ON VEHICLE ROUTING WITH UNCERTAIN DEMANDS

by

Mahdi Noorizadegan

a thesis supervised by

Prof. Bo Chen and Dr. Nalan Gulpinar

Submitted to the University of Warwick

for the degree of

Doctor of Philosophy

Warwick Business School

November 2013

Contents

Acknowledgments

I would like to thank my supervisors, Bo Chen and Nalan Gulpinar, for their support, guidance and understanding. Special thanks to Xuan Vinh Doan, for his invaluable helps and guidance. I would also like to express my appreciation to my examiners, Juergen Brank (Warwick Business School) and Frits C.R. Spieksma (Faculty of Business and Economics K.U.Leuven), for their valuable comments and suggestions. Very special thanks also go to all faculty, staff and fellow students of Warwick Business School, for the wonderful moments and experiences I shared with them during my PhD.

Finally, I would like to express my heartfelt gratitude to my parents. Without their incredible support and encouragement, I would not have made it through my PhD so smoothly.

Abstract

In this research, we present a theoretical and computational framework for studying the vehicle routing problem with uncertain demands (VRPUD). We combine approaches in stochastic optimization and techniques in mixed integer programming to solve two main variants of the vehicle routing problem with uncertain demands.

We first present a polyhedral study for deterministic heterogenous vehicle routing problems (HVRP) to develop a relatively efficient formulation such that its corresponding counterpart with uncertainty is tractable via mixed integer programming. Having assumed customers' demand is uncertain, we apply three single-stage approaches within stochastic optimization to the HVRP with uncertain demands. The three-single stage approaches are chance constrained programming, Ben-Tal and Nemirovski, and Bertsimas and Sim robust optimization approaches. Then, we plug the corresponding formulation for each approach into a branch-and-cut method.

Moreover, we propose a new framework within the branch-and-price framework to formulate the capacitated vehicle routing problem (CVRP) with uncertain demands. In addition to the three single-stage approaches, we apply a two-stage stochastic approach to the capacitated vehicle routing problem with uncertain demands. Our proposed framework enables us to model different types of uncertainty while the complexity of the resulting problem remains the same.

Finally, we present extensive computational experiments for the deterministic HVRP, the HVRP with uncertain demands and the CVRP with uncertain demands. In the computational experiments we first investigate efficiency of several types of valid inequalities and lifting techniques for the deterministic HVRP. Then, using simulation and a scenario based technique we assess the performance, advantages and disadvantages of the aforementioned stochastic optimization approaches for the

HVRP with uncertain demands and the CVRP with uncertain demands. We show that among single-stage approaches of stochastic optimization, those with control parameters outperform those without control parameters in terms of total expected cost. Also, we show that the higher protection level does not necessarily result in better solutions as higher protection levels may impose unnecessary extra costs. Moreover, as our computational experiments suggest, the two-stage models for the CVRP dominate the single-stage approaches for all protection level scenarios.

Chapter 1

INTRODUCTION

The share of transportation in the total cost of products is estimated to be 10-20 percent ([77]). Moreover, Hesse and Rodrigueb [43] report that in the year 2000, transportation presented 5.9% of the US's Gross Domestic Product (GDP). In a more global level, transportation's contribution in the greenhouse gas is estimated to be 24% of the greenhouse gas produced in the European Union ([33] and [64]). Therefore, improving the transportation system is a quite important task in the individual, domestic and global levels.

Among problems defined within transportation, the routing problem is one of the most important and challenging ones so that it has attracted academics for many decades. The Traveling Salesman Problem (TSP) is one of the oldest routing problems studied in academia which goes back to 1800s. The TSP has been extended and more realistic assumptions such as capacity limitation for vehicles and demand uncertainty have been considered for the routing problems. These additions of course increase the difficulty of the routing problems. In this dissertation, we study two main variants of the routing problems where the demands are not known in advance. We will focus on the heterogeneous vehicle routing problems with uncertain demands and its special case namely the capacitated vehicle routing problems with uncertain demands.

1.1 Problem description

The Capacitated Vehicle Routing Problem (CVRP), with its many variants, is one of the most widely studied NP-Hard problems in combinatorial optimization due to its various practical applications and theoretical challenges. However, the CVRP with *uncertain demands* has received much less attention, in particular within exact methods. The aim of this research is to solve two variants of vehicle routing problems with uncertain demands to optimality. Let us start with the definition of the classical CVRP. The classical CVRP is defined on an arc weighted directed graph $G = (V, E)$ with the set of vertices being $V = \{0, ..., n\}$ and the routing costs being c_e, $e \in E$. It consists in serving a set of customers $V_c = \{1, \dots, n\}$ with known demand $q_i,\ i \in V_c$, using a fleet of vehicles with identical capacity Q. The vehicles are stationed at the same (unique) depot which is usually denoted by vertex 0 in the graph, i.e., $V = \{0\} \cup V_c$. Each vehicle takes exactly one route starting from the depot, visiting a subset of the customers and returning to the depot. The customer's demand cannot be split among different routes and the sum of demands in each route must not exceed the vehicle capacity Q. The solution of the CVRP is a minimum cost partition of the customers according to the vehicle routes. The heart of the CVRP's difficulty lies on the conditions on the route feasibility. A feasible route is defined as follows.

Definition 1. *A feasible route is a route which starts from the depot, visits each customer at most once and returns to the depot without violating the vehicle capacity limitation.*

These conditions put the CVRP and its variants among the most challenging problems in combinatorial optimization. In the deterministic context, different methods have been suggested to formulate these conditions. Later in this chapter we present two key formulations to model the above conditions. But let us first present a general mathematical formulation for the CVRP. The decision variable x_e is a binary variable which takes value one if edge $e = (i, j) \in E$ is used, and zero otherwise. Also, for a given vertex $i \in V$, $\delta(i)$ denotes the set of incoming and outgoing edges. A generic Integer Programming (IP) formulation for the described CVRP is:

$$\text{G-CVRP:} \quad \min \quad \sum_{e \in E} c_e x_e \tag{1.1}$$

$$\text{s.t.} \quad x(\delta(i)) = 2, \quad \forall i \in V_c, \tag{1.2}$$

$$x(\delta(i)) \leq m, \quad i = 0, \tag{1.3}$$

$$x_e \in X_R, \tag{1.4}$$

$$x_e \in \{0, 1\}, \tag{1.5}$$

where constraint (1.2) known as degree constraints guarantees every customer is visited exactly once. Constraint (1.3) makes sure that at most m routes are used. Set X_R represents the set of constraints that form feasible routes as defined. Several formulations have been suggested to present set X_R. Later in this chapter we present two main formulations for set X_R. The above model is defined for the deterministic CVRP, but as the unknown demands are embedded in set X_R, we can still use the G-CVRP for Vehicle Routing Problems with Uncertain Demands (VRPUD) by replacing X_R with $\bar{X}_R$ which is the set of feasible routes for the VRPUD. To characterize set $\bar{X}_R$ for the VRPUD, we study two types of uncertainty cases. In the first case, we assume that the demands are random variables with known probability distributions. In the second case, we assume that only partial data such as lower and upper bounds are available for each uncertain demand. In the VRPUD, It is a common assumption that the customer's demand is revealed upon the vehicle's arrival ([67] and [68]). In the presence of the demand uncertainty, the route feasibility conditions may be violated for a pre-planned route, i.e., the available vehicle capacity may not be sufficient to serve a customer as the customer's demand becomes known on the vehicle's arrival. To capture the demand uncertainty, we use three different approaches: *chance constraint programming*, *robust optimization* and *stochastic programming*. With respect to each approach, set $\bar{X}_R$ is formulated accordingly. Before going through technical issues and mathematical models, in the next section we describe three main strategic policies which are used to deal with the demand uncertainty in the vehicle routing problem.

1.2 Policies

Due to the capacity limitation of each vehicle and also due to the fact that the demand of a customer is usually revealed on the vehicle's arrival, a vehicle may fail to serve a customer on its pre-planned route. Three main policies have been proposed to deal with the demand uncertainty for the VRPUD based on the fact that routing and replenishment decisions are dynamic or static: *restock*, *reoptimization* and *a priori* approaches.

Restock policy In the restock policy, routes are static and replenishments are dynamic (proactive) i.e., a set of routes is fixed in advance but replenishment decisions are made after visiting each customer (before actually a failure occurs). The decision for a replenishment i.e., when to make a return trip to the depot can be made using a simple threshold or using states defined in Markov Decision Process based on unserved customer's demand, the vehicle's remaining capacity and the vehicle's current location ([80]). The advantages of the restock policy are as follows:

- it is easier for drivers to follow a fixed route every day,
- managerial processes are easier than the reoptimization policy (will be explained next) as one decision has to be made at each time,
- customers deal with the same drivers.

However, this policy is difficult to formulate and solve, and solutions may not be as good as solutions obtained by the reoptimization policy.

The reoptimization policy In the reoptimization policy which is also known as the real-time policy, both routing and replenishment decisions are dynamic (proactive), in the sense that they are decided according to the current state and are un-planned. Drivers in each stage decide to visit an unvisited customer or return to the depot for a replenishment. The stage's state is defined based on *the remaining capacity*, *unserved customers* and *the current location of the vehicle*. This policy is formulated by a finite horizon Markov Decision Process (MDP) and solved via dynamic programming (see [67] and [68]). Dynamic Programming (DP) provides a powerful framework for formulating complex problems which can be broken into simpler problems via formulating a sequential decision problems. But it suffers from two difficulties. The first difficulty is known as *the curse of dimensionally* which is due to the size of state space as it exponentially grows. The second difficulty of DP is that all aspects of a system (e.g., transition and value functions) are required to be known which is not always possible.

To overcome these two main disadvantages, first a decision making process/-function (as the core of DP) may be approximated, and second a sample of possible events which may happen should be randomly generated. The resulting approach is called Approximation Dynamic Programming (ADP) which provides near optimal solutions/policies. When the value functions are approximated, then the output of ADP can be used along with simulations to learn and improve the approximations if needed [65].

