider time windows. In order to adapt the dominance rules to the constraints of the drayage problem presented in this chapter, a new rule must be added.

Algorithm 2 Labeling algorithm

Require: Set U_i of unprocessed labels at node i; set P_i of processed labels at node i; $i(L)$ last node of the path associated with label L; $DOM(U_j, P_j)$ dominance algorithm applied to labels in U_j and P_j that returns a possibly reduced set U_j.

Ensure: Shortest 0-d path.

```
Set U_i = P_i = ∅   ∀i ∈ C \ 0
U_0 = {(0, InitialO, InitialO), (0, InitialO, Empty), (0, InitialO, Absent),
(0, InitialS, Empty), (0, InitialS, Absent), (0, Empty, Empty), (0, Empty, Absent),
(0, Absent, Absent)}
while ⋃_{i∈C\0} U_i ≠ ∅ do
    Choose a label L ∈ ⋃_{i∈C\0} U_i and remove L from U_{i(L)}
    for all arcs (i(L), j) ∈ A do
        Extend L along (i(L), j) creating a new label L'
    end for
    if L' is feasible then
        Add L' to U_j
        U_j = DOM(U_j, P_j)
    end if
    Add L to P_{i(L)}
end while
Filter P_d to find a shortest 0-d path.
```

Theorem 1. *Let p_1 and p_2 be two routes that end at the same customer. Let a_1 be the number of absent slots that the truck has after serving the last customer in p_1, e_1 be the number of empty slots, io_1 the number of initial ordinary loads, is_1 the number of initial special loads, cs_1 the number of closed special slots and co_1 the number of closed ordinary slots (a_2, e_2, io_2, is_2, cs_2, co_2 for p_2 respectively). All customers or partial paths which can extend p_2 can also extend p_1 (in other words, p_1 can dominate p_2) if the following conditions are met:*

- $a_1 \geq a_2$
- $e_1 \geq e_2$
- $io_1 = io_2$
- $is_1 = is_2$
- $cs_1 \leq cs_2$
- $co_1 \leq co_2$

Proof. Assuming these conditions, the customers still not visited that require an absent slot or an empty slot or an initial ordinary load or an initial special load can be served by p_1 because in all possible partial paths p_1 has at least as many absent, empty or initial as p_2 at every point in time. In case some customers cannot be served by p_1 because the loads of the status initial special or closed special or closed ordinary forbid that another load is taken, then these customers cannot be served by p_2 too. □

With the additional requirements introduced in Theorem 1 for the drayage constraints in comparison with the standard dominance rules, less routes can be dominated. In fact, if p_1 dominates p_2 considering the additional requirements, then p_1 dominates p_2 also with the standard dominance rules but it can be found p_2 such that p_1 dominates p_2 with the standard dominance rules and p_1 does not dominate p_2 with the additional requirements of Theorem 1. We can think for example to two paths $p_1 = \{1,2,3\}$ and $p_2 = \{1,3\}$ with reduced cost $r_1 = -2$ and $r_2 = -1$, where customer 1 request is *DLO*, customer 2 *PLO*, customer 3 *PLO* and customer 4 *PLO*. Paths p_1 and p_2 finish to the same customer 3, the reduce cost of p_1 is smaller than the reduced cost of p_2 but p_2 can be extended to customer 4 as opposed to p_1. With the standard requirements for the classic VRP both paths can be extended to customer 4. From the fact that less routes can be dominated, it follows then that the labeling algorithm becomes slower. On the other hand however, because of the drayage constraints some routes are already discarded. In fact, not all customers represent a feasible extension of the paths if they do not meet the vehicle status and the service types requirements.

Concerning a comparison between the SPPRC and the ESPPRC, we can say that using the SPPRC is much faster than using the ESPPRC because cycles are allowed and therefore more routes can be dominated. The advantage of the ESPPRC is that it gives better bounds. Some preliminary tests adapted to our problem have been performed to compare the SPPRC with the ESPPRC. The first performed way better than the second and this is the reason why the SPPRC is chosen.

Since in the SPPRC cycles are allowed, the labeling algorithm tends to create very long path using always the same customers in intermediate steps. This can be time consuming and these routes are often not taken at the end of column generation algorithm. Therefore the algorithm using the SPPRC can be improved significantly by introducing a heuristic pricing.

The *heuristic pricing* consists also in searching a SPPRC with the additional requirement that only x customers per route are allowed (in the experimental result in Section 2.7 $x = 6$). When the heuristic pricing does not find new columns, the exact pricing has to be executed.

2.7 Experimentation

This section presents the computational results of the methods presented for the drayage problem. It compares first the three exact methods obtained with the node-arc formulation of Section 2.4.2, the Set Partitioning formulation of Section 2.4.3 and the column generation approach with the branch-and-bound of Section 2.6. Next the heuristic methods of Section 2.5 are compared.

The presented models are implemented in Java 1.8.0_212 using IBM ILOG CPLEX Studio 12.9. Tests have been run on an Intel Xeon CPU E5-2680 v2 with 2.80 GHz. The time limit is set to 10 minutes and memory limit to 32 GB.

The instances are built fixing the number of *PE* and *DE* requests per setting and assigning the other 8 types of requests with probability of $\frac{1}{8}$ each. Concerning customer coordinates, the x-coordinate is a random integer between 0 and 1000, the y-coordinate is a random integer between -500 and 500. The port has coordinates $(0,0)$. Therefore, instances are built by a square of 1000x1000 units where the port is placed in the middle of one edge of the square. This choice reflects the situation of the shipment between a port and the customers in the hinterland of the port.

In the experimentation we do not assume time windows.

For the heuristic pricing, it is a good compromise to set the maximum number of customers per route allowed $x = 6$.

Reading the results Table 2.5 and Table 2.6 show the results of the method presented in this Chapter of the thesis. They have the same structure and are built on the same instances. Each row of the table represent a setting of 10 instances in which the number of customers $|C|$ (first subcolumn of the column "Setting"), the number of *PE* requests "*#PE*" (second subcolumn of "Setting") and the number of *DE* requests "*#DE*" requests (third subcolumn of "Setting") are specified. Then, all averages results on the 10 setting of the instances are made over the instances solved within the time and memory limits. The subcolumn "time" represents the average running time in seconds over the 10 instances for the method specified in the column of the table. The subcolumn "s.i." represents the number of instances out of the 10 solved within the time limit.

In particular, in Table 2.5, the subcolumn "*#routes*" of column "Set Partitioning" represents the average number of feasible routes obtained with the Set Partitioning; the subcolumn "*#nodes*" of column "Branch-and-price" represents the average final number of nodes used in the branch-and-bound; the subcolumn "gap at rn" represents the average gap at root node (between the relaxation and the exact value at the root node) for the branch-and-bound. Since this table shows the results of the exact methods, the objective values are omitted because they are always reached to the optimal, a part in one instance setting with 60 customers and 3 *PE* and 2 *DE*, in which Cplex stopped the Set Partitioning to a non optimal feasible solution.

Concerning the results shown in Table 2.5, it can be seen that the node-arc formulation does not solve the majority of the instances. The Set Partitioning formulation is useful for instances with less *PE* and *DE* customer requests. The column generation is more effective for instances with more *PE* and *DE* customers. The running time for the branch-and-price is more volatile than for the Set Partitioning: it can happen that some instances are solved very fast and others not within the time limit. Finally we can say that for instances with less than 4 *PE* and *DE* customer requests it is better to use the Set Partitioning. For instances with at least 6 *PE* or *DE* customer requests, the branch-and-price is better. For instances with 5 *PE* and *DE* customer requests, the branch-and-price is faster but less reliable in terms of solved instances within the limits.

Concerning the heuristics methods presented in Section 2.5, Table 2.6 shows the results obtained using Heuristic 1 setting $lim = 2$ (column "Heuristic 1"), Heuristic 2 setting $m = 0.75|C|$ (column "Heuristic 2") and the combination of the two heuristics, (column "Heuristic 3"). The subcolumn "gap" represents the difference between the objective value obtained with the method specified in the column and the objective value obtained with an exact method if possible or with the best heuristic otherwise. All the three heuristics are much faster as opposed to the case of complete enumeration of all feasible routes (that can be seen in the column "Set Partitioning" of Table 2.5), without losing much precision. In particular, Heuristic 1 becomes slower with the increase of customers, Heuristic 2 becomes slower with the increase of *PE* and *DE* customers, Heuristic 3 is always fast.

Setting			Set Partitioning			Branch-and-price				Node-arc formulation	
$\lvert C\rvert$	$\#PE$	$\#DE$	time	s.i.	# routes	time	s.i.	# nodes	gap at rn	time	s.i.
10	1	0	0.2	10	213.3	0.5	10	19.4	0.02	-	0
10	1	1	0.3	10	718.6	0.5	10	11.8	0.02	-	0
10	2	1	0.3	10	1173.6	0.9	10	30.2	0.03	-	0
10	2	2	0.4	10	4240.7	1.2	10	74.0	0.03	-	0
10	3	2	0.7	10	13324.6	1.9	10	74.2	0.04	-	0
10	3	3	1.3	10	56854.2	4.8	10	248.6	0.09	-	0
10	4	3	1.9	10	124571.5	5.2	10	164.0	0.08	320.7	1
10	4	4	5.9	10	519611.6	3.8	10	49.0	0.03	356.4	1
20	1	0	0.3	10	1272.8	1.1	10	30.6	0.01	-	0
20	1	1	0.5	10	4347.1	2.2	10	156.0	0.01	-	0
20	2	1	0.7	10	12203.9	4.8	10	293.2	0.02	-	0
20	2	2	1.7	10	72047.3	6.7	10	244.8	0.02	-	0
20	3	2	2.9	10	170068.3	34.7	10	8395.2	0.03	-	0
20	3	3	10.3	10	933784.3	8.4	10	402.8	0.02	-	0
20	4	3	43.5	10	3852510.8	13.7	9	217.7	0.03	-	0
20	4	4	235.1	2	12625999.0	27.4	9	626.8	0.02	-	0
30	1	0	0.8	10	9515.1	30.1	10	3991.4	0.01	-	0
30	1	1	1.2	10	22013.9	23.8	10	3276.4	0.01	-	0
30	2	1	2.0	10	66421.5	9.1	9	1016.3	0.01	-	0
30	2	2	5.1	10	330123.8	27.9	9	2361.9	0.01	-	0
30	3	2	15.1	10	1255465.5	13.7	10	471.0	0.01	-	0
30	3	3	77.8	10	5981590.8	56.2	10	3471.2	0.02	-	0
30	4	3	161.0	3	11753644.7	119.6	9	7551.7	0.02	-	0
30	4	4	-	0	-	68.3	10	3421.4	0.02	-	0
40	1	0	1.4	10	26097.9	49.3	10	4433.6	0.01	-	0
40	1	1	2.3	10	82433.4	38.2	9	2570.6	0.01	-	0
40	2	1	4.1	10	227331.4	128.0	9	6250.8	0.01	-	0
40	2	2	10.7	10	798149.5	56.3	7	2528.4	0.01	-	0
40	3	2	46.6	10	3499784.2	40.3	8	1440.5	0.01	-	0
40	3	3	203.4	7	12314072.7	183.0	10	8278.0	0.01	-	0
40	4	3	-	0	-	72.0	10	1896.2	0.01	-	0
40	4	4	-	0	-	53.1	7	1365.3	0.01	-	0
50	1	0	2.0	10	44428.0	66.4	8	4435.5	0.01	-	0
50	1	1	3.9	10	175562.4	83.5	4	2882.0	0.01	-	0
50	2	1	8.5	10	503619.1	113.2	8	5104.2	0.01	-	0
50	2	2	28.0	10	1961953.5	143.2	9	4335.9	0.01	-	0
50	3	2	113.0	10	7722791.2	81.1	6	2194.7	0.01	-	0
50	3	3	-	0	-	176.4	6	5173.0	0.01	-	0
50	4	3	-	0	-	159.4	5	3617.0	0.01	-	0
50	4	4	-	0	-	33.6	5	380.2	0.00	-	0
60	1	0	3.5	10	97152.0	112.5	7	4767.9	0.01	-	0
60	1	1	6.8	10	334499.5	102.8	4	2972.5	0.01	-	0
60	2	1	17.5	10	1086615.7	170.3	6	4613.3	0.00	-	0
60	2	2	64.9	10	4063593.6	208.2	6	5144.7	0.01	-	0
60	3	2	260.0	7	13570967.7	60.3	4	526.5	0.00	-	0
60	3	3	-	0	-	124.3	4	2324.5	0.00	-	0
60	4	3	-	0	-	143.9	1	1717.0	0.01	-	0
60	4	4	-	0	-	252.8	2	4455.0	0.01	-	0

Table 2.5 Results for the exact methods

Setting			Heuristic 1				Heuristic 2				Heuristic 3			
$\|C\|$	$\#pe$	$\#de$	time	gap	s.i.	# routes	time	gap	s.i.	# routes	time	gap	s.i.	# routes
10	1	0	0.2	0.00	10	176.6	0.2	0.03	10	88.8	0.2	0.03	10	88.8
10	1	1	0.3	0.00	10	499.0	0.2	0.02	10	152.3	0.2	0.02	10	152.3
10	2	1	0.2	0.00	10	560.4	0.2	0.01	10	282.4	0.2	0.01	10	221.1
10	2	2	0.3	0.02	10	719.4	0.2	0.02	10	486.1	0.2	0.03	10	260.4
10	3	2	0.3	0.08	10	897.5	0.3	0.02	10	1004.9	0.2	0.10	10	305.2
10	3	3	0.3	0.24	10	951.5	0.3	0.02	10	2027.2	0.2	0.24	10	357.4
10	4	3	0.2	0.37	10	670.7	0.3	0.04	10	3306.5	0.2	0.38	10	272.5
10	4	4	0.2	0.64	10	535.9	0.4	0.07	10	4907.8	0.2	0.64	10	243.3
20	1	0	0.3	0.00	10	1160.1	0.3	0.00	10	620.0	0.3	0.00	10	620.0
20	1	1	0.5	0.00	10	3816.0	0.4	0.00	10	1746.8	0.4	0.00	10	1746.8
20	2	1	0.6	0.00	10	5753.0	0.4	0.01	10	2693.3	0.4	0.01	10	2099.2
20	2	2	0.8	0.00	10	13226.9	0.8	0.00	10	14900.0	0.5	0.01	10	4809.0
20	3	2	0.8	0.00	10	11400.7	1.1	0.00	10	22155.5	0.5	0.00	10	4003.7
20	3	3	0.8	0.00	10	14215.0	1.6	0.00	10	51140.3	0.5	0.01	10	4572.7
20	4	3	0.9	0.01	10	16163.0	2.6	0.00	10	142171.6	0.5	0.01	10	4888.3
20	4	4	0.9	0.02	10	18639.6	10.5	0.00	10	852606.5	0.6	0.03	10	6789.6
30	1	0	0.8	0.00	10	8418.4	0.5	0.00	10	3409.6	0.5	0.00	10	3409.6
30	1	1	1.2	0.00	10	20269.8	0.7	0.00	10	5670.7	0.7	0.00	10	5670.7
30	2	1	1.6	0.00	10	33467.4	1.0	0.00	10	12256.0	0.9	0.00	10	9161.3
30	2	2	2.1	0.00	10	60323.7	1.8	0.00	10	46538.6	1.2	0.00	10	16444.3
30	3	2	2.3	0.00	10	89480.6	3.1	0.00	10	135324.0	1.2	0.00	10	21683.1
30	3	3	2.5	0.00	10	100707.6	6.4	0.00	10	369452.3	1.5	0.00	10	28060.9
30	4	3	2.9	0.00	10	109318.6	22.7	0.00	10	1598527.0	1.6	0.00	10	31962.5
30	4	4	2.7	0.01	10	116011.3	58.9	0.00	10	4380303.9	1.5	0.01	10	29143.4
40	1	0	1.4	0.00	10	24138.6	0.9	0.00	10	8201.9	0.9	0.00	10	8201.9
40	1	1	2.3	0.00	10	76430.6	1.5	0.00	10	24842.0	1.5	0.00	10	24842.0
40	2	1	3.2	0.00	10	129606.6	2.0	0.00	10	53521.0	1.8	0.00	10	38364.4
40	2	2	3.6	0.00	10	154797.8	3.6	0.00	10	149599.3	2.4	0.00	10	49700.8
40	3	2	4.9	0.00	10	246961.9	6.6	0.00	10	339644.1	2.6	0.00	10	65869.2
40	3	3	5.1	0.00	10	271427.7	16.6	0.00	10	1101009.1	2.6	0.00	10	76066.0
40	4	3	6.7	0.00	10	414776.8	76.7	0.00	10	5214900.0	3.3	0.00	10	136812.1
40	4	4	6.7	0.00	10	413669.0	65.6	-	3	4424920.7	3.2	0.00	10	114682.3
50	1	0	1.9	0.00	10	41065.4	1.3	0.00	10	16210.3	1.3	0.00	10	16210.3
50	1	1	3.8	0.00	10	162075.2	2.2	0.00	10	54086.5	2.1	0.00	10	54086.5
50	2	1	5.9	0.00	10	282484.9	3.4	0.00	10	87342.1	3.0	0.00	10	69370.0
50	2	2	6.7	0.00	10	375376.8	7.0	0.00	10	321383.4	3.6	0.00	10	106554.6
50	3	2	9.3	0.00	10	569715.8	11.9	0.00	10	689948.8	4.1	0.00	10	147909.5
50	3	3	14.3	0.00	10	884531.9	52.4	0.00	10	3205071.0	5.7	0.00	10	219954.1
50	4	3	11.5	0.00	10	688935.7	105.2	0.00	10	5677371.8	4.6	0.00	10	165987.3
50	4	4	15.7	0.00	10	979020.7	-	-	0	-	7.1	0.00	10	269748.2
60	1	0	3.5	0.00	10	92746.4	2.1	0.00	10	39025.8	2.1	0.00	10	39025.8
60	1	1	6.8	0.00	10	303507.7	3.7	0.00	10	99055.5	3.9	0.00	10	99055.5
60	2	1	12.1	0.00	10	639618.3	6.7	0.00	10	268277.6	5.7	0.00	10	182760.0
60	2	2	14.9	0.00	10	813309.9	12.8	0.00	10	618998.1	5.9	0.00	10	225110.9
60	3	2	22.6	0.00	10	1315720.9	51.9	0.00	10	2662909.5	9.4	0.00	10	375667.0
60	3	3	27.3	0.00	10	1565877.7	145.3	0.00	10	7962911.6	10.8	0.00	10	470686.7
60	4	3	28.5	0.00	10	1626548.9	196.7	0.00	7	10459171.9	10.5	0.00	10	426588.0
60	4	4	48.3	0.00	10	2605507.6	-	-	0	-	17.8	0.00	10	760553.7

Table 2.6 Results for the heuristics

Chapter 3

Stochastic Vehicle Routing Problem with time windows and correlated travel times

3.1 Introduction

The research presented in this chapter is the result of my collaboration with Prof. Dr. Christoph Buchheim and Jonas Pruente, merged in [Bomboi et al. 2019]. The aim of this research was to devise a new approach to the VRP with time windows (VRPTW) assuming stochastic travel times. Uncertainty is taken into account in several ways: firstly, the cost of a route depends on its expected travel times and is hence a random variable itself. Secondly, if a driver arrives too early at a costumer's location, the resulting waiting time has to be taken into account. The presented model allows to calculate different costs for waiting and driving. Thirdly, and most importantly, a route may turn out to be infeasible under certain realizations of the travel times, due to the risk of missing some time window.

Unlike many other approaches presented in the literature, we explicitly deal with dependent travel times in our approach, assuming a joint normal distribution. In practice, neighboring streets often have highly correlated travel times, so that taking into account the dependency of travel times is important in order to obtain feasible solutions. This is also confirmed by the computational study below in the chapter: considering such dependencies improves the precision of the solutions significantly. The importance of correlations between travel times has also been underlined in [Park et al. 1999].

The presented approach can be used in any context where single routes are considered separately. This is the case, e.g., in the well-known Set Partitioning model presented in the introduction of the thesis in Chapter 1 and in Section 2.4.3 for solving the drayage problem of Chapter 2 , where the variables correspond to potential routes that are either enumerated in a first phase or are generated on the fly in a column generation approach. Also many heuristic approaches produce potential routes that have to be checked individually for feasibility. In particular, this allows in principle to use an arbitrary rule for deciding feasibility of a route and to determine its cost.

The solution method proposed in this chapter is based on the chance constraint approach, where a route is accepted if the risk of a failure stays below a given threshold. Two distinct application scenarios are considered. In the first scenario, the route is considered infeasible if for any of the customers on this route, the probability of missing her time window is too high. A (single) chance constraint formulation for this scenario is presented. In the second scenario, missing a customer's time window may render the entire route infeasible. The latter approach applies to variants of the VRP where pickups and deliveries are performed by the same vehicles, so that missing a customer might imply that some following customers cannot be served any more, e.g., due to lack of space in the vehicle. In this situation, we are interested in the (joint) probability that at least one of the time windows is missed. For this setting, a joint chance constraint formulation is presented. Moreover, for obtaining a more accurate stochastic model, we propose to consider truncated probability distributions in this case.

In both cases, an idea of Ehmke et al. [Ehmke et al. 2015] is used and extended. It addresses the fact that, due to possible waiting times, the arrival times at subsequent customers are not normally distributed any more. It is proposed in [Ehmke et al. 2015] to approximate these travel times, which can be defined as maxima between two normally distributed variables, again by normally distributed travel times, with the same expected values and variances. In this context, we also refer the reader to [Clark 1961], [Nadarajah et al. 2008] and [Sinha et al. 2007]. This approximation to the correlated case is extended in this chapter by also considering and updating the covariances between travel times. Additionally, a situation where travel times change over the day is considered, meaning that the time needed to travel an arc depends on when the arc is traversed.

It turns out that these extensions, though being computationally more expensive, lead to much more precise assessments of the feasibility of a route in realistic situations. In fact, our experiments on a set of realistic instances show that ignoring covariances leads to the creation of routes that are actually infeasible, when validated by sampling. The number of infeasible routes is decreased significantly when including covariance information in the presented algorithm. When travel times vary over the day, the difference between our approach taking this into account and an approach based on average travel times, is even larger, with many infeasible routes produced in the latter approach and, at the same time, objective values being better for our new algorithm.

3.1.1 Literature Review

Other works which use the chance constraint approach for the time window constraints are [Ehmke et al. 2015] with the assumption of independent travel times and [Li et al. 2010] with independent travel and service times, others set restrictions on the probability that a vehicle's capacity is exceeded [Dinh et al. 2018] and on the length of the travel time [Nahum et al. 2009].

In [Li et al. 2010], travel and service times are random variables following a normal distribution. In particular, together with the chance constraint approach, they propose to formulate their SVRPTW as a two-stage stochastic programming model with recourse. This consists in the first stage to determine the route scheduling before the stochastic travel and service times are known and in the second stage, once the two variables are realized, to take recourse actions to induce a penalty on the objective function. Unlike the chance constrained problem, stochastic programming model with recourse takes into account the possibility of route failure and the related correction costs. In [Dinh et al. 2018], the goals are to compute lower bounds on the minimum number of vehicles required to serve a subset of customers and to present a pricing for the branch-and-cut-and-price (BCP) approach for the chance constrained VRP problem. They show an improved relaxed pricing for independent normal demands and an extension to distributionally robust chance constraint. They show an interesting comparison between the chance constraint VRP (CCVRP) and the recourse models in the case of independent normal distribution. The recourse they assume is returning to the depot whenever a vehicle's capacity is exceeded. From the comparison, they conclude that "the CCVRP model tends to yield solutions that are high quality for the recourse model, whereas the reverse is not true. In addition, the CCVRP model is not dependent on a particular assumption of the recourse taken, and can be solved also when customer demands are not independent." [Dinh et al. 2018]. [Nahum et al. 2009] address the Stochastic Time-Dependent VRP (STDVRP). Their STDVRP algorithm is based on the savings algorithm of [Clarke and Wright 1964], designed for solving the deterministic CVRP. For transforming the stochastic and time-dependent data to deterministic data [Nahum et al. 2009] use the average value (average time for each time period and probability intervals), the best value (minimal time for all time periods, regardless of the probability) and the worst value (maximal time for all time periods, regardless of the probability) filters. In this way a candidate list of different deterministic estimators can be built for calculating routes, that are then analyzed by simulation. "Based on our findings for stochastic time-dependent vehicle-routing problems, the results of the STDVRP are similar to the results of the saving algorithm for CVRP" [Nahum et al. 2009].

Some reviews of SVRP literature can be found in [Gendrau et al. 1996 a], in the detailed [Oyola et al. 2018] and [Oyola et al. 2017], and [Bastian et al. 1992] for the case of uncertain, independent and identically distributed customer demands.

Articles dealing with uncertain travel times are [Ehmke et al. 2015], [Nahum et al. 2009], then [Tas et al. 2013], [Tas et al. 2014 - 1], [Tas et al. 2014 - 2], and [Woensel et al. 2003]. Among these, in particular [Nahum et al. 2009] and [Tas et al. 2014 - 1] study the time dependent case, in which travel times are stochastic and vary during the day.

Uncertain demands are considered in [Dinh et al. 2018] and [Golden et al. 1979] and [Guo et al. 2004], and [Novoa et al. 2006].