Advantage: Dror [30] notes that *"reoptimization is the most promising approach for solving [VRP with stochastic demand] exactly without narrowly restricting* *the policy space"*. Despite this advantage, there are some drawbacks for this policy as follows:

- this policy is difficult to formulate and solve,
- the managerial processes are difficult,
- customers may deal with different drivers,
- because of the large state space, not very large instances can be solved to optimality.

***A priori* policy** A policy is called *a priori* policy if routes are static and replenishment is reactive i.e., a set of routes planned in advance is executed and only when a failure occurs an action must be taken. This action can be simply to leave the rest of customers on the failed route unserved or the vehicle returns to the depot to reload (or empty the load) and then resumes the route from the failed customer. The possible actions will be discussed later in more detail. The output of this policy is a set of fixed routes that minimizes a specific measure of the total cost e.g., the expected cost. The advantages of this policy are:

- it is easier for drivers to follow a fixed route every day,
- managerial processes are easier,
- customers deal with the same drivers,
- this policy is easier to formulate and solve,

– larger instances can be solved to optimality.

The disadvantage of this policy is that solutions provided by this policy may not be as good as the two previous policies.

Due to advantages of the *a priori* policy we use this policy in this research. Hence, we focus on solution methods developed within the a priori policy and discuss its related topics in greater details as follows.

Two main approaches for modelling the demand uncertainty have been deployed to formulate and solve the VRPUD within this policy: *single-stage* and *two-stage* approaches. In the single-stage approach, the VRPUD is (normally) formulated as a Mixed-Integer Program (MIP) representing a specific situation of the system. The model can represent the worst-case situation or it can represent a situation with a tradeoff between route validity and the total cost. Chance Constraint Programming (CCP) and Robust Optimization (RO) are in particular two popular approaches within stochastic optimization which have been applied to formulate and solve the VRPUD. The key advantage of applying these approaches is that the deterministic equivalents of the VRPUD can remain tractable via mixed integer linear programming (MILP) depending on the initial model. But the downside of the single-stage models is that they do not consider any recourse action in the modelling phase and as a result any recourse cost. In practice, if a company wants to use these models, they have two options when they face a route failure. Firstly, their policy might be to serve all customers somehow, in other words, no lost sales are allowed. Secondly, they may prefer to leave the remaining customers on failed routes unserved in other words lost sales are allowed. In the first option, managers may decide to make a return trip to the depot for a replenishment or outsource serving the remaining customers on failed routes. In the second option, leaving the remaining customers unserved may impose an extra (penalty) cost as the lost sale cost.

In order to take the recourse cost into account, two-stage models are proposed. There are several types of recourse actions in the literature which will be discussed in Chapter 6 in detail. *Two-stage Stochastic Mixed Integer Programming* (2SMIP) provides a strong framework to formulate problems with uncertainty and mixed integer variables. In this framework, a problem is decomposed into two stages. The first stage consists of the master problem which is independent from the uncertain parameters while the second stage consists of the sub-problems. The sub-problems which are also known as recourse problems/actions usually correspond to possible scenarios of the uncertain parameters. Using the first-stage solution, each sub-problem is solved. Then, sets of *optimality cuts* and *feasibility cuts* are derived in respect to the first-stage variables and added to the master problem.

However, as a special case, the VRPUD can be formulated within the 2SMIP framework so that there is no need to set up the sub-problems as such. Instead, customized optimality cuts can be derived directly from the master problem's solutions when the demands follow known and specific distribution functions. The customized optimality cuts are used within the integer L-shaped method to provide a tighter approximation for the expected failure cost ([44] and [49]).

Traditionally, two-stage models have been applied to two-index formulations for the VRP. Recently, set-partitioning formulations of the VRP are also used to model and solve the SVRP ([22]). In this case, the cost of each route consists of the routing cost and the expected failure cost which are used to solve column generation sub-problems. In the next section, we explain these two popular formulations which have been used to formulate the VRPUD.

1.3 The VRP formulations

Studies on the VRPUD for the a priori policy have been mainly carried out on two formulations of the VRP: flow based and set-partitioning formulations. In this research we also base our models on these two formulations. Therefore, we present the basic deterministic formulations for the CVRP on which the stochastic formulations for the VRPUD will be built.

1.3.1 Flow formulation

Recall the G-CVRP where three types of constraints are defined. The first type (the degree/assignment constraints) makes sure that each customer is visited only once which implies each customer is assigned to a route. The second type guarantees that no more than m vehicles are used. The third type of constraints which is the most challenging one guarantees the route feasibility. This type of constraints has two implications. Firstly, the vehicle capacity limitation must not be violated and secondly, there must be no sub-tour. Dantzig et al. [26] and Miller et al. [55] suggest two different formulations to model the sub-tour elimination condition for the Traveling Salesman Problem (TSP). Since the CVRP is an extension of the TSP, the sub-tour eliminations formulations for the TSP are extended with some modifications for the CVRP.

DFJ formulation Dantzig et al's sub-tour elimination formulation (DFJ-SE) enumerates all possible tours and ensures that the total number of edges in each sub-set has to be less than the size of the set minus one. This constraint can be formulated as follows for the TSP.

$$x\left(E(S)\right) \leq |S| - 1,\ \forall S \subset V,\ |S| \geq 2 \tag{1.6}$$

where $E(S)$ is the set of edges whose both ends are in set S. Alternatively, the above constraint can be re-stated by

$$x\left(\delta(S)\right) \geq 2,\ \forall S \subset V,\ |S| \geq 2, \tag{1.7}$$

where $\delta(S)$ is the set of edges which have exactly one end in set S. Constraint (1.6) and (1.7) are facet defining constraints for the TSP. Note that when we refer to the TSP, we assume a traveling salesman problem defined on weighted directed graph $G = (V, E)$ where V is the set of the cities with unit demand and unlimited vehicle capacity. Constraint (1.7) can be adopted for the CVRP as follows:

$$x\left(\delta(S)\right) \geq 2k(S),\ \forall S \subset V,\ |S| \geq 2, \tag{1.8}$$

where $k(S)$ is the minimum number of vehicles required to serve the customers in S. Finding an optimal solution for $k(S)$ is the *bin packing problem* which by itself is an NP-hard problem ([54]). But it can be approximated by its lower bound $\frac{q(S)}{Q}$ where

$q(S) = \sum_{i \in S} q_i$. So, constraint (1.8) can be reformulated as follows:

$$x\left(\delta(S)\right) \geq \frac{2q(S)}{Q}, \ \forall S \subset V, \ |S| \geq 2, \tag{1.9}$$

Constraints (1.8) and (1.9) do not define facets of the CVRP. Therefore, it does not present the convex hull of the problem. However, the lower bounds provided by these constraints for the CVRP are very tight. The advantage of the DFJ-SE constraints is their good Linear Relaxation (LR) but at the cost of exponential number of constraints. Therefore, instead of adding them straight away in the initial formulation, they are added within a cutting plane based algorithm only when they are violated.

MTZ formulation Miller et al. [55] propose another formulation to model the sub-tour elimination constraints by introducing a new variable ($u_i \geq 0$) which denotes the load of the vehicle after visiting vertex i. The MTZ constraints eliminate sub-tours based on a contradiction for flow variables. Miller et al. suggest the following formulation for the TSP:

$$u_j - u_i + \mathrm{M}x_e \leq \mathrm{M} - 1, \ \forall e = (i, j), \ i, j \in V. \tag{1.10}$$

In the above constraint, M is a big number which is usually equal to $|V|$. The above constraint can be modified for the CVRP as follows:

$$u_j - u_i + Qx_e \leq Q - q_j \quad \forall e = (i, j),\ i, j \in V_c, \tag{1.11}$$

$$q_j \leq u_j. \tag{1.12}$$

The above set of inequalities is known to have a weak LP relaxation, hence they do not present the facet defining constraints. It will be shown how they can be improved and lifted. The advantage of the MTZ formulation is that these constraints are polynomial in size.

1.3.2 Set-partitioning formulation

In 1964, Balinski and Quandt [11] were the first to use the set-partitioning concept to formulate the CVRP. In set-partitioning formulations a column is a valid route which covers a set of customers and does not violate the vehicle capacity limitation. Since then, set-partitioning formulations of the CVRP have received a considerable attention. Until 2006, the most promising and successful method of formulating and solving the CVRP was the branch-and-cut algorithm (see [54]). In 2006, Fukasawa et al. [36] propose a set-partitioning based formulation which solves the CVRP efficiently. Later, Baldacci and Mingozzi [10] suggest a unified framework to solve different variants of the VRP. Here, we present a basic set-partitioning formulation for the CVRP.

Let $\mathcal{R}$ be the index set of all feasible routes charactrized by $q(\mathcal{V}(r)) = \sum_{i \in \mathcal{V}(r)} q_i \leq$

Q and $\mathcal{V}(r) = \{r_0 = 0, r_1, ..., r_{n_r}, r_{n_r+1} = 0\}$. Let $\mathcal{R}(i)$ be all routes that contain vertex $i \in V$ ($\mathcal{R}(i) = \{r : i \in \mathcal{V}(r)\}$). Also let f_r be the cost of route $r \in \mathcal{R}$. The CVRP can be formulated as follows:

$$
\begin{aligned}
(\mathrm{SP}): \quad \mathcal{Z}(\mathrm{P}) = \min \quad & \sum_{r \in \mathcal{R}} f_r z_r && (1.13)\\
\text{s.t.} \quad & \sum_{r \in \mathcal{R}} z_r \leq m, && (1.14)\\
& \sum_{r \in \mathcal{R}(i)} z_r \geq 1, \ \forall i \in V_c, && (1.15)\\
& z_r \in \{0,1\}, \ \forall r \in \mathcal{R}. && (1.16)
\end{aligned}
$$

Constraint (1.14) ensures that at most m vehicles will be used. Constraint (1.15) guarantees that each customer is assigned to a route. In fact, using $x_e = \sum_{r \in \mathcal{R}} a_r^e z_r$ where a_r^e is one if edge e is used in route r and zero otherwise, the G-CVRP can be lifted to the SP's polytope in $\Re^{m+n}$ ([36]). The above problem has exponentially many columns. So, it is impractical to initially include all routes in the SP. A well-known strategy is to start with an initial set of routes and gradually add proper feasible routes (those which reduce the total cost) to the problem. Different methods and heuristics have been proposed to identify routes. We will study this subject in more detail in Chapter 5.