An SVRP with simultaneous pickup and delivery under uncertain demands and travel times is dealt with by [Hou et al. 2010].

For works based on stochastic travel and service times, it can be seen [Li et al. 2010], [Miranda et al. 2016], and [Zhang et al. 2013].

For a VRP with stochastic service times we refer to [Errico et al. 2016].

Several papers assume soft time windows, namely [Guo et al. 2004], [Tas et al. 2013], [Tas et al. 2014 - 1], [Tas et al. 2014 - 2] and [Zhang et al. 2013].

Concerning the assumption of independence and dependence of the uncertain variables, [Ehmke et al. 2015] assume independent travel times. Independent travel and service times are assumed by [Li et al. 2010] and [Miranda et al. 2016]. [Dinh et al. 2018] consider correlated demands and [Golden et al. 1979] assume that demands follow a Poisson distribution and show extensions with the Binomial, Negative-Binomial and Gamma distributions for solving the case of independent demand and assume multivariate normally distributed demands in case of correlation.

With respect to solution methods, the majority of publications opt for heuristics and metaheuristics, [Guo et al. 2004] use a genetic based algorithm, [Woensel et al. 2003] an Ant Colony Optimization heuristic, [Dinh et al. 2018] adapt and extend the Clarke and Wright's heuristic [Clarke and Wright 1964] to obtain primal solutions to the chance-constrained VRP and investigate a Dantzig-Wolfe formulation, [Golden et al. 1979] and [Lambert et al. 1993] also use a heuristic procedure based on [Clarke and Wright 1964]. For the computational experiments, [Ehmke et al. 2015] embed the feasibility check and the estimation of arrival and start-service times into a tabu search algorithm. Other papers which use the tabu search are [Li et al. 2010], [Tas et al. 2013] and [Zhang et al. 2013]. Recourse methods are used by [Errico et al. 2016], [Li et al. 2010], [Novoa et al. 2006] and [Zhang et al. 2013].

[Ehmke et al. 2015] case. Concerning the literature, particular attention is directed to the paper of [Ehmke et al. 2015], whose study can be considered at the basis of the presented chapter. They deal with a SVRPTW assuming that the time to travel from one customer to another is normally distributed. Their approach is based on the (single) chance constraints method, which leads to accept a set of routes that visits the set of customers if the probability of arriving at each customer by the latest time that the service can begin at that customer is greater or equal to a fixed threshold. Another assumption they make is that if a vehicle arrives at a customer before the beginning of the time window at that customer, then it must wait until the starting of the time window to begin the service. One of the main assumptions of their study is the independence of the travel cost variables. Their interpretation of the correlation between arc travel times is that it generally decreases over time; “that is, although the travel time of two arcs may be correlated at any given time, the correlation of the travel time of one arc at the current time with the travel time of another arc at some future time (i.e. after intervening travel and service time) will be less pronounced” [Ehmke et al. 2015]. In their study, the time-dependency of travel times is not considered. For their computational experiments, they embedded the feasibility check used in route construction as well as the estimation of arrival and start-service times in a tabu search algorithm.

3.1.2 Contribution and Outline

In the following, a new approach for modeling and solving two variants of the VRPTW subject to stochastic travel times is presented. The main contributions of this chapter are:

- an investigation of the importance of considering correlations between travel times by solving real instances;
- an approach for solving the single chance constrained routing problem with correlations, not based on sampling;
- an investigation of the importance of considering travel times varying over the day by solving instances created with real data;
- an approach for solving single chance constrained routing problems with time dependencies, not based on sampling;
- the description of an algorithm considering correlations and time dependencies at the same time for solving single chance constrained routing problems;
- an approach for solving the joint chance constrained routing problem with and without correlations, not based on sampling;

- an estimation of the waiting times of the vehicles at the customer locations, that can be used in penalty based approaches.

Section 3.2 presents some notation for the problems in consideration. Section 3.3 deals with the single chance constrained problem. In Section 3.4, an approximation of the joint chance constrained problem is discussed. In Appendix B.1, some stochastic formulas used in the proposed algorithms are listed.

3.2 Preliminaries and Notation

The general aim of this chapter is solving Vehicle Routing Problems with Time Windows subject to uncertain travel times. Extending the terminology of [Toth and Vigo 2002] for the deterministic version of the problem, we will refer to this class of stochastic problems as SVRPTW. Since characteristics like capacities and demands are not in the scope of this work, vehicles with infinite capacity are assumed. Every customer is visited exactly once by exactly one vehicle and all vehicle routes start and end at a single depot.

More formally, we assume that a finite set C of nodes is given, where $0 \in C$ corresponds to the *depot* and the remaining nodes in $C \setminus \{0\}$ correspond to *customers* to be served. Moreover, it is assumed that each pair of nodes $i, j \in C$ is connected by a directed arc $(i, j) \in E$, we thus deal with a complete graph $G = (C, E)$ throughout the chapter. A *route* r in G is given by an ordered list of distinct costumers $c_{r_1}, \ldots, c_{r_k}$, the set of its arcs is denoted by $E_r := \{(0, c_{r_1}), (c_{r_1}, c_{r_2}), \ldots, (c_{r_{k-1}}, c_{r_k}), (c_{r_k}, 0)\}$. Each costumer is assigned a deterministic *time window*. It will be convenient to index the time windows by arcs, thus for an arc $e = (i, j)$ we will denote by a_e (b_e) the *earliest (latest) arrival time* at costumer j, so that the time window is defined by $[a_e, b_e]$.

All travel times are uncertain and thus modelled as random variables. More precisely, we make the assumption that the vector $X \in \mathbb{R}^E$, which defines the travel time X_e for each arc e, is jointly normally distributed with means $\mu \in \mathbb{R}^E_+$ and covariance matrix $\Sigma \in \mathbb{R}^{E \times E}$, i.e., $X \sim \mathscr{N}(\mu, \Sigma)$; we will denote the entries of Σ by $\sigma_{e,f}$ for $e, f \in E$. For simplicity, we assume that Σ is positive definite.

In particular, the travel time of each single arc e is again normally distributed with $X_e \sim \mathscr{N}(\mu_e, \sigma_e^2)$, where we set $\sigma_e = \sqrt{\sigma_{e,e}}$. This also implies that the travel time of a route, given as the sum of travel times of the contained arcs, is normally distributed – however, this is only true as long as no time windows are considered. Finally, we will use $\varepsilon \in (0, 1)$ throughout the chapter to denote the threshold for the risk of a failure, i.e., we will accept a route if the probability that it is infeasible, with respect to the uncertain travel times, is at most ε. For the reader's convenience, this notation is summarized in Table 3.1.

Table 3.1 Basic notation defining our instances

0	depot node
C	set of customer nodes including depot
E	set of all arcs
E_r	set of arcs contained in a route r
a_e	earliest arrival time at end node of arc e
b_e	latest arrival time at end node of arc e
X_e	random variable describing the travel time of arc e
μ_e	expected travel time of arc e
$\sigma_{e,f}$	covariance between travel times of e and f
σ_e	standard deviation of travel time of arc e
ε	threshold for feasibility of a route

In general terms, we are going to discuss the following two questions with respect to a *fixed* route r:

(a) is route r feasible with a high enough probability?

(b) if so, what is the expected cost of route r?

Again the Set Partitioning approach to the VRP is an important example for the class of approaches that can be reduced to the tasks (a) and (b).

In the following sections, we will concentrate on the above questions (a) and (b) in two different application scenarios. In the first one, a route becomes infeasible if we miss the time window of any of the customers with probability larger than $1-\varepsilon$; see Section 3.3. In the second scenario, a route is infeasible if the probability of missing at least one costumer exceeds $1-\varepsilon$; see Section 3.4. Even if the difference between the two scenarios seems very subtle, the second approach is much more challenging from a mathematical (and complexity-theoretic) point of view, as it requires to deal with joint chance constraints. In fact, we can only deal with the latter case in an approximate way. In both cases, the expected costs have to take possible waiting times into account.

3.3 SVRPTW – Single Chance Constraints

In this section, it is assumed that all demands are deliveries, deterministic and known before the optimization process, and not split. A typical application for this scenario are deliveries from a post office.

The drayage problem of Chapter 2 can be also seen as an example in the case all customers are either importers or exporters requiring a stay-with service.

An important characteristic of this kind of problem is that if a customer in a route happens not to be served for some reason, the next customer in this route can still be served by the same vehicle. This is because the service failure at one customer does not undermine the service at the next customer. Therefore, in the chance constraint approach to be developed, a risk level of service failure is fixed for each customer independently. In other words, the feasibility of a route is determined considering the union of the single chance constraints in the route. In the following, we describe our approach in mathematical and algorithmic terms and show experimental results concerning solution quality and running times. The issue of correlations is first addressed in Section 3.3.1, then time-dependent travel times are considered in Section 3.3.2.

3.3.1 Including correlations

The first aim is to extend the approach of Ehmke et al. [Ehmke et al. 2015] in order to deal with correlated travel times. As it will be shown in our experiments in Section 3.3.1, taking correlations into account leads to a much more precise assessment of the feasibility of a route.

Algorithm

Assume we are given a route r with arc set $E_r = \{e_1, \dots, e_k, e_{k+1}\}$ (in this order). As already mentioned, the chance constraint consists in placing a restriction on the probability that a given customer time window is missed. For the first customer, the constraint is easily modeled as $P(X_{e_1} > b_{e_1}) \leq \varepsilon$, which is equivalent to $b_{e_1} \geq \mu_{e_1} + \Phi^{-1}(1-\varepsilon)\sigma_{e_1}$ where Φ denotes the cumulative distribution function of the standard normal distribution. However, when arriving before a_{e_1}, the driver has to wait. This may lead to other costs than driving. More importantly, the potential waiting time will influence the arrival times at the following costumers in the route. In particular, the distributions of the arrival times at the subsequent customers in the route have to be re-modeled.

As already discussed by Ehmke et al. [Ehmke et al. 2015], the adapted arrival times at a customer c_i do not follow a normal distribution any more. Anyway, they propose to approximate the resulting distributions by normal distributions again, iteratively at every customer, and show experimentally that the resulting error is negligable. More precisely, the idea is to compute means and variances of the distributions of every arrival time and to replace the random arrival times by the corresponding normal distributions.

In this section, we follow the same idea, relaxing however the assumption of independent travel times and instead taking into account the information of the correlations between routes and arcs. This is more complicated to do because we now also have to calculate the covariances between routes and arcs. Formally, we replace the vector containing the arrival time at the current node and the travel times of the subsequent arcs in the route by a jointly normally distributed vector having the same means and covariances. Clearly, this generalization leads to a higher running time due to the quadratic input in terms of the covariances, but we think it is worth to follow this approach for having a more realistic formulation and better solutions, as it will be shown in detail in the following Section 3.3.1.

Our approach is described in Algorithm 3. After initialization in Lines 1–6, it loops through the costumers of the given route r. It first updates the distribution information of the arrival time at the next customer (exp_k and var_k) as well as its covariance with all arcs f coming later in the route ($cov_{k,f}$); see Lines 8–14. Next, the route is discarded in case the chance constraint is violated (Line 15–17). Finally, the distribution information is updated once again due to a possible waiting at the current customer (Lines 18–22). All formulas needed for updating the distributions are derived in Appendix B.1.

Compared to the method proposed in [Ehmke et al. 2015], Lines 3–5, 10–13, 20–22, and 23 are new. All but the latter concern the calculation of the correlations between arcs and routes that [Ehmke et al. 2015] do not consider because of the assumption of independent travel times. Line 23 calculates the expected waiting time W of the vehicle at every customer. Together with the expected driving time D, it can be used to calculate the expected cost of route r, using any function in the two values D and W.

A first comparison between the method involving dependent travel times here proposed with Algorithm 3 and the method with the assumption of independent travel times of [Ehmke et al. 2015] can be already made. By the introduction of covariances it is possible to estimate better the variance of the arrival time at every customer and, consequently, of the travel time of the route $\tilde{X}_k$ at every step k. Without considering the covariances between arcs and routes, the variance is underestimated by [Ehmke et al. 2015] in case of positive correlation (see Line 10). For $\varepsilon < 0.5$ (and particularly for small ε), a smaller value of the variance leads to accepting as feasible a higher number of routes. A deeper analysis on the comparison between the solution methods is given in the following section.

Experimental Results

The instances used for the experiments are based on real traffic data for the surroundings of the port of Duisburg, the biggest hinterland hub in Europe. The port itself is chosen as depot and 19 nearby locations are picked as customer positions. The expected values and covariances of the travel times were calculated by a sample of data taken on 25 consecutive days at 3 pm from Google Maps [Google Maps]. If necessary, we added a value of 10^{-4}

Algorithm 3 Feasibility Check with Single Chance Constraint, taking correlations into account

Input: route $r = (0, r_1, \ldots, r_t, 0)$

Output: decision if r is feasible with high probability; expected driving time D; expected waiting time W

$k = 0$
$exp_k = 0$, $var_k = 0$
for $e \in E_r$ **do**
    $cov_{k,e} = 0$
end for
$W = 0$
for $e \in E_r \setminus (r_t, 0)$ **do**
    $exp_k \mathrel{+}= \mu_e$
    $var_k \mathrel{+}= \sigma_e^2$
    $var_k \mathrel{+}= 2cov_{k,e}$
    for $f \in E_r$ after e **do**
        $cov_{k,f} \mathrel{+}= \sigma_{e,f}$
    end for
    assume $\tilde{X}_k \sim \mathcal{N}(exp_k, var_k)$
    if $P(\tilde{X}_k > b_e) > \varepsilon$ **then**
        discard route r
    end if
    $exp_{k+1} = E[max\{\tilde{X}_k, a_e\}]$ ▷ Formula (B.1)
    $var_{k+1} = Var[max\{\tilde{X}_k, a_e\}]$ ▷ Formula (B.2)
    for $f \in E_r$ after e **do**
        $cov_{k+1,f} = Cov[max\{\tilde{X}_k, a_e\}, X_f]$ ▷ Formula (B.3)
    end for
    $W \mathrel{+}= E[max\{a_e - \tilde{X}_k, 0\}]$ ▷ Formula (B.1)
    $k \mathrel{+}= 1$
end for
$D = exp_k + \mu_{(r_t,0)}$
accept route r and return D and W

to all diagonal entries of the resulting covariance matrix in order to guarantee positive definiteness and to avoid numerical problems. Note that, if any, this has the effect of making the covariances slightly less relevant with respect to the variances.

Each arc of the graph corresponds to the shortest path between two customers or the port and a customer at the time point the data is taken. Therefore, an arc does not necessarily refer to the same path in all of the samples. This seems more realistic because a driver would always choose the shortest path at a given time. The considered network of customers and arcs is directed with an asymmetric matrix of the costs that satisfies the triangle inequality.

The arcs lengths observed in our samples ranged between 5 and 167 minutes, with an average of 60 minutes. In the resulting instance, the average correlation coefficient at a current time is 0.64 between two adjacent arcs and 0.60 between two non-adjacent arcs. The maximum correlation coefficient is 1.00 in both cases. This confirms our assumption that it is important to take the covariances into account in real life instances.

We created 10 different instances that only differ from each other in the time windows. The time windows of each instance were computed randomly in the following way: for each customer, the lower bound a_e of the time window is chosen uniformly at random as an integer between 0 and 7. The length of all time windows is 1 hour. If the time windows do not allow a feasible solution for the whole VRP for any of the algorithms considered, the time windows are recomputed randomly until they do.

In the following, we compare our new Algorithm 3 to the same algorithm using zero covariances (which essentially agrees with the algorithm of Ehmke et al. [Ehmke et al. 2015]), by solving the VRP problem on all 10 instances with each $\varepsilon \in \{0.01, 0.05, 0.1\}$; see Table 3.2. We then evaluate the solutions of the algorithms by sampling with 100,000 samples using the covariances, which enables us to calculate the "real" objective value and count how many of the chosen routes are actually infeasible. The objective function is $D + \frac{1}{2}W$, i.e., waiting is half as expensive as driving. For every sample with covariances, we calculated a vector of standard gaussians z with dimension $|E|$, multiplied it with a matrix L calculated by Cholesky decomposition of Σ, and added the vector of expected values. For solving the exact VRP we used the Set Partitioning formulation and solved it with CPLEX 12.6.3.0. All algorithms are implemented in Java version 1.8.0_191 and run on an Intel(R) Xeon(R) CPU E5-2640 0 with 2.5 GHz.

Table 3.2 consists of three main columns. In the first column, the value of ε is specified. The second column describes the results for the algorithm with covariances and the third for the algorithm without covariances. The second and the third column are divided into three subcolumns. The first presents the average cpu time in seconds, the second the total number of "optimal" solutions containing at least one infeasible route, and the third the objective of

Table 3.2 Comparison with Ehmke et al. [Ehmke et al. 2015] for Single Chance Constraint

ε	Algorithm 3 w/ covariances			Algorithm 3 w/o covariances		
	cpu time	# bad	obj	cpu time	# bad	obj
0.01	85.6	1	1.00	68.4	4	1.00
0.05	102.9	0	1.00	80.3	0	1.00
0.10	111.8	0	1.00	86.6	1	1.00

the method divided by the objective of the algorithm with covariances in the cases in which both methods have produced feasible solutions. In other cases a comparison would be unfair because the algorithm with more infeasible routes clearly has an advantage in terms of the objective value.

We can see that our Algorithm 3 outperforms the algorithm of [Ehmke et al. 2015] in terms of feasibility in 13% of the settings. In 4 instances it had less infeasible routes in the solution than the algorithm of [Ehmke et al. 2015] and there was only one instance in which both algorithms were infeasible. For all other instances, both algorithms returned the same solution and therefore the objective is the same in all feasible settings. The algorithm of [Ehmke et al. 2015] needs slightly less running time (between 77% and 80%), which is not surprising because it has to perform less calculations for every route. On the other hand, it computes more feasible routes and therefore the gap is not significant. We can conclude that in our opinion the advantages of the algorithm considering covariances in terms of feasibility clearly outweigh the slighly higher running time.

3.3.2 Including time dependency

We next address time dependency, i.e., we now allow that the traveling time of every arc e in the network varies depending on the time of the day. More precisely, we assume that the expected value and the variance of the travel time needed for an arc are functions of the point in time when the arc is entered. In practice, the traffic situation and hence the travel times strongly depend on the time of the day.

The difficulty here is that the time in which an arc is entered is itself a random variable, so that we have to deal with normally distributed random variables having expectations and variances that are implicitly defined by random variables again. An important modelling issue is how to define the dependency, i.e., which type of functions to allow. We decided to use a piecewise constant model, as it keeps the definition of instances easy and at the same time allows to efficiently update expected values and variances in our algorithm. Alternative approaches could use piecewise linear models or polynomials, splines, or even trigonometric approximations.

Algorithm

We assume to have information on travel times for a fixed set $t_0, ..., t_\Omega$ of time points during the day. We produce two piecewise constant functions describing the distribution at time t for arc e by the expected value $\mu_e(t)$ and the variance $\sigma_e^2(t)$, given the values $\mu_e(t_i)$ and $\sigma_e^2(t_i)$ for $i = 0, \ldots, \Omega$, as $\mu_e(t) := \mu_e(t_i)$ if $t \in [t_i, t_{i+1}]$ and analogously for $\sigma_e(t)$ (setting $t_{\Omega+1} = \infty$).

In our algorithm, we do not know the exact time when arc e is entered, it is given by a normal distribution. Hence, we have to consider t a random variable. Given its distribution function F_t, we can obtain the expected parameters for the distribution of travel time for e as

$$E[\mu_e(t)] = \sum_{i=0}^{\Omega} \mu_e(t_i) P\big(t \in (t_i, t_{i+1}]\big) = \sum_{i=0}^{\Omega} \mu_e(t_i)\big(F_t(t_{i+1}) - F_t(t_i)\big)$$

and analogously for $\sigma_e^2(t)$. This formula is used in Algorithm 4, Lines 6–7. The remaining parts of Algorithm 4 are analogous to Algorithm 3 except that no correlations are taken into account.

Experimental Results

Because we do not have hourly data for more than one day, we artificially extended the instances described above by using data of one day from the same streets hourly taken from 8 am to 5 pm. For every edge and for the value of every hour we computed the quotient to the value of 3 pm. These quotients were multiplied with the expected value of the edge to generate the expected value for that edge in that time. The variances remained unchanged, which is a disadvantage to our algorithm with time dependencies because in reality the variances also tend to be higher in times of the day with more traffic, and our algorithm could exploit this information while the other algorithm cannot.

Results are shown in Table 3.3. We compare our new Algorithm 4 with Algorithm 3 described above, however without using correlation information, i.e., with the approach proposed by Ehmke et al. [Ehmke et al. 2015]. The results show that in 5 out of 30 settings the algorithm without time dependencies produced infeasible routes, whereas the algorithm taking them into account always returned feasible solutions. The feasibility was again checked using sampling with 100,000 samples. Also in terms of solution value the algorithm without time dependency is less efficient and returns solutions with a value between 1 and 3 percent higher. On the other hand, it uses only 12 to 14 percent of the running time compared

Algorithm 4 Feasibility Check with Single Chance Constraint, taking time dependency into account (but no correlations)

Input: route $r = (0, r_1, \dots, r_t, 0)$
Output: decision if r is feasible with high probability; expected driving time D; expected waiting time W

```
k = 0
exp_k = 0, var_k = 0
W = 0
for e ∈ E_r \ (r_t, 0) do
    assume S ~ 𝒩(exp_k, var_k)
    exp_k += E[μ_e(S)]
    var_k += E[σ_e^2(S)]
    assume X̃_k ~ 𝒩(exp_k, var_k)
    if P(X̃_k > b_e) > ε then
        discard route r
    end if
    exp_{k+1} = E[max{X̃_k, a_e}]                ▷ Formula (B.1)
    var_{k+1} = Var[max{X̃_k, a_e}]              ▷ Formula (B.2)
    W += E[max{a_e − X̃_k, 0}]                   ▷ Formula (B.1)
    k += 1
end for
D = exp_k + μ_(r_t,0)
accept route r and return D and W
```

Table 3.3 Comparison with Ehmke et al. [Ehmke et al. 2015] for Single Chance Constraint with time dependencies (without covariances)

ε	Algorithm 4 w/ time dependencies			Algorithm 4 w/o time dependencies		
	cpu time	# bad	obj	cpu time	# bad	obj
0.01	346.3	0	1.00	46.4	0	1.01
0.05	416.3	0	1.00	57.1	2	1.03
0.10	462.7	0	1.00	62.2	3	1.03

to the new algorithm. In summary, we suggest to use the algorithm with time dependencies because it outperforms the other algorithm in terms of feasibility and objective value at the cost of a reasonable increase in running time.

3.3.3 Combining correlations and time dependency

In the preceding sections, we have explained how to take correlations and time dependency into account separately. It is also possible to combine both in one algorithm. For this, every time we need to know a covariance $\sigma_{e,f}(t_e,t_f)$ between two edges e and f, we need to compute an expected covariance of a two-dimensional piecewise constant function over two jointly normally distributed random variables t_e and t_f (the starting times of the two edges). As before, the starting time of an edge is just the random variable describing the length of the route up to that time. Therefore, we also need the covariances between subroutes. However, we only need to consider subroutes that start from the depot, so that the number of covariances to be computed remains quadratic in the route length. These covariances can be computed analogously to the covariance between the entire route and a single edge. All necessary data can be computed in the moment when it is required. When using piecewise constant distributions, as above, we need the joint distribution function of two normally distributed random variables to calculate the expected covariance. Computing this is numerically challenging but possible. It is also possible to sample the expected covariance with the given expected values and covariance matrix for the two starting points. This would lead to a hybrid approach using a combination of sampling and a deterministic algorithm.

We would like to emphasize that a pure sampling approach is very time-consuming when considering correlations and time dependencies. In both approaches discussed so far, the same samples could be used for all routes, and the covariance matrix Σ was fixed (for every time in the time dependent case). In particular, the Cholesky decomposition, needed to sample travel times in the dependent case, had to be computed only once. When considering the time-dependent case, we need the knowledge about the order of customers in the route already during the process of sampling. Therefore, we need to consider every route separately and cannot create one sample used for all routes.

Even worse, since the expected covariance $\sigma_{e,f}(t_e, t_f)$ depends on the starting times t_e and t_f now, the Cholesky decomposition cannot be computed in the preprocessing phase any more, since the matrix Σ now depends on t_e and t_f. This computation is further complicated when taking waiting into account.

3.4 SVRPTW with Backhauls and Linehauls – Joint Chance Constraints

So far, we have assumed that feasibility of a route is determined by a high enough probability to reach each customer within its time window. An implicit assumption in our model was that the failure of serving a customer does not influence the feasibility of the rest of the route directly. We now consider the case where failing to serve a customer may directly imply the failure of serving one of the remaining costumers on the route, motivated by problems where customer demands can be either deliveries or pickups. We will refer to this problem as a Stochastic VRPTW with Backhauls (pickup) and Linehauls (deliveries), SVRPTW-BL. The drayage problem in Chapter 2 is an example of VRP with pickup and deliveries with data not subject to uncertainty. Note however that our problem differs from the classical VRP with Backhauls [Toth and Vigo 2002], because no precedence constraints on the deliveries are assumed, and it differs from the VRP with Pickup and Delivery [Toth and Vigo 2002], in that the latter assumes a delivery and a pickup for each customer.