1.4 Basic background

Since in the next chapters we will be working with valid inequalities and lifting techniques, here we provide a brief introduction to these topics in the following sub-sections. We assume the reader has a working knowledge of the theory and practice of integer programming. For more details and in-depth, we refer to Nemhauser and Wolsey [57] and Wolsey [78].

1.4.1 Basic polyhedral theory

We need to have a basic knowledge of the polyhedral theory as we will try to find constraints (valid inequalities) that represent the feasible region of the problem as tightly as possible. Many books and articles have been devoted to study IP as it is one of the most challenging problems in the field of mathematical programming. Some basic definitions will be presented here.

A polyhedron is the intersection of finitely many affine halfspaces, where an affine halfspace is a set that can be defined as follows:

$$H(a, b) = \{x \in \Re : a^T x \leq b\}.$$

Let us assume an integer programming problem as follows.

$$z_{IP} = \min \{cx : x \in \Psi\}, \qquad \Psi = \left\{x \in \mathbb{Z}_+^n : Ax \leq b\right\}. \tag{1.17}$$

where c is an n-vector, (A, b) is a matrix of $m \times (n + 1)$, x is the decision vector and Ψ is the feasible solution set. x^* is an optimal solution for IP (1.17) if only if (*iff*), it is optimal to

$$z = \min \{cx : x \in \operatorname{conv}(\Psi)\}. \tag{1.18}$$

The term "conv" in the above problem stands for *convex hull*. The convex hull of Ψ is the smallest set which includes all points in Ψ and their convex combinations. In other words, the smallest convex set which contains Ψ. Several theories and ideas have been developed to find the convex hull of a set or at least find its local convex hull but finding all facets of a convex hull itself is NP-hard. Theory of valid inequalities, disjunctive programming and linear relaxation are the main theories in this attempt. Valid inequality and facets of a set, which are key concepts in IP, are defined as follows.

Definition 2. *The inequality $\pi x \leq \pi_0$ or $[(\pi, \pi_0)]$ is called a valid inequality for Ψ if it is satisfied by all points in Ψ.*

In other words, $\pi x \leq \pi_0$ is a valid inequality for Ψ if $\pi x \leq \pi_0$ for all $\forall x \in \Psi$ ([78]).

Definition 3. *If $[(\pi, \pi_0)]$ is a valid inequality for Ψ, and $\Gamma = \{x \in \Psi | \pi x = \pi_0\}$, Γ is called a face of Ψ.*

Definition 4. *A face Γ of Ψ is a facet of Ψ if $\dim(\Gamma) = \dim(\Psi) - 1$.*

Many different types of valid inequalities have been suggested to present or approximate the convex hull of a set such as Gomory, cross, split, disjunctive, intersection cuts, etc. We refer the reader to [6] and [57].

1.4.2 Lift-and-project cuts

Sherali and Adams [70] introduce a reformulation technique for 0-1 integer programming which generates cuts to present/approximate the convex hull of feasible solutions. Following their work, several methods have been proposed to generate cuts based on their procedure (see [7]). In fact, the lift-and-project procedure is a sequential convexification procedure that generates cuts via first, lifting a polyhedron to a higher dimensional space and then projecting the new polyhedron into the original variable space. A simple description is as follows. Let $\Omega = \{x \in \Re^n | Ax \geq b, x \geq 0\}$. Note that $x_i \leq 1,\ \forall i$ is embedded in $Ax \geq b$. And let $\Omega_I = \{x \in \Omega | x_i \in \{0,1\}\}$. Then, multiply $Ax \geq b$ by x_j and $(1-x_j)$ and linearize the inequalities using $x_j = x_j^2$ and $y_{ij} = x_i x_j$ for $i \neq j$. Let $\Phi_j(\Omega)$ be the polyhedron defined by the resulting valid inequalities. The projection of $\Phi_j(\Omega)$ into the original variable space is denoted by $\Phi_j^x(\Omega) = \{x | (x,y) \in \Phi_j(\Omega)\}$.

Balas et al. ([7]) prove the following main theorems which show the convex hull of Ω_I can be obtained using a sequential convexification procedure. For proofs see [7].

Theorem 1.

$$\Phi_j^x(\Omega) = conv\left(\Omega \cap \{x \in \Re^n | x_j \in \{0,1\}\}\right).$$

Theorem 2. *For $t \in \{1, ..., n\}$,*

$$\Phi_{i_1,...,i_t}^x(\Omega) = conv\left(\Omega \cap \{x \in \Re^n | x_j \in \{0,1\} \text{ for all } j \in \{i_1, ..., i_t\}\right).$$

These theorems imply that if in each stage the projection of the resulting space of the previous stage onto some of variables are calculated, then it leads into the convex hull of the original space i.e., $\Phi_{1,...,n}^x(\Omega) = \text{conv}\,(\Omega_I)$.

1.5 Dissertation overview

This dissertation is organized in three main parts. In the first and second parts we study two variants of VRP with uncertain demands. In part one, we formulate and solve the Heterogeneous Vehicle Routing Problem with Uncertain Demands (HVRPUD) within a branch-and-cut method. We study HVRPUD in a single stage framework. In part two, we study a special case of HVRPUD, the Capacitated Vehicle Routing Problem with Uncertain Demands (CVRPUD), where there is only one type of vehicles. We study this special case in single stage and two stage frameworks. To model CVRPUD in these two frameworks, we use set-covering formulations which is also known as column generation methods. The advantage of column generation methods over branch-and-cut methods is that if for a specific problem there is an efficient pricing problem, then column generation methods usually outperform branch-and-cut methods, but finding an efficient pricing problem may not be easy. On the other hand, branch-and-cut methods provide a more flexible framework for formulating optimization problems. Also, more efficient and advanced software have been developed to solve problems within branch-and-cut methods while to implement column generation methods, there is only one developed software (SCIP). If one chose not to use SCIP, one would have to implement the whole structure, which might result in a less efficient software compared to commercially developed. Therefore, we first formulate and solve HVRPUD within a well-defined branch-and-cut framework and then for its special case, we move to a more advanced framework (column generation methods) and propose new formulations. In the third part, we present an extensive computational experiments for these two variants within branch-and-bound and column generation methods, respectively.

More precisely, in Chapter 2, we first present a basic formulation for the deterministic HVRP. Then, we investigate and extend four different types of valid inequalities namely: capacity, sub-tour elimination, comb and multistar valid inequalities plus we introduce a customized set of valid inequalities. In addition to valid inequalities, we study and extend three types of lifting techniques for the deterministic HVRP. These valid inequalities and lifting techniques improve the approximation of the convex hull of our problem. This investigation results in a formulation with much better lower bounds in comparison with the basic formulation.

In Chapter 3, we apply three single-stage approaches to the models introduced in Chapter 2. The aim of this chapter is to derive tractable deterministic equivalents for the HVRPUD. As mentioned, for single-stage approaches, we study chance constraint programming and two different types of robust optimization approach: Ben-Tal and Nemirovski (BN), and Bertsimas and Sim (BS) approaches.

As sub-tour elimination and comb valid inequalities are exponential in size, they have to be added to the problem sequentially if they are violated. In Chapter 4, we first review separation algorithms for these two types of valid inequalities within cutting plane based algorithms. Then, we propose greedy algorithms to separate them.

The second part of this dissertation consists of two chapters where we study single-stage and two-stage models for the CVRPUD within column-generation based methods. In Chapter 5, we review set-partitioning formulations for the deterministic CVRP and introduce a formulation which is most suitable for the CVRPUD. Then, the related issues such as the column generation problem and the pricing problem will be explained. Finally, we apply CCP, BN and BS approaches to the CVRPUD.

In the Chapter 6, we study the CVRP with stochastic demands and recourse actions. Different recourse actions have been suggested in the literature to serve the remaining customers. The recourse action we consider here is that if a failure occurs, the vehicle must make a return trip to the depot for a replenishment and resume the pre-planned route. In this case, lost sales are not allowed and the remaining customers on the failed route have to be served. Unlike the single-stage models, in the two-stage models, the recourse actions are modelled within the initial formulation. In Chapter 7, we present an extensive computational experiment to assess the performance of the models and approaches within stochastic optimization we apply to the HVRPUD and the CVRPUD.

Finally, Chapter 8 comprises a conclusive summary of the whole thesis. It also discusses the line of future inquiry flowing out of the present research as well as other possible approaches that can be adopted to extend this work.

Before proceeding to the rest of this thesis, we will briefly describe below the key research questions and the contributions.

In this research we address the following questions:

- What are specific properties of our proposed VRPUD formulations?
- Which methods in stochastic programming and mixed-integer programming can be used to improve VRPUD solution algorithms?

The contributions of this dissertation are categorized into two parts. The contributions of the first part are as follows. We improve the formulation for the deterministic

HVRP in the sense that the resulting polyhedral is a better approximation of the convex hull of the deterministic HVRP. To this end, we adapt four types of valid inequalities to the deterministic HVRP and also propose a new type of valid inequalities. Moreover, we extend three lifting techniques for this problem. In particular, we extend the *reformulation and linearization technique* which was originally developed for the TSP. Then, for the first time we apply three single-stage approaches to the HVRP with uncertain demands to capture the demand uncertainty and solve the resulting problems. In addition, we propose better probability bounds for Bertsimas and Sim's approach for specific types of constraints.

In terms of the solution method, we propose two new greedy separation algorithms to separate sub-tour elimination and comb valid inequalities. Finally, in the computational experiments, we present the computational studies for the models and algorithms we develop in the previous chapters. Using computational experiments, we show the impact of each type of the valid inequalities and the lifting techniques on the deterministic HVRP polyhedral. Then, using simulation we conduct a scenario-based analysis for the HVRP with uncertain demand for the control parameters of the single-stage models. When lost sales are not allowed, we investigate which approach among the three single-stage approaches with which control parameters will lead to the least actual cost. In this case, the actual cost consists of the routing cost plus the cost of return trips to the depot. Moreover, when lost sales are allowed, we calculate optimum intervals of lost sale costs for each scenario of the control parameter.