Our investigation of this VRP variant is motivated by problems in intermodal freight transport, for example freight transportation in intermodal containers using trucks, which involves the distribution of loaded and empty containers between an intermodal facility or depot and fixed export and import customers like the drayage problem in Chapter 2. In these applications, a large number of different customer request types and container constraints may arise [Bomboi et al. 2018]. In particular, if a customer in a route is not served for some reason, the feasibility of the rest of the route is not guaranteed anymore, e.g., a failed pickup of an empty container leads to the failure in loading the freight at the next customer or a failed delivery leads to a lack of space in the truck for a subsequent pickup.

We thus have to guarantee a service level for the whole route and not for each single customer, that is, we need to consider a joint chance constraints in the place of single chance constraints. In other words, we have to deal with the risk of infeasibility of the whole route rather that considering the risk of failing to serve any of the customers in the route independently. For simplicity, we assume here that failing to serve any customer within its time window makes the entire route infeasible. In the remainder of this section, an algorithm for this case, using joint chance constraints in an approximative way, is provided and computational results are presented. In principle, by a combination with the approach presented in the preceding section, our approach could be extended to the case where not all failures render the rest of the route infeasible. Besides having to deal with joint chance constraints now, another complication is that, after passing a customer, distributions have to be truncated, as we are only interested in travel times under the assumption that the customers visited so far have been served in time.

3.4.1 Algorithm

We now present an algorithm to address the SVRPTW-BL problem described above; see Algorithm 5. It takes correlations into account, but does not deal with time dependency. With respect to Algorithm 3, two major changes arise. Firstly, instead of checking a chance constraint for every customer, we have to check one chance constraint for the whole route. Unfortunately, dealing with joint chance constraints is hard, and there is no compact formula modeling the joint risk, therefore it is often approximated in the literature by calculating the risk of every single event (here a failure of one customer), summing it up (using ε_{total} in Algorithm 5), and comparing it to the maximum allowed risk for the joint chance constraint ε. We follow this approach; the corresponding changes in Algorithm 5 concern Lines 19-22.

When Algorithm 5 ends with a feasible route, i.e. when $\varepsilon_{total} \leq \varepsilon$, the value ε_{total} represents the probability of the route failure. This information could also be used in a bicriteria-style approach: instead of discarding all routes with $\varepsilon_{total} > \varepsilon$, one could sort the routes by their risk and solve the optimization problem for different risk levels ε without having to recompute the feasible routes. E.g., in a column generation approach, increasing ε would then just lead to the addition of more columns.

The second change concerns the truncation discussed above: for calculating the expected value and variance for later arrival times in the route, only the scenarios being feasible so far should be considered. E.g., if a driver misses the time window of the first customer and thus cannot serve it, he cannot continue to the second customer, and therefore the arrival time in this scenario does not influence the arrival time at the second customer, and so on. To model this, we use a one sided truncated normal distribution of the upper tail (using $\hat{\mu}$ and $\hat{\sigma}$ to denote expectations and covariances after truncation). After checking the chance constraint, the truncation for the expected value and for the variance are performed in Line 23 and 24. As we consider joint normal distributions, the truncation of one variable also affects the covariances between two other variables, so the whole submatrix of Σ corresponding to

Table 3.4 Comparison for Joint Chance Constraint

ε	Algorithm 5 w/ covariances & truncations			Algorithm 5 w/o covariances			Algorithm 5 w/o truncations		
	cpu time	# bad	obj	cpu time	# bad	obj	cpu time	# bad	obj
0.01	427.4	0	1.00	77.7	2	1.00	64.6	0	1.00
0.05	531.6	1	1.00	94.7	2	1.00	79.2	1	1.00
0.10	586.3	0	1.00	101.8	1	1.00	86.9	0	1.00

the current route has to be updated; see Lines 25–28. Note that the truncated distributions are again replaced by normal distributions. The formulas used for computing the truncated distributions can be found in Appendix B.1.

3.4.2 Experimental Results

In the experimental evaluation, we create instances in exactly the same way as described in Section 3.3.1, except that different sets of time windows may now be discarded due to infeasibility. Table 3.4 shows the results of a comparison between Algorithm 5 considering correlations and truncations, the variant of it not considering correlations, and another variant not performing truncations (but considering correlations). The algorithm considering correlations and truncations returned in only one of thirty settings an infeasible route. Ignoring the truncations did not lead to more infeasible settings, but decreased the running times by around 85 percent. Ignoring the correlations, on the other hand, led to five times more infeasible settings. In fact, every sixth setting resulted in an infeasible route in this approach. At the same time, it did not even run faster than the algorithm ignoring truncations, because it allows more feasible routes, which outweighs the additional running time per route. As a conclusion, the recommended algorithm for solving instances of SVRPTW-BL considers correlations but ignores truncations.

Algorithm 5 Feasibility Check for Joint Chance Constraint for a Route r

Input: route $r = (0, r_1, \dots, r_t, 0)$
Output: decision if r is feasible with high probability; expected driving time D; expected waiting time W

$k = 0$
$exp_k = 0$, $var_k = 0$
for $e \in E_r \setminus (r_t, 0)$ **do**
    $cov_{k,e} = 0$
end for
for $e, f \in E_r$ with $e \neq f$ **do**
    $\hat{\mu}_e = \mu_e$
    $\hat{\sigma}_{e,f} = \sigma_{e,f}$
end for
$W = 0$, $\varepsilon_{total} = 0$
for $e \in E_r$ **do**
    $exp_k \mathrel{+}= \hat{\mu}_e$
    $var_k \mathrel{+}= \hat{\sigma}_e^2$
    $var_k \mathrel{+}= 2cov_{k,e}$
    for $f \in E_r$ after e **do**
        $cov_{k,f} \mathrel{+}= \hat{\sigma}_{e,f}$
    end for
    assume $\tilde{X}_k \sim \mathcal{N}(exp_k, var_k)$
    $\varepsilon_{total} \mathrel{+}= P(\tilde{X}_k > b_e)$
    if $\varepsilon_{total} > \varepsilon$ **then**
        discard route r
    end if
    $exp_k = E[\tilde{X}_k \mid \tilde{X}_k < b_e]$ ▷ Formula (B.5)
    $var_k = Var[\tilde{X}_k \mid \tilde{X}_k < b_e]$ ▷ Formula (B.7)
    for $f, g \in E_r$ after e with $f \neq g$ **do**
        $\hat{\mu}_f = E[X_f \mid \tilde{X}_k < b_e]$ using $cov_{k,f}$ ▷ Formula (B.5)
        $\hat{\sigma}_{f,g} = Cov[X_f, X_g \mid \tilde{X}_k < b_e]$ using $cov_{k,f}$ and $cov_{k,g}$ ▷ Formula (B.6)
    end for
    assume $\tilde{X}_k \sim \mathcal{N}(exp_k, var_k)$
    $exp_{k+1} = E[max\{\tilde{X}_k, a_e\}]$ ▷ Formula (B.1)
    $var_{k+1} = Var[max\{\tilde{X}_k, a_e\}]$ ▷ Formula (B.2)
    for $f \in E_r$ after e **do**
        $cov_{k+1,f} = Cov[max\{\tilde{X}_k, a_e\}, \hat{X}_f]$ ▷ Formula (B.3)
    end for
    $W \mathrel{+}= E[max\{a_e - \tilde{X}_k, 0\}]$ ▷ Formula (B.1)
    $k \mathrel{+}= 1$
end for
$D = exp_k + \mu_{(r_t,0)}$
accept route r and return D and W

Chapter 4

Comparison of the deterministic and stochastic methods

The aim of this chapter is to show why it is a profitable investment considering the uncertainty of the travel times in the routing problems of this study. In this part, time dependency is not taken into account. Considering the algorithms used in Chapter 3 (Algorithm 3 and Algorithm 5) and to the obtained solutions (Table 3.2 and Table 3.4), we will see how choosing a deterministic algorithm (an algorithm which does not manage a possible uncertainty in the data) leads to the infeasibility of several routes in the optimal solution.

In the experimental evaluation, the instances used are the same created for obtaining the results shown in Section 3.3.1 and Section 3.4.2 (see Table 3.2 and Table 3.4). Table 4.1 compares Algorithm 3 for checking the feasibility with single chance constraints (scc) and taking correlations into account to the deterministic algorithm using expected values as deterministic data, by solving the VRP problem on all 10 instances with each $\varepsilon \in \{0.01, 0.05, 0.1\}$; see Table 4.1. We then evaluate the solutions of the algorithms by sampling with 100,000 samples considering correlated travel times, which enables us to calculate the "real" objective value and count how many of the chosen routes by the algorithms are actually infeasible. For solving the exact VRP, the Set Partitioning formulation is used and it is solved with CPLEX 12.6.3.0. All algorithms are implemented in Java version 1.8.0_191 on an Intel(R) Xeon(R) CPU E5-2640 0 with 2.5 GHz.

Table 4.1 consists of three main columns. In the first column, the value of ε is specified. The second column describes the results for the algorithm with covariances and the third for the deterministic algorithm. As in Chapter 3, the second and the third columns are divided into subcolumns. The column "cpu time" presents the average cpu time in seconds, the column "# bad" the total number of "optimal" solutions containing at least one infeasible route, the column "# infeas" the number of infeasible routes contained in the "optimal"

Table 4.1 Comparison between deterministic and scc stochastic solution with covariances

ε	Alg. 3 w/ covariances				Deterministic			
	cpu time	# bad	# infeas	obj	cpu time	# bad	# infeas	obj
0.01	85.6	1	1	1.00	0.9	9	22	1.00
0.05	102.9	0	0	1.00	0.9	9	14	1.00
0.10	111.8	0	0	1.00	0.8	9	12	0.99

Table 4.2 Comparison between deterministic and jcc stochastic solution with covariances and without truncations

ε	Alg. 5 w/o truncations				Deterministic			
	cpu time	# bad	# infeas	obj	cpu time	# bad	# infeas	obj
0.01	64.6	0	0	1.00	0.7	10	17	-
0.05	79.2	1	1	1.00	0.8	9	12	1.00
0.10	86.9	0	0	1.00	0.7	9	11	1.00

solutions in the ten given instances, and the column “obj” the objective of the method divided by the objective of the algorithm with covariances in the cases in which both methods have produced feasible solutions. In other cases a comparison would be unfair because the algorithm with more infeasible routes clearly has an advantage in terms of the objective value. In the results of Table 4.1, we can see that in terms of cpu time, the deterministic algorithm is clearly faster than Algorithm 3. However, as regards the quality of the solution in terms of the number of feasible routes in the optimal solution, we can see that in 90% of the instances the deterministic algorithm created at least one infeasible route. In the instance in which the optimal solution has all the routes feasible, the objective value was nearly the same (slightly better with the deterministic algorithm).

Table 4.2 has the same structure of Table 4.1 and it shows the results of a comparison between the variant of Algorithm 5 not performing truncations (considering correlations) and the deterministic algorithm. In this case, the subcolumn “obj” presents the objective of the method divided by the objective of Algorithm 5 with covariances and without truncations in the cases in which both methods have produced feasible solutions. The first row of Table 4.2 does not show the value obj because the deterministic algorithm produced infeasible solutions in all ten instances for $\varepsilon = 0.01$. We can see that the deterministic algorithm is faster than Algorithm 5, but again regarding the number of infeasible routes in the solution, in more than in 90% of the instances the deterministic algorithm created at least one infeasible route.

Seen the experimental evaluation above, we can say that the algorithms that take uncertainty into account are more efficient than the deterministic one. Considering also that the cpu time used by these algorithm is reasonably small for the instances considered, we can conclude that, in the context of daily decision making in routing problem (for example drayage starting in a port or shipment from a post office, etc..) it is the best to choose the algorithm that can manage uncertainty.

Chapter 5

Contributions

In this thesis new routing problems have been studied in the context of 0-1 integer programming optimization.

In the first part of this thesis, in Chapter 2, a drayage problem has been studied. The distribution of containers between customers and ports is not a new issue but in Chapter 2 is faced a drayage problem with a number of new characteristics that are relevant for practical applications. Although trucks can carry up to two 20 ft ordinary containers, no paper has focused on both ordinary and special containers, which are requested to be carried one at a time. Moreover, most of the research concerns stay-with or drop & pick services, but limited attention has been devoted to their integration, as customers can typically choose between both service types. The set of feasible routes has been explicitly enumerated in a short time and a Set Partitioning formulation has been used to determine the routes serving each customer request and minimizing routing costs. The presented method for doing this is very flexible and can be used for all problem statements in which the feasibility of a route can be decided fast.

Since the derived number of feasible routes may be too large for a Mixed-Integer programming solver, two heuristics are proposed to select subsets of promising routes and solve restricted Set Partitioning models. To produce the two heuristics, an analysis on the "parameters" of the problem that effect the number on the feasible routes has been useful. It showed in fact that, not the number of customers involved, but some kind of requests, is the main term in influencing the number of routes. The heuristics can be tuned: one could cut few routes in order to obtain higher quality solutions in longer running times, or cut many routes in order to determine lower quality solutions in a shorter running time or set possible tradeoffs. In this way, the "restricted" Set Partitioning formulation has been solved with the subsets of routes given by the heuristics or by their combination.

From the experimentation, it can be seen that the proposed heuristics are always able to determine optimal and near-optimal solutions in a very short time, achieving thus the target set at the beginning of the thesis. Moreover, the heuristic solutions are reached much faster than the exact solution method calculated with the Set Partitioning formulation in which all the feasible routes are used. In support of the decisional study of this drayage case, a formula to estimate the overall number of feasible routes precisely in short time is presented in Appendix A. It can be adapted to various problem statements with different constraints. The number of feasible routes highly correlates with the running time of the algorithm and therefore with this information the user can decide quickly which method to apply.

In the same Chapter 2, a node-arc formulation is also presented and a pricing formulation using an adapted labeling algorithm for a column generation approach has been developed. The column generation has been embedded in a branch-and-bound. From the results it can be seen that with the increase of some types of customer requests, the branch-and-price becomes better than the Set Partitioning. The node-arc formulation instead does not solve the majority of the instances and it does not represent a good option for solving the drayage problem under study.

In the second main part of the thesis, in Chapter 3, we devised feasibility check algorithms for solving the Stochastic Vehicle Routing Problem with time-windows and with correlated and time-dependent travel times, using either single or joint chance constraints depending on whether missing a customers time window makes the entire route infeasible or not. In particular, the contributions achieved in this context started paying attention on the importance of considering correlations between travel times in stochastic routing problems. This has been done by solving realistic instances. The first achieved objective has been giving an approach for solving the single chance constrained routing problems with correlations. An example of problem which can be solved in this way is the classical delivery service from a post office. This class of problems has been also solved with time dependencies after an investigation on the importance of considering travel times varying over the day by solving instances created with real data. After that, an algorithm has been described considering correlations and time dependencies at the same time for single chance constrained routing problems.

Motivated by the study of the drayage problem of the first part of the thesis, the attention has been moved then to the case where failing to serve a customer may directly imply the failure of serving one of the remaining costumers on the route. For this class of problem, an approach has been given for solving the joint chance constrained routing problem with and without correlations, as proposed in the objectives of the thesis. The presented algorithms can be embedded into any algorithm for solving the Vehicle Routing Problem with hard time windows and waiting times as a feasibility check of a given route. Due to the fact that the algorithms also compute expected waiting times, they can be easily adapted to variants of the problem with soft time windows and penalties. The experimental results show that the algorithms assess the feasibility of a given route with a reasonable precision, in particular when correlations are taken into account. Still, due to the approximation errors we make, a pure sampling approach can give more accurate results. However, for a large number of samples, the latter approach is time consuming.

Finally, in Chapter 4, to answer the question asked in the introduction whether it is a profitable investment considering the uncertainty of the travel times in the routing problems of this study, a comparison between the deterministic and stochastic methods is briefly presented. Seen the experimental evaluation, we can say that the algorithms that take uncertainty into account are more efficient in terms of feasibility of the solution than the deterministic one. Concluding then, considering the cpu times used by these algorithms, it is the best to choose the algorithm that can manage uncertainty.

Future extension A possible future extension for the drayage problem of Chapter 2 is studying the problem assuming time windows for each customer. The introduction of the time windows decreases the number of feasible routes and consequently makes the Set Partitioning formulation easier to solve. On the other hand, it makes a possible node-arc model slower to solve. Further analysis on new methods for this last point can be carried out.

Concerning the study of Chapter 3, considering the complementary strengths of the deterministic and the sampling approach, one could also investigate combinations of both approaches. E.g., one could check for a given route how likely it is to become infeasible, and if the result is very close to the value of acceptance, do a second check performing sampling with a high number of samples – possibly only for routes that were accepted, but very close to be not accepted, because adding an infeasible route is in general more problematic than missing a feasible one. Another hybrid approach would be to use sampling with a high number of samples just as a callback inside the optimization process: first compute an optimal solution using our approach, then check the feasibility of all routes used in this solution by sampling, and if some route turns out to be infeasible, remove it and re-optimize. The advantage of this approach is that the sampling has to be performed for a very small set of routes.

Bibliography

[Ackooij et al. 2011] van Ackooij W., Zorgati R., Henrion R., Möller A., *Chance Constrained Programming and Its Applications to Energy Management*, Stochastic Optimization - Seeing the Optimal for the Uncertain, Ioannis Dritsas, IntechOpen, DOI: 10.5772/15438, 2011.

[Bastian et al. 1992] Bastian C., Rinnooy Kan A.H.G., *The stochastic vehicle routing problem revisited*, European Journal of Operational Research, vol. 56, issue 3, 407-412, 1992.

[Bomboi et al. 2018] Bomboi F., Pruente J., *A new drayage problem with different customer services and container requirements*, Technical Report, 2018.

[Bomboi et al. 2019] Bomboi F., Buchheim C., Pruente J., *Stochastic Vehicle Routing Problem with time windows and correlated travel times*, Technical Report, 2019.

[Braekers et al. 2013] Braekers K., Caris A., Janssens G. K., *Integrated planning of loaded and empty container movements*, OR Spectrum, vol. 35, issue 2, 457 - 478, 2013.

[Charnes et al. 1959] Charnes A., Cooper W. W., *Chance-Constrained Programming*, Management Science, vol. 6, issue 1, 73 - 79, 1959.

[Clark 1961] Clark C. E., *The Greatest of a Finite Set of Random Variables*, Operations Research, vol. 9, issue 2, 145 - 162, 1961.

[Clarke and Wright 1964] Clarke G., Wright, J. W., *Scheduling of Vehicles from a Central Depot to a Number of Delivery Points*, Operations Research, vol. 12, issue 4, 568-581, 1964.

[Deidda et al. 2008] Deidda L., Di Francesco M., Olivo A., Zuddas P., *Implementing the street-turn strategy by an optimization model*, Maritime Policy & Management, vol. 35, issue 5, 503 - 516, 2008.

[Desaulniers 2018] Desaulniers G., *Constrained vehicle routing and shortest path problems*, Column Generation School 2018, Paris, France, February 26 - March 2, 2018.

[Dinh et al. 2018] Dinh T., Fukasawa R., Luedtke J., *Exact algorithms for the chance-constrained vehicle routing problem*, Mathematical Programming, vol. 172, issue 1, 105 - 138, 2018.

[Dotoli et al. 2016] Dotoli M., Epicoco N., *A technique for the optimal management of containers' drayage at intermodal terminals*, 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 566 - 571, 2016.

[Ehmke et al. 2015] Ehmke J. F., Campbell A. M., Urban T. L., *Ensuring service levels in routing problems with time windows and stochastic travel times*, European Journal of Operational Research, vol. 240, issue 2, 539 - 550, 2015.

[Errico et al. 2016] Errico F., Desaulniers G., Gendreau M., Rei W., Rousseau L. M., *A priori optimization with recourse for the vehicle routing problem with hard time windows and stochastic service times*, European Journal of Operational Research, vol. 249, issue 1, 55 - 66, 2016.

[Escudero-Santana et al. 2015] Escudero Santana A., Cortes Achedad P.F., Munuzuri J., Aparicio Ruiz P., *A viral system to optimise the daily drayage problem*, International Journal of Bio-Inspired Computation, vol. 7, issue 3, 176-182, 2015.

[Funke et al. 2016] Funke J., Kopfer H., *A model for a multi-size inland container transportation problem*, Transportation Research Part E: Logistics and Transportation Review, vol. 89, 70 - 85, 2016.

[Gendrau et al. 1996 a] Gendreau M., Laporte G., Sèguin R., *Stochastic vehicle routing*, vol. 88, issue 1, 3-12, 1996.

[Ghezelsoflu et al. 2018] Ghezelsoflu A., Di Francesco M., Frangioni A., Zuddas P. *A Set-Covering formulation for a drayage problem with single and double container loads*, Journal of Industrial Engineering International, 2018.

[Golden et al. 1979] Golden B. L., Yee J. R., *A Framework For Probabilistic Vehicle Routing*, A I I E Transactions, vol. 11, issue 2, 109-112, 1979.

[Google Maps] Google Maps, `maps.google.com`.

[Guo et al. 2004] Guo Z. G., Mak K. L., *A heuristic algorithm for the stochastic vehicle routing problems with soft time windows*, Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753), vol. 2, 1449-1456, 2004.

[Hoff 2009] Hoff P. D., *A first course in Bayesian Statistical Methods*, Springer Texts in Statistics, 2009.

[Hou et al. 2010] Hou L., Zhou H., *Stochastic Vehicle Routing Problem with Uncertain Demand and Travel Time and Simultaneous Pickups and Deliveries*, 2010 Third International Joint Conference on Computational Science and Optimization(CSO), vol. 01, 32-35, 2010.

[Huber 2017] Huber M., *Data Modeling & Analysis Techniques*, Notes Probability & Statistics University of Texas at Arlington, http://ranger.uta.edu/ huber/cse5301/Notes/Probability.pdf, 2017.

[Ileri et al. 2006] Ileri Y., Bazaraa M., Gifford T., Nemhauser G., Sokol J., Wikum E., *An Optimization Approach for Planning Daily Drayage Operations*, Central European Journal of Operations Research, vol. 14, issue 2, 141 - 156, 2006.

[Irnich et al. 2005] Irnich S., Desaulniers G., *Shortest Path Problems with Resource Constraints*, In: Desaulniers G., Desrosiers J., Solomon M.M. (eds) Column Generation. Springer, Boston, MA, 2005.

[Jula 2006] Jula H., Chassiakos A., Ioannou P., *Port dynamic empty container reuse*, Transportation Research Part E: Logistics and Transportation Review, vol. 42, issue 1, 43 - 60, 2006.

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

[Lambert et al. 1993] Lambert V., Laporte G., Louveaux F., *Designing collection routes through bank branches*, Computers & Operations Research, vol. 20, issue 7, 783-791, 1993.

[Li et al. 2008] Li P., Arellano-Garcia H., Wozny G., *Chance constrained programming approach to process optimization under uncertainty*, Computers & Chemical Engineering, vol. 32, issue 1, 25 - 45, 2008.

[Li et al. 2010] Li X., Tian P., Leung S.C.H., *Vehicle routing problems with time windows and stochastic travel and service times: Models and algorithm*, International Journal of Production Economics, vol. 125, issue 1, 137-145, 2010.

[Miranda et al. 2016] Miranda D. M., Conceicao S. V., *The vehicle routing problem with hard time windows and stochastic travel and service time*, Expert Systems with Applications, vol. 64, 104-116, 2016.

[Nadarajah et al. 2008] Nadarajah S., Kotz S., *Exact Distribution of the Max/Min of Two Gaussian Random Variables*, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, vol. 16, issue 2, 210 - 212, 2008.

[Namboothiri et al. 2004] Namboothiri R., Erera A. L., *A set partitioning heuristic for local drayage routing under time-dependent port delay*, 2004 IEEE International Conference on Systems, Man and Cybernetics (IEEE Cat. No.04CH37583), vol. 4, 3921 - 3926, 2004.

[Namboothiri et al. 2008] Namboothiri R., Erera A. L., *Planning local container drayage operations given a port access appointment system*, Transportation Research Part E: Logistics and Transportation Review, vol. 44, issue 2,185 - 202, 2008.

[Nahum et al. 2009] Nahum O. E., Hadas Y., *Developing a model for the stochastic time-dependent vehicle-routing problem*, 2009 International Conference on Computers Industrial Engineering, 118-123, 2009.

[Nemhauser et al. 1988] Nemhauser G. L., Wolsey L. A., *Integer and Combinatorial Optimization*, Wiley Interscience series in discrete mathematics and optimization, 1988.

[Nossack et al. 2013] Nossack J., Pesch E., *A truck scheduling problem arising in intermodal container transportation*, European Journal of Operational Research, vol. 230, issue 3, 666 - 680, 2013.

[Novoa et al. 2006] Novoa C., Berger R., Linderoth J., Storer R., *A set-partitioning-based model for the stochastic vehicle routing problem*, technical report 06T-008, 2006.

[Oyola et al. 2017] Oyola J., Arntzen H., Woodruff D. L., *The stochastic vehicle routing problem, a literature review, Part II: solution methods*, EURO Journal on Transportation and Logistics, vol. 6, issue 4, 349 - 388, 2017.

[Oyola et al. 2018] Oyola J., Arntzen H., Woodruff D. L., *The stochastic vehicle routing problem, a literature review, part I: models*, EURO Journal on Transportation and Logistics, vol. 7, issue 3, 193 - 221, 2018.