In the second part of this dissertation we study the CVRPUD within column generation based methods. For the first time we formulate the single-stage models for the CVRPUD within a column generation based method. The contributions here mainly lie in the pricing problem how to generate feasible routes which satisfy the conditions within each approach of stochastic optimization. As mentioned, we study four approaches to model the demand uncertainty: three single-stage and one two-stage approaches. We present new pricing problems to formulate the demand uncertainty where for CCP in addition to different distribution functions, demand scenarios can also be used. Then we formulate the CVRPUD with recourse action within the context of the two-stage stochastic programming where a new method of calculating recourse functions is proposed in the pricing problem. Our proposed method of calculating the recourse cost guarantees that no feasible routes, which may be part of the optimal solution, will not be eliminated. We propose a new dominant rule to make sure that no such elimination will take place and at the same time not all possible routes will be enumerated. Similar to the single-stage approaches, distribution functions as well as demand scenarios can be used to present the demand uncertainty. In addition, we compare these four approaches of formulating the CVRPUD and discuss their advantages for the first time.

Chapter 2

VALID INEQUALITIES AND LIFTING TECHNIQUES

2.1 Introduction

In this chapter, we consider an important generalization of the classical CVRP known as Heterogeneous Vehicle Routing Problem (HVRP), in which a heterogeneous fleet of vehicles is stationed at the depot and is used to serve the customers. To give an indication how difficult it is to solve the HVRP, it is worth mentioning that up to the date of writing this dissertation, the computational results show HVRP instances involving only up to 75 vertices can be solved to optimality ([10] and [63]) whereas CVRP instances solved to optimality are far larger, up to 200 vertices ([10] and [36]). These results themselves suggest that the HVRP is more difficult to solve than the CVRP. Despite this fact, there are very few works on the HVRP and its variants.

Due to the lack of research on the HVRP's polyhedron, in this chapter we address the integer programming representation of the HVRP and its two main variants: multi-depot HVRP (MD-HVRP) and capacitated multi-depot HVRP (CMD-HVRP).

The reason behind studying the HVRP's polyhedron is as follows. To solve MIPs within branch and bound/cut methods, there are two general strategies based on which exact algorithms are developed. The first strategy focuses on the initial formulation so that it identifies efficient cutting plane and (if it is possible) facet defining inequalities and adds them to the initial model to tighten the polyhedral representation of the problem before any computational solution procedure is started. This strategy is called *static*. The second strategy is *dynamic* where cutting planes are added during run-time, which successively reduces the size of the polyhedral region. Therefore to solve an integer program efficiently using any of these strategies, it is very important to study the problem's valid inequalities and polyhedron presentation.

In view of the fact that we intend to use the resulting formulations to construct the HVRP with uncertain demand, we ought to take into consideration another goal for our formulations as well. This gaol is to formulate the deterministic HVRP in such a way that the corresponding counterparts of uncertainty remain tractable via mixed integer linear programming. To achieve this goal, we build our basic model based on Miller et al. (MTZ) [55] formulation for the Symmetric TSP (STSP) where sub-tours are eliminated using an extra continuous variable on the MTZ formulation. The main advantage of this basic model is that uncertainty is restricted to the right-hand side of the constraints. This leads to compact and tractable uncertain counterparts. Since the MTZ formulation is well known to provide a rather weak linear programming (LP) relaxation, which performs poorly when plugged into a branch-and-bound framework, we aim to overcome this weakness by using valid inequalities and lifting techniques. We begin with a basic integer programming model for the deterministic HVRP as our basic model. Then, we study and extend different classes of valid inequalities and lifting techniques to the HVRP and its variants to improve the formulation.

Since the HVRP is a generalization of the CVRP and as a result a generalization of the TSP, and also since the CVRP's polyhedral is connected to other IP problems' polyhedral (such as the spanning tree and many others), developing and extending existing valid inequalities of the CVRP to the HVRP seem to be a reasonable approach to study the polyhedron of the HVRP. However, even though many constraints and valid inequalities have been proven to be facet-defining for the TSP, they are only valid for the CVRP and the HVRP due to their complex convex hull.

The reminder of this chapter is organized as follows. In Section 2.2 we introduce a basic model then in Section 2.3 we study capacity, sub-tour elimination, comb, multistar valid inequalities as well as a customized version of cross cuts and extend them for the HRVP if possible. In Section 2.4, we review and adapt three lifting techniques for our basic mixed integer formulation: Desrochers and Laporte, Yaman and Sherali and Driscoll lifting techniques. Finally in Section 2.5, we extend the model to multi-depot HVRP (MD-HVRP) and capacitated multi-depot HVRP (CMD-HVRP).

2.2 The basic formulation

The HVRP can be formally defined as follows. We are given a complete directed graph $G = (V, E)$, where $V = \{0, \ldots, n\}$ is the set of vertices, E the set of edges and $E_c \subset E$ is the sub-set of edges between customers. Node 0 denotes the (unique) depot and the other vertices $V_c = \{1, \ldots, n\}$ represent customers. A fleet of heterogeneous vehicles is stationed at the depot. Without loss of generality we assume that there are m different vehicle types $K = \{1, \ldots, m\}$ and, for each type $k \in K$, there is only one vehicle available with capacity $Q_k > 0$, where $Q_1 \leq \cdots \leq Q_m$. Accordingly K corresponds to the set of all vehicles and m is the total number of vehicles available at the depot. The cost of traveling from vertex i to vertex j (arc $e = (i, j)$) by vehicle k is denoted by c_e^k. Each customer i has an integer demand q_i, with $0 < q_i \leq Q_m$. Since splitting demand is not allowed, each customer must be served by exactly one vehicle. Furthermore, a vehicle cannot serve a set of customers whose total demand exceeds its capacity. The problem is to find m vehicle routes of minimum cost, where each vehicle leaves the depot, visits a sub-set of customers and finally returns to the depot.

There are three main classes of formulations: *vehicle flow*, *two-commodity flow* and *set partitioning*. In this chapter, we will follow a vehicle flow formulation. In this method, one can choose a *two-index* vehicle flow formulation, which uses x_{ij}, $e = (i, j) \in E$ variables, or a *three-index* vehicle flow formulation, which uses x_{ij}^k, $e = (i, j) \in E$, $k \in K$ variables. We will use the latter formulation as it is particularly suited for heterogeneous vehicles.

Let x_e^k be a binary variable, indicating whether vehicle k travels from vertex

i to vertex j (edge $e = (i, j)$). Also, let $u_i,\ i \in V_c$ be a continuous variable representing the total demand of vertices on a route from the depot (vertex 0) to vertex i. Finally, given a vertex $i \in V$, let $\delta^-(i)$ and $\delta^+(i)$ denote the set of incoming and outgoing edges of i, respectively. In addition, we set $\delta(i) = \delta^+(i) \cup \delta^-(i)$. The MILP formulation is then:

$$\min \quad \textstyle\sum_{k\in K} \sum_{e\in E} c_e^k x_e^k \tag{2.1}$$

$$\text{s.t.} \quad \textstyle\sum_{e\in\delta^+(i)} x_e^k - \sum_{e\in\delta^-(i)} x_e^k = 0, \quad i \in V,\ k \in K \tag{2.2}$$

$$\textstyle\sum_{k\in K} \sum_{e\in\delta^+(i)} x_e^k = 1, \quad i \in V_c \tag{2.3}$$

$$\textstyle\sum_{k\in K} \sum_{e\in\delta^-(i)} x_e^k = 1, \quad i \in V_c \tag{2.4}$$

$$\textstyle\sum_{e\in\delta^+(0)} x_e^k = 1, \quad k \in K \tag{2.5}$$

$$\textstyle\sum_{e\in\delta^-(0)} x_e^k = 1, \quad k \in K \tag{2.6}$$

$$-u_j + u_i + Q_m \textstyle\sum_{k\in K} x_e^k \leq Q_m - q_j, \quad e = (i, j) \in E_c \tag{2.7}$$

$$q_i \leq u_i \leq \textstyle\sum_{k\in K} Q_k \sum_{e\in\delta^+(i)} x_e^k, \quad i \in V_c \tag{2.8}$$

$$x_e^k \in \{0, 1\}, \quad e \in E,\ k \in K. \tag{2.9}$$

The degree equations (2.2–2.6) ensure that all customers are visited exactly once and for each vehicle there is exactly one route starting from and terminating at the depot. Inequalities (2.7–2.8), referred to as Miller-Tucker-Zemlin constraints, ensure that the routes are connected and, at the same time, impose vehicle capacity restrictions. Constraints (2.9) are the integrality conditions on the x_e^k variables.

2.3 Valid inequalities

In this subsection, we present valid inequalities for the HVRP. Recall that the definition of valid inequality for polyhedron Ψ is as follows: $\pi x \leq \pi_0$ is a valid inequality for Ψ if $\pi x \leq \pi_0$ for all $x \in \Psi$. Hence, we need to search for pairs (π, π_0) that are valid for the HVRP. Since the HVRP is defined on a graph, the main focus on finding valid inequalities is to identify appropriate combinations of sub-sets of vertices and appropriate coefficients that lead to a valid inequality.

2.3.1 Capacity inequalities

Although the current MTZ constraints (2.7–2.8) forbid violation of the vehicle capacity, we introduce the capacity inequality to our model in order to straighten the LP relaxation. The capacity constraint can be presented as follows:

$$\sum_{i \in V_c} \sum_{e \in \delta^+(i)} q_i x_e^k \leq Q_k, \ k \in K. \tag{2.10}$$

The above constraint simply guarantees that the demands on route k have to be less that the capacity of vehicle Q_k. Yaman [79] improves the above constraint and introduces two new constraints as follows:

$$\sum_{i \in V_c} \sum_{e \in \delta^+(i)} q_i x_e^k \leq Q_k \sum_{i \in \delta^+(0)} x_e^k, \ k \in K, \tag{2.11}$$

$$\sum_{k \in K} \sum_{i \in V_c} \sum_{e \in \delta^+(i)} \lceil \frac{Q_k}{Q} \rceil x_e^k \geq \lceil \frac{q(V_c)}{Q} \rceil, \tag{2.12}$$

where Q can be any of $Q_1, ..., Q_m$. Yaman calls this type of valid inequalities *cover inequalities.*

These valid inequalities imply that total demand assigned to a vehicle has to be less than or equal to the vehicle capacity. Computational results (see Chapter 7) suggest that this type of valid inequalities has a significant impact on the LP relaxation.