[Park et al. 1999] Park D.,Laurence R. R., *Forecasting Freeway Link Travel Times with a Multilayer Feedforward Neural Network*, Computer-Aided Civil and Infrastructure Engineering, vol. 14, issue 5, 357-367, 1999.

[Reinhardt et al. 2016] Reinhardt L. B., Pisinger D., Spoorendonk S., Sigurd M. M., *Optimization of the drayage problem using exact methods*, INFOR: Information Systems and Operational Research, vol. 54, issue 1, 33 - 51, 2016.

[Shiri et al. 2016] Shiri S., Huynh N., *Optimization of drayage operations with time-window constraints*, International Journal of Production Economics, vol. 176, 7 - 20, 2016.

[Sinha et al. 2007] Sinha D., Zhou H., Shenoy N. V., *Advances in Computation of the Maximum of a Set of Gaussian Random Variables*, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 26, issue 8, 1522-1533, 2007.

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

[Srour et al. 2010] , Srour J., Mahr T., de Weerdt M., Zuidwijk R., *MIPLIB Truckload PDPTW Instances Derived from a Real-World Drayage Case*, ERIM report series research in management Erasmus Research Institute of Management, issue ERS-2010-036-LIS, 2010.

[STHDA] Statistical tools for high-throughput data analysis, web site, http://www.sthda.com/english/.

[Tas et al. 2013] Tas D., Dellaert N., van Woensel T., de Kok T., *Vehicle routing problem with stochastic travel times including soft time windows and service costs*, Computers & Operations Research, vol. 40, issue 1, 214-224, 2013.

[Tas et al. 2014 - 1] Tas D., Dellaert N., van Woensel T., de Koktitle T., *The time-dependent vehicle routing problem with soft time windows and stochastic travel times*, Transportation Research Part C: Emerging Technologies, vol. 48, 66-83, 2014.

[Tas et al. 2014 - 2] Tas D., Gendreau M., Dellaert N., Van Woensel T., De Kok A.G., *Vehicle routing with soft time windows and stochastic travel times: A column generation and branch-and-price solution approach*, European Journal of Operational Research, vol. 236, issue 3, 789 - 799, 2014.

[Toth and Vigo 2002] Toth P., Vigo D., *The Vehicle Routing Problem*, Siam, 2002.

[Vividovic et al. 2012] Vidovic M., Nikolic M., Popovic D., *Two mathematical formulations for the containers drayage problem with time windows*, Int. Journal of Business Science and Applied Management, vol. 7, issue 3, 2012.

[Woensel et al. 2003] Van Woensel T., Kerbache L., Peremans H., Vandaele N., *A Vehicle Routing Problem with Stochastic travel times*, In Fourth Aegean International Conference on Analysis of Manufacturing Systems location, Samos, 2003.

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

[Xue et al. 2015] Xue Z., Lin W.H., Miao L., Zhang C., *Local container drayage problem with tractor and trailer operating in separable mode*, Flexible Services and Manufacturing Journal, vol. 27, issue 2, 431 - 450, 2015.

[Zhang et al. 2013] Zhang J., Lam W. H. K., Chen B. Y., *A Stochastic Vehicle Routing Problem with Travel Time Uncertainty: Trade-Off Between Cost and Customer Service*, Networks and Spatial Economics, vol. 13, issue 4, 471-496, 2013.

Appendix A

Set of feasible routes

The possibility to solve the Set Partitioning problem by the algorithms of Mixed-Integer Programming solvers mostly depends on the number of feasible routes. Therefore, it is very useful to determine this number before solving any problem instance, in order to calibrate the heuristics and control the problem size. Integrated to the study of Chapter 2, a formula is derived to determine the number of feasible routes as a function of ten variables, each representing the number of each type of customer request. There are two different ways to derive this formula. In the first way only the customers on a route are regarded and in the second way also the initial state of the route is regarded. In other words, in the second case we can take into account the same sequence of visits more than one time due to the different initializations of the truck. Clearly, the first set of routes is a subset of the second. For example, if a DE request must be served in the route of sequence $\{Port, DE, Port\}$, this route is counted as one in the first way, in the second instead, at least two different routes can be performed with the same sequence: one with initial states of the truck slots Empty - Absent, and one starting with states Empty - Empty. In our study, the number of routes is computed in the second way because one does not have to check in the list of routes if a new route is already listed (according to that sequence). In addition, the number of redundant routes is very small (around 10% of the feasible routes). Generally, the computation of the number of routes by this formula is very fast.

Example of formula

The formula cannot be reported in Chapter 2 of this thesis for the sake of space. Nevertheless, it is shown hereafter in the specific case of up to 2 customers per route without limits on the number of PE or DE request:

$$
\begin{aligned}
|R| =& 3|DLO| + 3|DLSWO| + 3|DLO| \cdot |PE| + 3|DLO| \cdot |PLO| + |DLO|(|DLO| - 1) + \\
& |DLO| \cdot |DLSWO| + 3|DLSWO| \cdot |DE| + |DLSWO| \cdot |DLO| + 3|DLSWO| \cdot |PLSWO| + \\
& |DLSWO|(|DLSWO| - 1) + 4|PE| + 3|PLO| + \mathbf{4}|\mathbf{PE}| \cdot |\mathbf{DE}| + |PE| \cdot |DLO| + \\
& 3|PE| \cdot |PLSWO| + |PE| \cdot |DLSWO| + |PLO| \cdot |DLO| + |PLO| \cdot |DLSWO| + \\
& 2|DLO| \cdot |PLS| + |DLSWO| \cdot |PE| + |DLSWO| \cdot |PLO| + |DLSWO| \cdot |PLS| + \\
& 2|DLSWO| \cdot |PLSWS| + 4|DE| + 3|PLSWO| + \mathbf{4}|\mathbf{DE}| \cdot |\mathbf{PE}| + 3|DE| \cdot |PLO| + \\
& |DE| \cdot |DLO| + |DE| \cdot |DLSWO| + |DLO| \cdot |DE| + |DLO| \cdot |PLSWO| + \\
& |DLO| \cdot |PLSWS| + |PLSWO| \cdot |DLO| + |PLSWO| \cdot |DLSWO| + 2|PLS| + \\
& |PE|(|PE| - 1) + |PE| \cdot |PLO| + |PE| \cdot |PLS| + 2|PE| \cdot |PLSWS| + |PLO| \cdot |PE| + \\
& |PLO|(|PLO| - 1) + |PLS| \cdot |PE| + 2|PLSWS| + 2|DE| \cdot |PLS| + |PLO| \cdot |DE| + \\
& |PLO| \cdot |PLSWO| + |PLS| \cdot |DE| + |PLSWO| \cdot |PE| + |PLSWO| \cdot |PLO| + \\
& |PLSWS| \cdot |PE| + |DE|(|DE| - 1) + |DE| \cdot |PLSWO| + |DE| \cdot |PLSWS| + \\
& |PLSWO| \cdot |DE| + |PLSWO|(|PLSWO| - 1) + |PLSWS| \cdot |DE| + 2|DLS| + \\
& 2|DLSWS| + |PE| \cdot |DLS| + |PE| \cdot |DLSWS| + 2|DLS| \cdot |PE| + 2|DLS| \cdot |PLO| + \\
& 2|DLS| \cdot |PLS| + |DLSWS| \cdot |PE| + 2|DLSWS| \cdot |DE| + |DLSWS| \cdot |PLO| + \\
& |DLSWS| \cdot |PLS| + 2|DLSWS| \cdot |PLSWO| + 2|DLSWS| \cdot |PLSWS| + |DE| \cdot |DLS| + \\
& |DE| \cdot |DLSWS| + |DLS| \cdot |DE| + |DLS| \cdot |PLSWO| + |DLS| \cdot |PLSWS|
\end{aligned}
$$

This formula has 74 summands obtained by simple enumeration. The choice not to sum some similar summands (for examples the summands in bold) is due to the fact that these summands are derived from different routes. If the maximum number of customer requests per route is set to 8 without any limit on the number of *PE* or *DE* customers, 9068 summands are obtained. Constraints depending explicitly or implicitly on the customer coordinates due to the time cannot be modeled exactly in the formula.

Estimation

In our problem statement we have three types of constraints the formula above does not take into account: the 8 hours working day constraint and the constraints of the second heuristic. Considering these constraints, we want to find a good estimation of the number of routes in a short time. To deal with this problem, we calibrate the probability p_s that every summand s appears in the formula. The calibration is performed in a sample of k

routes by the computation of the ratio between the number of routes meeting all additional requirements and the number of routes computed by the formula. If k is larger, the estimation of p_s is better. In our test we set k to 10,000. If the value of the summand s is less than k, instead of simulating we just enumerate all the feasible routes. This makes the estimation better and decreases the running time. To solve this problem, we need to multiply every summand s in the formula with the probability p_s that a route made by the sequences of types represented by this summand is feasible. To make it clear we want to adapt the formula shown before for the first five summands to the new constraints:

$$\begin{aligned}|R| =& 3p_1|DLO| + 3p_2|DLSWO| + 3p_3|DLO| \cdot |PE| + 3p_4|DLO| \cdot |PLO| + \\ & p_5|DLO|(|DLO| - 1) + \dots\end{aligned}$$

This formula can be used for all kinds of problem statements in which it is possible to decide the feasibility of a route fast.

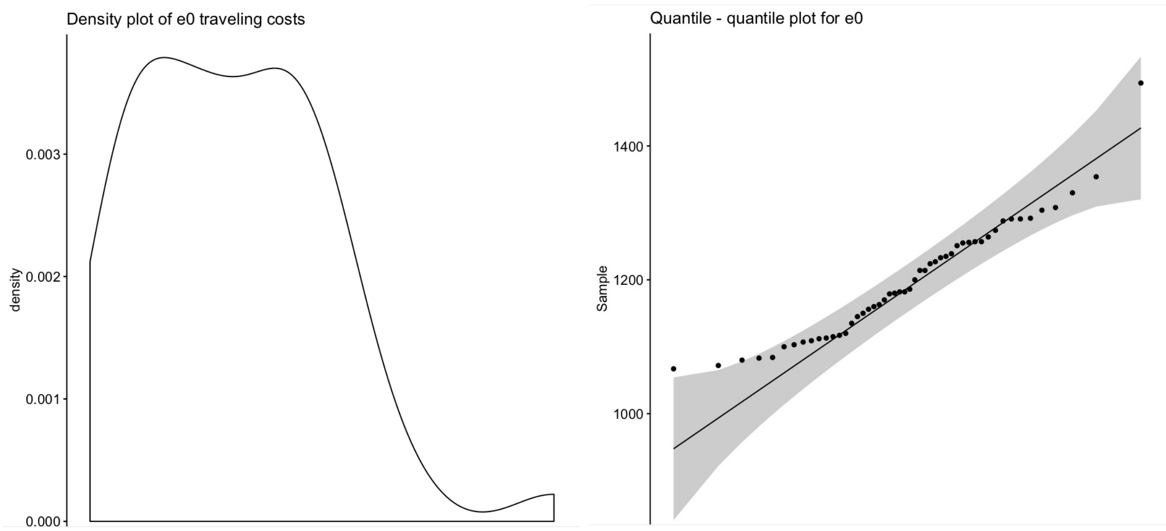
In Chapter 2 the formula has been useful for having an idea of the effective dimension of the instances. In fact, as it can be seen in details in the thesis, some kind of requests and not the number of customers involved mainly influence the number of routes. With the help of the formula and the experimental results of the methods presented in Chapter 2, a hypothetical user of these methods can choose the most suitable one according to the problem they have. The adapted formula is also suitable in the case time windows constraints are considered.

Appendix B

Details on statistics

On the use of the Normal Distribution Family The gaussian distribution or normal distribution developed into a standard of reference for many applications because it approximates very well several events and phenomena. Some typical easy examples of use are known for modeling heights and weight of adults, GRE scores of the applicants for a scholarship, family income, time that a student stays in a classroom, life-time of the TV sets, volume of liquid per bottle (considering random fluctuations in the automatic bottling machine), monthly food expenditure, etc. Anyway, it has to be mentioned that, even if this extremely important distribution arises in many real world applications, many others phenomena are not well modeled by a normal distribution. There are families of many other distributions that are useful to analyze events. Briefly e.g., "the continuous uniform distribution models the likelihood that a particular outcome will result from an experiment where every outcome value is equally likely, exponential distribution models the likelihood of an event happening for the first time at time x in a Poisson process (i.e. a process where events occur with the same likelihood at any point in time, independent of the time since the last occurrence). The normal distribution models the likelihood of results if the results are either distributed with a "Bell curve" or, alternatively, the result of the summation of a large number of random effects" [Huber 2017]. "Perhaps it is the most useful (or used) probability model for data analysis. There are several reasons for this, one being the central limit theorem, and another being that the normal model is a simple model with separate parameters for the population mean and variance - two quantities that are often of primary interest" [Hoff 2009].