2.3.2 Sub-tour elimination inequalities

It is well known that any valid inequality for the two-index vehicle flow formulation can be transformed into a valid inequality for the three-index vehicle flow formulation by using $x_e = \sum_{k=1}^{m} x_e^k$. These inequalities are called *aggregated* by Letchford and Salazar-González [53]. *Sub-tour elimination* inequalities introduced by Dantzig et al. are rather common constraints for the CVRP two-index vehicle flow formulation, sometimes called *rounded capacity* inequalities. As briefly mentioned in Chapter 1, these constraints forbid sub-tours and those routes that exceed the vehicle's capacity. Constraint (2.13) states that for any sub-set S of customers (excluding the depot) at least $\lceil q(S)/Q \rceil$ vehicles enter and leave S, where $q(S) = \sum_{i \in S} q_i$ and Q is the vehicle capacity. Indeed these inequalities are extended and relaxed version of constraint (1.8) which is the DFJ sub-tour elimination inequalities for the TSP. Here we present an extension to the three-index vehicle flow representation for the heterogeneous case. Let $(S : T) = \{(i, j) = e \in E : i \in S, j \in T\}$ and $x(E(S : T)) = \sum_{k \in K} \sum_{e \in (E(S:T))} x_e^k$.

For any $S \subseteq V_c$, the inequality

$$x(E(S:\bar{S})) \geq 2\left\lceil \frac{q(S)}{Q_m} \right\rceil \tag{2.13}$$

is a valid inequality for the HVRP three-index vehicle flow formulation ($\bar{S} = V_c \setminus S$). Note that, although this extension provides valid inequalities for HVRP and forbids all sub-tours, it may allow routes that exceed the vehicle capacity. This is due to the fact that in the HVRP the right-hand side of the inequality depends on the capacity of the vehicle (and hence, by using Q_m, we overestimate the denominator), whereas in the classical CVRP, all vehicles have the same capacity Q. To overcome this problem we use the constraints adopted by Yaman [79] and disaggregate such inequalities in the following way:

$$x(E(S:\bar{S})) \geq 2\left\lceil \frac{q(S)}{Q_k} \right\rceil, \ k \in K, \ S \subseteq V_c. \tag{2.14}$$

2.3.3 Comb valid inequalities

Edmonds [32] introduced *2-matching* constraints for the TSP. Following this work, Grotschel and Padberg [41] study several classes of inequalities for the symmetric travelling salesman problem and introduce a new class of valid inequalities known as *comb valid inequalities* which are the more generalized type of 2-matching constraints. They prove the 2-matching and comb valid inequalities are facet-defining for the TSP polytope. Grotschel and Holland [40] present an extensive study on the STSP's polyhedron and solve large-scale STSPs. Later, comb valid inequalities were adapted for the CVRP (see [3],[12],[40]) and have been used successfully within cutting plane based methods. Here we first introduce 2-matching and comb valid inequalities and then extend them for our heterogeneous cases. Let $H, T_1, ..., T_s$ be a set of sub-sets of V such that they satisfy

$$|T_t \cap H| = 1,\ t = 1, ..., s,$$

$$|T_t \setminus H| = 1,\ t = 1, ..., s,$$

then the following inequality is called 2-matching valid inequalities introduced by Edmonds [32]

$$x(E(H)) + \sum_{t=1}^{s} x(E(T_t)) \leq |H| + \sum_{t=1}^{s}(|T_t| - 1) - \lceil \frac{1}{2}s \rceil. \tag{2.15}$$

The above valid inequality can be equally restated:

$$x(\delta(H)) + \sum_{t=1}^{s} x(\delta(T_t)) \geq 3|T| + 1. \tag{2.16}$$

Figure 4.4 shows the setting for a 2-matching inequality with $s = 3$. The first generalization of the 2-matching inequality which was carried out by Grotschel and Padberg [41] was to extend the condition on the teeth and the handle i.e., each tooth can have more than two vertices. The new condition is presented as follows:

$$|T_t \cap H| \geq 1,\ t = 1, ..., s,$$

$$|T_t \setminus H| \geq 1,\ t = 1, ..., s.$$

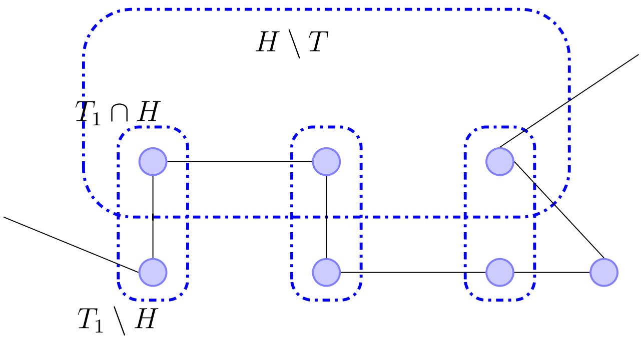

Figure 2.1: A 2-matching with $s = 3$.

Under the above conditions, inequalities (2.15) and (2.16) are called *comb valid inequalities*. Later these valid inequalities were extended to the CVRP in several works ([3],[12],[40]). While combs with disjoint teeth and odd numbers are facets for the TSP, they are only valid inequalities for the CVRP and its variants. It can be proven that the above inequalities are valid for the variants of the VRP. The right-hand side of (2.16) is in fact the minimum number of vehicles required to serve sub-sets of a comb as follows:

$$\bar{r} = \sum_{t=0}^{s} (k(T_t \cap H) + k(T_t \setminus H) + k(T_t)) + 1$$

where $k(S)$ is the minimum number of vehicles needed to serve customers in S. Replacing the right-hand side of (2.16) with $\bar{r}$ results in a so-called *strengthened comb inequality* which is valid for the family of HVRPs as the number of vehicles is its only requirement ([54]). Due to the difficulty in calculating an exact value for $\bar{r}$, it can be approximated as follows. $3|T|$ is the smallest possible number of vehicles needed to serve the sets of $T_t \cap H$, $T_t \backslash H$ and T_t.

2.3.4 Multistar inequalities

Araque et al. [4] study the polyhedrons of the subtree cardinality-constrained minimal spanning tree problem and the capacitated identical customer vehicle routing problem. For the first time, they introduced *multistar* and *partial multistar* inequalities for both problems. These two problems are closely related such that if the last link of each route is eliminated, then any feasible set of routes becomes a feasible solution to the subtree cardinality-constrained minimal spanning tree problem. Hence, the polyhedral structure of these two problems are also connected ([4]).

Letchford et al. [52] extend the work of Araque et al. to the CVRP with general demands. They show validity of different types of multistar and partial multistar inequalities for the CVRP. Another important topic discussed in their paper is cutting plane procedures for the valid inequalities they study. Araque et al. [4] initially introduced three types of multistar inequalities (large, intimidate and small) and four types of partial multistars valid inequalities. Here we review them and extend to the HVRP if possible.

Multistar Valid Inequalities Multistars were defined initially for the capacitated vehicle routing problem with unit demand and the related integer programs, using two types of sub-sets of the vertices called *nucleus* and *satellite*. Araque et al. define a multistar as follows: *a multistar consists of the complete sub-graph on a set of nucleus vertices N, together with a set of satellite vertices S and the edges connecting*

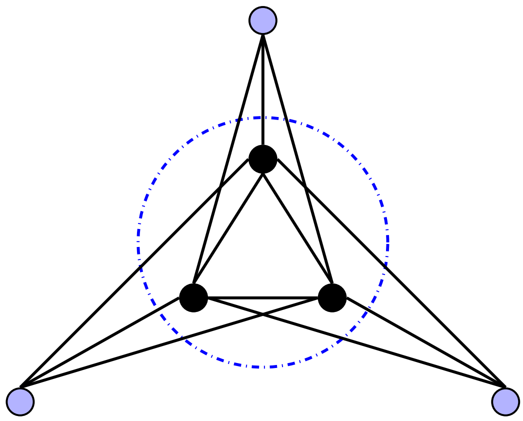

Figure 2.2: A multistar with $|N| = 3$ and $|S| = 3$

every satellite vertex to every nucleus vertex. Figure 2.2 shows a multistar with the nucleus (N) presented with the black vertices and the satellite (S) presented with the blue vertices. For a given nucleus $N \subset V \setminus \{0\}$ and a given satellite $S \subseteq \bar{N}$ (where $\bar{N} = V \setminus (N \cup \{0\})$), Araque et al. form the following general multistar inequality:

$$bx(E(N)) + x(E(N : S)) \leq rhs \tag{2.17}$$

where the constants b and rhs depend on the sizes of the nucleus and satellite sets, and $E(N : S)$ denotes the set of edges between N and S. In the capacitated vehicle routing problem with unit demand there is only one type of vehicle available with capacity Q and all customers have unit demand. When $b = Q$ and $rhs = (Q-1)|N|$, inequality (2.17) is valid and is called the *large multistar* (LM) inequality. When $b = 2 + |N| \bmod (Q-2)$ and $rhs = b|N| - (b-2)\lceil\frac{|N|}{Q-2}\rceil$, inequality (2.17) is again valid and but is called the *intermediate multistar* (IM) if $3 \leq b \leq 2\lceil\frac{|N|}{Q-2}\rceil$. Finally, inequality (2.17) is valid and called the *small multistar* (SM) if $|N \cup S| > Q$ and $2 \leq b < |S|$ for $b = |N \cup S| \bmod Q$ and $rhs = b(|N| - k(N \cup S)) + |S|$. Recall from (1.8) that $k(N \cup S)$ is the minimum number of vehicles needed to serve customers in $N \cup S$.

The large multistar inequality can be extended to the capacitated vehicle routing problem with general demands as follows:

$$Qx(E(N)) + \sum_{j \in \bar{N}} q_j x(E(N : \{j\})) \leq Q|N| - q(N), \ \forall N \in V_c : |N| \geq 2. \quad (2.18)$$

The above inequality is known as *generalized large multistar* (GLM) inequality ([52]). In fact the GLM inequality guarantees that the total demands of customers in the nucleus and the customers visited by vehicles immediately after leaving the nucleus is less than or equal to a specific value. This specific value is the number of edges leaving the nucleus multiplied by Q. A natural generalization of the GLM inequality for the HVRP is to substitute Q with the smallest and largest vehicle capacities Q_1 and Q_m, respectively:

$$Q_1 x(E(N)) + \sum_{j \in \bar{N}} q_j x(E(N : \{j\})) \leq Q_m|N| - q(N), \ \forall N \in V_c : |N| \geq 2.$$

Of course, these valid inequalities can be disaggregated for each type of vehicle by replacing Q_1 and Q_m with Q_k. Yaman [79] modifies the GLM for the HVRP and suggests a valid inequality that dominates above valid inequality. She uses the same idea to compute the number of edges leaving the nucleus for each type of vehicles

(k).

$$\sum_{k\in K_{ij}} \sigma_{ik} x^k(E(S:N)) \geq q(N) + \sum_{j\in \bar{N}} q_j x^k(E(N:\{j\})),\ \forall N \in V_c : |N| \geq 2 \quad (2.19)$$

where $\sigma_{ik} = \min\{Q_k - q_i, q(N) + \max_{l\in\delta^k(N)} q_l\}$. As we can see, the idea of the capacity inequalities is implicitly embedded in the last three inequalities.

However, attempts to extend the IM and the SM to the CVRP have failed ([52]). Inequalities (2.18) and (2.19) are called *inhomogenous* as the coefficients of x vary in these inequalities depending on the customers' demand.

Partial Multistar Valid Inequalities Araque et al. [4] generalize the multistars for the capacitated vehicle routing problem with unit demand by making the following changes: *instead of including all the edges connecting the nucleus vertices and the satellite vertices, the support graph contains only those edges that are incident to a sub-set C of the nucleus vertices; we refer to this sub-set as the connector vertices.* For a given nucleus $N \subset V \setminus \{0\}$, a given satellite $S \subseteq \bar{N}$ and a given connector $C \subset N$, Araque et al. define a general *partial multistar inequality* as follows:

$$ax(E(N)) + x(E(C:S)) \leq rhs \quad (2.20)$$

where a is a constant depending on N, S, C and the type of partial multistar. There are four types of partial multistars and each one is valid for certain conditions. Here we list the conditions for each type ([52]).

1. The first type is called *one-connector partial multistars* ($|C| = 1$). When $Q \geq 3$

and $|N|$ is a multiple of Q, then $a = 2$ and $rhs = 2(|N| - k(N))$.

2. The second type is called *two-connector partial multistars* ($|C| = 2$). When $Q \geq 4$ and $|N|$ mod $Q = 1$, then $a = 2$ and $rhs = 2(|N| - k(N) + 1)$.

3. The third type is called *three-connector partial multistars* ($|C| = 3$). When $Q \geq 4$ and $|N|$ is a multiple of Q, then $a = 3$ and $rhs = 3(|N| - k(N))$.

4. The forth type is also called *three-connector partial multistars* ($|C| = 3$). When $Q \geq 4$ and $|N|$ is a multiple of Q, then $a = 2$ and $rhs = 2(|N| - k(N)) + 1$.

The homogenous multistar inequalities: Letchford et al. [52] propose some approximations for the homogenous multistar inequalities since it is NP-hard to find homogeneous multistar inequalities for the CVRP. Recall $k(S)$ be the minimum number of vehicles required to serve the customers in S. All feasible solutions of the CVRP satisfy

$$x(E(C:S)) \leq \min\{2|C|, 2|S|, |C| + |S| - k(C \cup S)\}, \tag{2.21}$$

$$x(E(C:S)) \geq 2\lceil \tfrac{q(S)}{Q_m} \rceil. \tag{2.22}$$

It is easy to see the validity of inequality of (2.21). It is trivial that $x(E(C:S)) \leq \min\{2|C|, 2|S|\}$ and constraint (1.8) implies $x(E(C:S)) \leq \{|C| + |S| - k(C \cup S)\}$. As inequality (2.22) suggests, it is an extension of the sub-tour elimination constraint (2.13). One can see the above approximations are also valid for the HVRP.

2.4 Lifting technique

It is known that valid inequalities can be strengthened via lifting. Desrochers and Laporte [27] propose a simple lifting technique for the MTZ constraints for the TSP. Here we extend their technique to the HVRP. To simplify notation we denote by $x_{ij} = \sum_{k \in K} x_{ij}^k$.

Proposition 1. *The lifted version of constraints (2.7) is as follows:*

$$-u_j + u_i \quad + Q_m x_{ij} + (Q_m - q_j - q_i) x_{ji} \leq Q_m - q_j, \ (i,j) \in E_c. \tag{2.23}$$

Proof. If $x_{ij} = 1$ then $x_{ji} = 0$, so we obtain the original MTZ inequality. On the other hand, if $x_{ji} = 1$, then the inequality reduces to $u_i \leq u_j + q_i$, which is again valid according to MTZ. □

Similarly it is possible to lift the MTZ upper bound in (2.8) as follows:

$$u_i \leq \sum_{k \in K} Q_k \sum_{j \in V} x_{ij}^k - \sum_{j \in V_c} q_j x_{ij}, \ i \in V_c. \tag{2.24}$$

For any customer $i \in V_c$, its successor can be either another customer or the depot. If it is a customer $j \in V_c$, then $u_i \leq u_j - q_j$ is valid. If it is the depot, the term $\sum_{j \in V_c} q_j x_{ij}$ is zero and we obtain the original MTZ upper bound. We call the model of (2.1–2.6) & (2.8–2.9) & (2.23-2.24) HVRP-DL for brevity.

2.4.1 Yaman's technique

Yaman [79] proposes constraints (2.25) similar to the MTZ constraint to calculate the flow of the product using a new flow variable (t_{ik}) for each vehicle i.e., it calculates the total demand of vertices on a route that uses vehicle k after visiting customer i.

$$t_{jk} \geq t_{ik} + q_j \textstyle\sum_{i \in V_c} x_{ij}^k - Q_k(\sum_{j \in V_c} x_{ij}^k - x_{ij}^k), \ \forall i, j \in V_c, \ \forall k \in M, \quad (2.25a)$$

$$t_{ik} \geq q_i \textstyle\sum_{j \in V_c} x_{ij}^k + \sum_{j \in V_c} q_j x_{ji}^k, \ \forall i \in V_c, \ \forall k \in M, \quad (2.25b)$$

$$t_{ik} \leq Q_k \textstyle\sum_{j \in V_c} x_{ij}^k, \ \forall i \in V_c, \ \forall k \in M. \quad (2.25c)$$

The first two constraints compute t_{it} and the third constraint ensures that the vehicle capacity is not violated. However, Yaman argues that the above set of constraints provides a weak lower bound. Therefore, Yaman improves them by adding the following terms to the right-hand side of the first two constraints, respectively.

$$(Q_k - q_i - q_j)x_{ji}^k, \quad (2.26)$$

$$-(Q_k - q_i - \max_j q_j) \textstyle\sum_{j \in V_c} x_{ji}^k. \quad (2.27)$$

This improvement can be seen as an extension of Desrochers and Laporte [27] lifting technique as Yaman uses the same idea. Hence, the new constraints will be written as follows for $k \in K$ and $\forall i, j \in V_c$:

$$t_{jk} \geq t_{ik} + q_j \textstyle\sum_{i \in V_c} x_{ij}^k - Q_k(\sum_{j \in V_c} x_{ij}^k - x_{ij}^k) + (Q_k - q_i - q_j)x_{ji}^k, \quad (2.28a)$$

$$t_{ik} \leq Q_k \textstyle\sum_{j \in V_c} x_{ijk} - \sum_{j \in V_c} q_j x_{ij}^k - (Q_k - q_i - \max_j q_j) \sum_{j \in V_c} x_{ji}^k. \quad (2.28b)$$

2.4.2 Reformulation and linearization technique

We apply a specialized version of the well-known Reformulation-Linearization Technique (RLT) by Sherali and Adams [70] to the MTZ constraints to improve its LP relaxation. In particular, to contain the size of the resulting model, we follow Sherali and Driscoll [71], who only apply a partial first-level RLT version and provide a relatively tight formulation for the TSP. The MTZ constraints (2.7) can be re-stated as follows:

$$u_j x_{ij} = (u_i + q_j)x_{ij}, \; (i,j) \in E_c, \quad (2.29a)$$

$$u_j x_{0j} = q_j x_{0j}, \; j \in V_c. \quad (2.29b)$$

We call the model (2.1–2.6), (2.8–2.9) and (2.29a–2.29b) HVRP-NL for brevity.

We now apply the specialized version of RLT by Sherali and Driscoll [71] to HVRP-NL. The approach consists of two steps. First, we *reformulate* by generating additional (non-linear) implied constraints. Second, we *linearize* the nonlinear terms using a substitution of variables in place of each distinct nonlinear term.

Reformulation: We reformulate the HVRP-NL by generating three sets of quadratic constraints as follows.

(S1): Multiply by u_i both the degree constraints (2.3) and (2.4).

(S2): Multiply the first inequalities in (2.8) by x_{ij} and $(1 - x_{ij} - x_{ji})$, respectively.

(S3): The second inequalities in (2.8) suggest that $(Q_m - u_j) \geq 0$, which we multiply by x_{ij} and $(1 - x_{ij} - x_{ji})$, respectively.

Linearization: We linearize the HVRP-NL along with the three new sets of constraints (S1)–(S3) generated above using the following substitution of variables:

$$y_{ij} = u_i x_{ij} \text{ and } z_{ij} = u_j x_{ij}. \tag{2.30}$$

Note that y_{ij} can be interpreted as the load of the vehicle *before* visiting customer j, if j is served after customer i, and zero otherwise. Similarly, z_{ij} can be interpreted as the load of the vehicle *after* visiting customer j, if j is served after customer i, and zero otherwise. Also, we can replace $u_j x_{0j}$ by $q_j x_{0j}$ using (2.29b), and we can bound $u_j x_{j0}$ from above using $Q_k x_{j0}$. Note that we can always eliminate z_{ij} using the relationship $z_{ij} = y_{ij} + q_j x_{ij}$. The linearization step yields the inequalities given below.