Here we describe how to check the normality of the data to make sure that the assumption of the normality of the traveling costs are met or reasonable to assume. This can be done initially by visual inspection [STHDA]. In Fig. B.1 and Fig. B.2 is represented a first visual analysis regarding the first two edges of the network. In particular, the density plot (Fig. B.1(a) and Fig. B.2(a)) "provides a visual judgment about whether the distribution is bell



(a) Density of the first edge of the network (b) Q-Q plot of the first edge of the network

Fig. B.1 Visual inspection for the first edge of the network

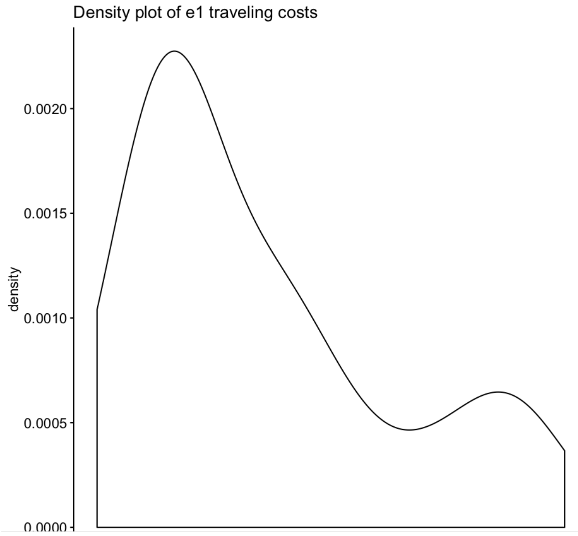
shaped" [STHDA]. Another visual method is the Q-Q plot (Fig. B.1(b) and Fig. B.2(b)) "or quantile-quantile plot, which draws the correlation between a given sample and the normal distribution. A 45-degree reference line is also plotted" [STHDA].

Since visual inspection is usually unreliable, normality test is in general the next step to introduce. "It is possible to use a significance test comparing the sample distribution to a normal one in order to ascertain whether data show or not a serious deviation from normality. Shapiro-Wilk's method is widely recommended for normality test. It is based on the correlation between the data and the corresponding normal scores" [STHDA]. The null hypothesis of these tests is that "sample distribution is normal". If the test is significant, the distribution is non-normal. In Listing B.1 "Distribution of the traveling costs", the R code for getting visual inspection and normality test for the traveling costs is shown. From the results obtained by Shapiro-test on all the edges, it is possible to see that the number of the p-values greater than 0.05 is just 11 out of 380.

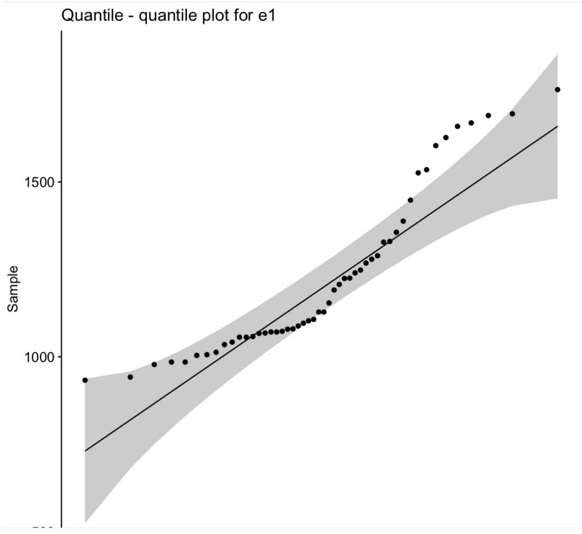
Maximum of two independent normal random variables [Clark 1961] was the first to investigate the maximum of a finite number of normal variables in the general case in which these variables do not have a common expected value, variances are unequal, or correlation exists. He approximated the first four moments. The maximum of two normals is approximated with another normal by matching the first two moments of their distribution. The results obtained for normal distributions are used with adequate accuracy in some cases involving non-normal distributions. It was shown that the moments of $\max\{\xi_1, \xi_2\}$

Listing B.1 Distribution of the traveling costs

```
# dplyr for data manipulation
install.packages("dplyr")
library("dplyr")
# Load other required R packages
 install.packages("ggpubr")
 library("ggpubr")

# Import data into R
traveling_costs <- read.delim("traveling_costs.txt")

# Assess the NORMALITY of the data in R

# Visual method
# 1. Visual method: the density plot.
 ggdensity(traveling_costs$e0, main = "Density␣plot␣of␣e0␣traveling␣costs",
          xlab = "e0␣traveling␣costs")

# 2. Visual method: Q-Q plot or quantile-quantile plot.
ggqqplot(traveling_costs$e0, title = "Quantile␣-␣quantile␣plot␣for␣e0")

# NORMALITY TEST
# Shapiro-Wilk's test.
# Null hypothesis of the test = "sample distribution is normal".
# If p-value > 0.05: H0 is accepted
# The R function shapiro.test() can be used to perform
# the Shapiro-Wilk test of normality for one variable (univariate):
shapiro.test(traveling_costs$e0)
# OUTPUT Shapiro-Wilk normality test
# data: traveling_costs$e0
# W = 0.94731, p-value = 0.02647

# The R function shapiro.test() can be used for one variable (univariate)
# Shapiro-test for all edges
lshap <- lapply(traveling_costs[,2:381], shapiro.test)
lres <- sapply(lshap, `[`, c("statistic","p.value"))
lres <- t(lres)
# Data export
write.table(lres, "lres_shapiro.txt", sep = ",")
```



(a) Density of the second edge of the network



(b) Q-Q plot of the second edge of the network

Fig. B.2 Visual inspection for the second edge of the network

with ξ_1 and ξ_2 not normally distributed are adequately approximated by the moments of $\max\{\eta_1, \eta_2\}$ with η_1 and η_2 normally distributed. Recently, [Sinha et al. 2007] quantified the approximation error when results obtained by [Clark 1961] are employed to compute the maximum of Gaussian random variables. The maximum of multiple normals is not a normal, and approximating its distribution with that of a normal, as [Clark 1961] made, introduces inaccuracies. For multiple normals, the max operation is performed a pair at a time and each of these pairwise operations introduces errors by approximating the resulting distribution with a normal [Sinha et al. 2007]. [Sinha et al. 2007] derived a closed expression for the true distribution of the maximum and they used it to quantify the approximation error. Furthermore, also the propagation of these approximation errors affect accuracy. They found out that the final loss in accuracy during the max of multiple Gaussians is dependent on the order in which pairwise max operations are performed and they proposed good orderings for pairwise max operations on a given set of Gaussians to reduce the loss in accuracy without significant increase in run times but for this thesis the order is always fixed. [Nadarajah et al. 2008] provided some of the known expressions for the probability density function, moment generating function, and the moments of maximum and minimum of correlated normal random variables.

Motivation of the use of the Normal distribution Having regards to the visual inspection, the significant test and the studies on the approximation with normal distributions of the previous paragraphs, we decided to model the travel costs as normally distributed random variables, as [Ehmke et al. 2015] did. In fact, already [Ehmke et al. 2015] demonstrated via simulation that the equations they derive for modeling the arrival times work well with instances that have travel times that are not normally distributed. In the results presented in the thesis it works good in practice.

Furthermore, even if the approximation used for the maximum of the two variables (normal or not with a normal distribution according to the extreme value theory) has objectionable accuracy, the error can always be quantified. And on the other hand, due to the extensive use of the normal distribution in practical problems, useful formulas are known, can be calculated more easily and can be exploited. Anyway, it is not excluded that more suitable distributions to the travel costs can be found and the methods presented in the thesis can be adapted to these.

B.1 Computation of Updated Moments

For the convenience of the reader, we explain in the following how to compute the expected values, variances and covariances of the random variables appearing in Algorithms 3 and 5 of Chapter 3. For this, we use the assumptions and notation of Section 3.2, in particular, we assume $X \sim \mathcal{N}(\mu, \Sigma)$ and thus $X_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$. For indices $i_1, \ldots, i_k$, we consider the joint density function of $X_{i_1}, \ldots, X_{i_k}$, given by

$$f_{X_{i_1},\ldots,X_{i_k}}(s_1, \ldots, s_k) := \frac{1}{\sqrt{(2\pi)^k \det(\bar{\Sigma})}} e^{-\frac{1}{2}(s-\bar{\mu})^\top \bar{\Sigma}^{-1}(s-\bar{\mu})} ,$$

where $\bar{\mu}$ denotes the vector with entries $\mu_{i_1}, \ldots, \mu_{i_k}$ and $\bar{\Sigma}$ the corresponding submatrix of Σ. We again assume here that Σ and hence $\bar{\Sigma}$ is positive definite.

Rectified Gaussian Distributions We first consider the random variable $\max\{c, X_i\}$ for a constant c. For the expected value, we obtain

$$\begin{aligned} E(\max\{c, X_i\}) &= \int_{-\infty}^{\infty} \max\{c, s\} f_{X_i}(s)\, ds \\ &= \int_{-\infty}^{c} c f_{X_i}(s)\, ds + \int_{c}^{\infty} s f_{X_i}(s)\, ds \\ &= cP(X_i \le c) + \mu_i P(X_i \ge c) + \sigma_{ii} f_{X_i}(c) \end{aligned} \tag{B.1}$$

To compute the variance

$$Var(\max\{c, X_i\}) = E(\max\{c, X_i\}^2) - E(\max\{c, X_i\})^2 \tag{B.2}$$

we can use (B.1) and

$$\begin{aligned}
E(\max\{c,X_i\}^2) &= \int_{-\infty}^{\infty} \max\{c,s\}^2 f_{X_i}(s)\,ds \\
&= \int_{-\infty}^{c} c^2 f_{X_i}(s)\,ds + \int_{c}^{\infty} s^2 f_{X_i}(s)\,ds \\
&= c^2 P(X_i \le c) + (\mu_i^2 + \sigma_{ii}) P(X_i \ge c) + \sigma_{ii}(c+\mu_i) f_{X_i}(c)
\end{aligned}$$

Finally, we have

$$\begin{aligned}
E(X_j \cdot \max\{c,X_i\}) &= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} t \max\{c,s\} f_{X_i,X_j}(s,t)\,ds\,dt \\
&= \int_{-\infty}^{\infty}\int_{-\infty}^{c} tc f_{X_i,X_j}(s,t)\,ds\,dt + \int_{-\infty}^{\infty}\int_{c}^{\infty} ts f_{X_i,X_j}(s,t)\,ds\,dt \\
&= c\mu_j P(X_i \le c) + (\mu_i\mu_j + \sigma_{ij}) P(X_i \ge c) + \mu_j \sigma_{ii} f_{X_i}(c)
\end{aligned}$$

and thus obtain

$$\begin{aligned}
Cov(X_i, \max\{c,X_j\}) &= E(X_i \cdot \max\{c,X_j\}) - E(X_i)E(\max\{c,X_j\}) \\
&= \sigma_{ij} P(X_j \ge c)
\end{aligned} \tag{B.3}$$

Truncated Gaussian Distributions We next develop formulas for the moments of the random vector X truncated by a condition $X_i < b$, where b is again a constant. For the expected values, we have

$$\begin{aligned}
E(X_j \mid X_i \le b) &= \frac{1}{P(X_i \le b)} \int_{-\infty}^{+\infty}\int_{-\infty}^{b} t f_{X_i,X_j}(s,t)\,ds\,dt \\
&= \mu_j - \sigma_{ij} \frac{f_{X_i}(b)}{P(X_i \le b)}
\end{aligned} \tag{B.4}$$

and, in particular,

$$E(X_i \mid X_i \le b) = \mu_i - \sigma_{ii} \frac{f_{X_i}(b)}{P(X_i \le b)} \tag{B.5}$$

For the covariances, we have

$$Cov(X_j, X_k \mid X_i \le b) = E(X_j X_k \mid X_i \le b) - E(X_j \mid X_i \le b) E(X_k \mid X_i \le b) \tag{B.6}$$

which can be computed using (B.4) and

$$\begin{aligned} E(X_jX_k \mid X_i \leq b) &= \frac{1}{P(X_i \leq b)} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{b} tu f_{X_i,X_j,X_k,X_i}(s,t,u)\, du\, ds\, dt \\ &= \mu_j\mu_k + \sigma_{jk} - \left(\frac{\sigma_{ij}\sigma_{ik}}{\sigma_{ii}}(b-\mu_i) + \sigma_{ik}\mu_j + \sigma_{ij}\mu_k \right) \frac{f_{X_i}(b)}{P(X_i \leq b)} \end{aligned}$$

As a special case, we obtain

$$Var(X_j \mid X_i \leq b) = Cov(X_j, X_j \mid X_i \leq b) \tag{B.7}$$

which also applies to the case $i = j$.