Proposition 2. *Denote by $\delta_c^+(i)$ the set of arcs $(i, j) \in E_c$. Linearization of (S1) leads to the following:*

$$\sum_{(i,j)\in\delta_c^+(i)} y_{ij} + \sum_{k\in K} Q_k x_{i0}^k - u_i \geq 0. \tag{2.31}$$

and

$$\sum_{(j,i)\in\delta_c^-(i)} z_{ji} + q_i x_{0i} - u_i = 0. \tag{2.32}$$

Proof. Multiplying (2.3) by u_i we obtain

$$\sum_{(i,j)\in\delta^+(i)} u_i x_{ij} - u_i = 0.$$

Then substituting y_{ij} and observing that the load of a vehicle u_i leaving customer i and entering the depot must be less than or equal to the capacity of the vehicle Q_k, yields the inequalities. Similarly, multiplying (2.4) by u_i we obtain

$$\sum_{(j,i)\in\delta^-(i)} u_i x_{ji} - u_i = 0.$$

Then substituting z_{ji} and using (2.29b) we obtain the equations. □

Next, (S2) and (S3) can be linearized simply by substituting the quadratic terms with their corresponding variables. Hence, linearization of (S2) leads to

$$z_{ij} \geq q_j x_{ij}, \tag{2.33a}$$

$$u_j \geq z_{ij} + y_{ji} + q_j - q_j x_{ij} - q_j x_{ji}; \tag{2.33b}$$

and linearization of (S3) leads to:

$$z_{ij} \leq Q_m x_{ij}, \tag{2.34a}$$

$$u_j \leq Q_m(1 - x_{ij} - x_{ji}) + z_{ij} + y_{ji}. \tag{2.34b}$$

Note that in all the new sets of constraints introduced above, z_{ij} can be eliminated by substituting it by $y_{ij} + q_j x_{ij}$.

Extending the argument of Sherali and Driscoll [71], we conclude on validity and the tightness of our new formulation as follows.

Proposition 3. *The formulation obtained by replacing (2.7–2.8) with (2.31), (2.32), (2.33a–2.34b) is valid and provides an LP relaxation that is tighter than the LP relaxation of the HVRP-DL.*

Proof. The validity follows by construction. Hence it suffices to show that the constraints (2.31), (2.32), (2.33a–2.34b) imply (2.23). To do so, first we replace z_{ij} with $y_{ij} + q_j x_{ij}$ in (2.33b) and in (2.34b), then we multiply (2.34b) by -1 and finally we interchange i and j in (2.34b). By surrogating the resulting inequalities we obtain

$$0 \geq u_i - u_j - Q_m + (Q_m - q_i - q_j)x_{ji} + Q_m x_{ij} + q_j,$$

which is (2.23). $\square$

This proposition will be supported by computational experiments in Section 7.1.

2.4.3 Customized valid inequalities

In addition to the valid inequalities developed in the literature, we can improve the LP relaxation of the HVRP by using some customized valid inequalities. The type of valid inequalities presented here can be seen as a specific version of *intersection cuts* developed by Balas [5]. Intersection cuts are known as one of the most successful cuts developed for mixed integer programs [28]. On the basis of the intersection cuts, two other classes of cuts have been developed: *split cuts* and *cross cuts*. The aim of these techniques is to generate facets of the integer hull. The idea is as follows. Assume $\bar{x}$ is a non-integer optimal solution of the LP relaxation. A unit hypercube can be defined so that it contains $\bar{x}$ and its vertices which are integer. Also, a hypersphere is defined so that it circumscribes the hypercube. The hyperplanes whose intersections define $\bar{x}$, intersect the hypersphere at n points (let us assume the solution is not degenerated). The hyperplane through these n points is a valid cut. Figure 2.3 shows an example in three-dimensional space. The shaded hyperplane is the cut passing through a, b and c.

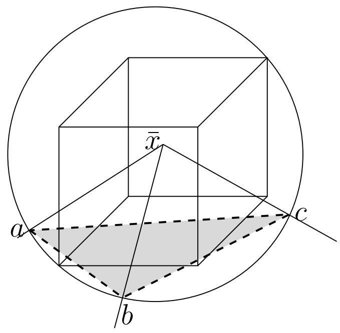

Figure 2.3: Intersection cut

Following the above idea, we propose a set of customized valid cuts for the HVRP. The main difference is that we do not define the hypersphere, instead we use trivial bounds of the binary variables. For given customer i and customer j, a valid cut can be derived using three variables: u_i, x_{ij}^k and x_{ji}^k. In Figure 2.4 for the sake of presentation let us consider only two variables (u_i and x_{ij}^k). Let A, B and C be three constraints in "$\leq$" format where their intersections with $x_{ij}^k = 0$ and $x_{ij}^k = 1$ are a_t, b_t and c_t for $t = 0, 1$, respectively. Any cut connecting p_0 to p_1 (where $p = a,\ b,\ c$) is a valid cut for the feasible region. In this simple example the most efficient cut is the cut that passes through c_0 and a_1.

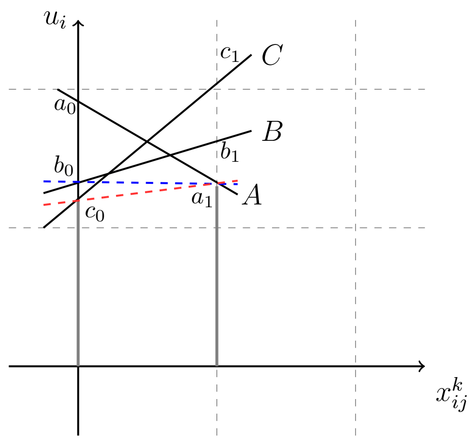

Figure 2.4: Customized cuts

Similarly, we can derive valid cuts for the HVRP. Let $f_t(u_i, x_{ij}^k, x_{ji}^k)$ be a function resulting from a valid inequality by moving every term to the left hand side and dropping the inequality sign. These three functions can be simply obtained $p_t^1 = f_t(u_i, x_{ij}^k = 0, x_{ji}^k = 1)$, $p_t^2 = f_t(u_i, x_{ij}^k = 1, x_{ji}^k = 0)$ and $p_t^3 = f_t(u_i, x_{ij}^k = 0, x_{ji}^k = 0)$. Any triple of $(p_t^1, p_{t'}^2, p_{t''}^3)$ presents a valid cut. To identify the cuts, we can simply find the planes which pass through these three points by calculating the following determinant:

$$\begin{vmatrix} x_{ij}^k & x_{ji}^k - 1 & u_i - p_t^1 \\ x_{ij}^k - 1 & x_{ji}^k & u_i - p_{t'}^2 \\ x_{ij}^k & x_{ji}^k & u_i - p_{t''}^3 \end{vmatrix} = (x_{ij}^k + x_{ji}^k - 1)(u_i - p_{t''}^3) - x_{ji}^k(u_i - p_t^1). \qquad (2.35)$$

The resulting cut will contain non-linear terms. The non-linear terms can be linearized using (2.30) and using the fact that $x_{ij}^k x_{ji}^k = 0$ and $(x_e^k)^2 = x_e^k$ (recall that $x_e^k \in \{0, 1\}$).

2.5 Extending the model

In this section, we present the modifications necessary to generalize the model introduced in this chapter so far to the multi-depot HVRP (MD-HVRP) and the capacitated multi-depot HVRP (CMD-HVRP). Let V_d be the set of depots in which a set of vehicles ($K_j,\ \forall j \in V_d$) are stationed. Each vehicle must return to the depot from which it started its trip. Since by extending the problem to the MD-HVRP, no extra limitation will be added to the problem, there is no need to add or remove any constraints. The only modification required is to consider all vertices and edges for $\delta(i)$.

However, when there is a capacity limitation of the depots, additional constraints are required to guarantee that the capacity limitations are not violated. This type of constraints can be formulated in two ways. Let F_i be the capacity of depot $i \in V_d$. And let a_{ij} be a binary variable taking value one if customer i is assigned to depot j. Then the depot capacity constraints can be presented by

$$\sum_{e\in\delta(i)} x_e^k + \sum_{e\in\delta(j)} x_e^k - a_{ij} \leq 1,\ \forall i \in V_c,\ \forall j \in V_d,\ \forall k \in K_j, \tag{2.36}$$

$$\sum_{i\in V_c} q_i a_{ij} \leq F_j,\ \forall j \in V_d. \tag{2.37}$$

Constraint (2.36) states if customer i is assigned to depot j, then a_{ij} takes value one. Constraint (2.37) guarantees the total demands assigned to depot j is less than the depot's capacity. Here, we present another set of constraints using the load variables (u_i) to impose the depots' capacity limitation. The idea is to make sure that $u_{i^*} \leq F_j$ where i^* is the last customer on a route which has been assigned to depot j. The following constraint represents the depots' restrictions.

$$\sum_{k\in K_j} \sum_{a\in\delta^+(j)} u_i x_e^k \leq F_j, \quad \forall j \in V_d. \tag{2.38}$$

The above constraint is a non-linear constraint. It can be linearized using new continuous variables:

$$v_{ij} \leq u_i,\ \forall i \in V_c,\ \forall j \in V_d, \tag{2.39}$$

$$v_{ij} \geq u_i - M\left(1 - \sum_{k\in K_j} x_e^k\right),\ \forall e = (i,j):\ i \in V_c,\ j \in V_d, \tag{2.40}$$

$$v_{ij} \leq M \sum_{k\in K_j} x_e^k,\ \forall e = (i,j):\ i \in V_c,\ j \in V_d, \tag{2.41}$$

$$\sum_{i\in V_c} v_{ij} \leq F_j, \quad \forall j \in V_c. \tag{2.42}$$

where M is a big number and $v_{ij} \geq 0,\ \forall i \in V_c$ and $\forall j \in V_d$ is an auxiliary variable.

2.6 Concluding remarks

We studied the polyhedral presentation of the deterministic HVRP which addresses the first research question to some extend. In fact, we studied different types of valid inequalities and lifting techniques which lead to developing a new formulation for the HVRP whose uncertain counterpart is tractable and at the same time provide a relatively tight approximation for the convex hull of the deterministic HVRP.

Among the valid inequalities and lifting techniques we studied in this chapter, the capacity inequalities, customized valid inequalities and the reformulation-linearization technique provide a significant improvement on the lower bound of the deterministic HVRP.

In addition, we extended our models for the multi-depot HVRP and the capacitated multi depot HVRP.

Chapter 3

The HVRPUD WITHOUT RECOURSE ACTIONS

3.1 Introduction

As briefly explained in Section 1.2, within the a priori policy, two main approaches have been suggested to formulate the VRPUD and to capture its uncertainty: single-stage and two-stage approaches. In this chapter, we present single-stage models for the Heterogeneous Vehicle Routing Problem with Uncertain Demand (HVRPUD) in which no recourse action is considered in the models. That is, if a route failure occurs, the solution obtained from a single-stage approach does not provide any (optimal) action and subsequently does not provide any information on possible extra cost even though there are parameters to control the probability of failure. Therefore, drivers/managers decide upon suitable recourse actions once a failure occurs to minimize the cost. In order to reduce the risk of encountering failures, managers may prefer to increase the validity of routes subject to the demand uncertainty. However, this risk reduction usually involves an increase in cost. Therefore, managers may consider a tradeoff between the route validity and the extra cost. To formulate this situation, *Chance Constrained Programming* (CCP) and/or *Robust Optimization* (RO) may be used to capture the uncertainty in the VRPUD. These methods provide a set of routes which are guaranteed to be valid with a high probability or to be immune/protected against demand variations. In this chapter, we present models based on CCP and two different approaches within RO to the HVRPUD. But, let us first present a brief literature review on the single-stage VRPUD and its variants.

Literature review There are several studies carried out on single-stage CVRP with uncertain demand in the literature. The most recent surveys on the VRPUD are Gendreau et al. [38], Dror [29] and Erera et al. [34]. The first results on the VRPUD dates back to the early 1960s with Tillman [76]. In the 1980s SVRP received more attention with Stewart and Golden [74], Dror and Trudeau [31], Laporte and Louveau [46] and Laporte et al. [47]. Stewart and Golden [74] was one of the earliest works to solve the single-stage VRPSD to optimality. There, they formulated the HVRP with stochastic demand using chance constrained programming and showed that the VRPSD is convertable into a tractable equivalent deterministic problem for some random demand distribution. Also, they presented two models in which recourse costs are considered. Dror and Trudeau [31] extended Clarke and Wright [24] heuristic which was originally developed to solve the deterministic VRP, and solved stochastic VRP. Laporte et al. [47] later study a location-routing problem with stochastic demand in which they investigated two main models. In the first model, they use CCP to formulate the uncertainty and in the second model, similar to [74], they make sure that the expected penalty of route failure does not exceed a pre-specified fraction of the route length. Laporte et al. [48] studied a relevant problem where instead of demands, traveling time is subject to uncertainty. They also apply chance constrained programming to formulate the problem. After this work, researchers have mostly focused on two-stage SVRP, the VRP with stochastic demand and recourse cost. It is due to the fact that although chance constrained programming provides a suitable framework to formulate and solve the VRPUD, it does not provide any information on possible extra cost of route failure even though it controls the probability of failure. In the literature, it is mainly assumed that all customers have to be served, thus the recourse action defined to fulfil this assumption is back-and-forth trips to the depot to serve remaining customers on failed routes. CCP ignores the location of the failure, therefore corresponding recourse costs cannot be taken into account. A set of recourse routes can have quite different costs depending where failures occur.

Three solutions can be suggested to cope with this drawback. Firstly, a set of recourse actions and their corresponding costs can be included into modelling phase, that results in two-stage Stochastic Vehicle Routing Problem (SVRP) or also known as SVRP with recourse costs. This solution will be reviewed and studied in Chapter 6. The second solution is to analyze the risk level using simulation. In this case, the risk level (α) can be considered as a variable in the model. This assumption results in a much more complicated and intractable problem. Shen [69] studies such a situation where probability of constraint violation (α) in CCP is considered as a new variable. She investigates special cases in linear programming in which the resulting problems are easier to deal with although they are still difficult to solve. In this dissertation, we provide a scenario-based analysis to identify the best risk level scenario for two actions in Chapter 7. The first (recourse) action is that a return trip has to be made to the depot for a replenishment and the pre-planned route will have to be resumed as all customers are required to be served. For a given set of routes obtained from solving the single-stage VRPUD, we can compute and analyze the actual cost of serving all customers (the routing cost plus the cost of return trips) for each risk level scenario. The second action is that we relax the assumption that all costumers have to be served. It is quite often in practice that unserved costumers are left unserved and a lost sale cost is imposed. In Chapter 7 for a given set of routes obtained from solving the single-stage VRPUD for each risk level scenario, we computationally analyze the optimal intervals of the lost sale cost for each risk level scenario. The third solution which has recently received more attention is to apply robust optimization to formulate the problem and uncertainty sets. As mentioned, RO considers the worst case possible for the uncertain parameters. The goal is to find routes that are feasible for all demand (scenario) realizations, so that failure can *never occur*. Literature is rather scarce on this topic and we are only aware of a recent study by Sungur et al. [75], who use the robust optimization methodology introduced by Ben-Tal and Nemirovski [15] to formulate the Robust CVRP (RCVRP). As their method is known to be conservative, later on, Bertsimas and Sim [17] propose an adjustable RO approach where using a control parameter we can adjust the probability of violation of constraints. To best of our knowledge there is no work on Bertsimas and Sim's approach for the VRPUD. As there is a control parameter in their approach, similar to CCP, we carry out the risk level analysis for this approach as well. In the remainder of this section we briefly describe these three approaches (CCP, Ben-Tal and Nemirovski and Bertsimas and Sim RO approaches).

Chance Constrained Programming: The chance constrained programming was developed by Charnes and Cooper in the fifties and early sixties ([20],[21]). In CCP, for given parameters of random variables, such as distributions with their means and variances, one subjectively specifies a control probability for a constraint not to incur a violation. Simply, the constraint-wise CCP for a single constraint $\tilde{a}_i x \leq \tilde{b}_i$ can be presented by

$$\Pr\left(\tilde{a}_i x \leq \tilde{b}_i\right) \geq 1 - \alpha_i, \tag{3.1}$$

where $\tilde{a}_i \in \Re^n$ and $\tilde{b}_i \in \Re$ are uncertain parameters with known distribution functions and α_i is the pre-specified probability. Constraint (3.1) guarantees that constraint i will be valid $(1 - \alpha_i)\%$ times. Another type of CCP is chance-constrained programming with joint constraints where it is guaranteed that the union of n constraints is satisfied with a pre-specified probability $1 - \alpha$:

$$\Pr\left(\cap_{i=1}^{n}\{\tilde{a}_i x \leq \tilde{b}_i\}\right) \geq 1 - \alpha. \tag{3.2}$$

There are two difficulties that make CCP intractable. Firstly, it is very difficult to check the feasibility of a given solution. Secondly, the feasible region induced by chance constraints may be non-linear and non-convex ([2] and [45]). However, for some special cases it has been proven that the feasible region is convex or convex approximations can be proposed. For more detail see [58]. Another drawback with CCP is that it requires access to *reliable data* such as the parameters' distribution which is not always possible.

Robust optimization: This approach overcomes the two difficulties of CCP (access to reliable data and intractability) i.e., there is no need to have access to any distribution parameters and also the resulting problems are tractable if the original problem is tractable. However, RO may lead to a very conservative approach. The constraint-wise RO for a single constraint $\tilde{a}_i x \leq \tilde{b}_i$ is derived from solving the following problem:

$$\max_a ax \leq \min_b b \tag{3.3}$$

Many researchers have tried to find a tractable representation of the above inequality for different types of uncertainty sets. Soyster [73] propose a conservative approach to formulate data uncertainty. His work later was developed and extended by many people most notably Ben-Tal, Bertsimas, El-Ghaoui, Nemirovski and Sim. Soyster's approach provides a full protection against any data variation. In Soyster [73], an uncertain parameter is modelled as symmetric and bounded variable which takes values in an interval. Then, he proposes a tractable counterpart for (3.3).

Ben-Tal and Nemirovski [16] propose a less conservative approach. They consider a slightly different uncertainty set than the one considered by Soyster i.e., they cut the corners of the uncertainty set which is not very likely to happen. They also propose a general case where uncertain parameters are formulated in a cone. However, their resulting robust counterpart of (3.3) is a non-linear constraint, more precisely a second-order cone constraint when the original problem is a linear program. Ben-Tal and Nemiroveski later relax the assumption that the constraints are *hard*, so that they permit some constraints to be violated. These constraints are called *soft* constraints and are immunized/protected against uncertainty in a more flexible way. The recent robust counterpart is known as *generalized robust counterpart* where a parameter (α) known as "global sensitivity" and a distance between uncertain parameters' normal range ($\mathcal{Z}$) set and their physically possible set ($\mathcal{Z}_+ \supset \mathcal{Z}$) adjust the flexibility of (3.3) as follows:

$$\max_{a_i} a_i x - \min_{b_i} b_i \leq \alpha \text{dist}(\zeta, \mathcal{Z}), \ \forall \zeta \in \mathcal{Z}_+ \tag{3.4}$$

Ben-Tal et al. [14] present a comprehensive study on this issue and related topics. As mentioned the resulting problems are non-linear and difficult to solve, in particular when there are integer variables in the original model. With binary variables, Ben-Tal and Nemirovski [16]'s approach tends to be even more conservative so much so that sometimes the resulting problem becomes infeasible.

Bertsimas and Sim [17] propose a tractable and adjustable robust approach which is also very suitable for mixed integer programs with uncertainty. Their main idea is based on the fact that all uncertain parameters do not always take their worst possible values simultanously. Based on this idea they introduce a new notion: *price* *of robustness* which is associated with a parameter (Γ_i) for constraint i. Indeed, this parameter controls the degree of conservatism of the robust solution which is guaranteed to be feasible when up to Γ_i of the parameters simultaneously take their worst values for a given constraint. This parameter controls the trade-off between the probability of violation of constraints and the value of objective function. They assume that uncertain parameters are independently and symmetrically distributed in intervals $[a_{ij} - \hat{a}_{ij}, a_{ij} + \hat{a}_{ij}]$. For a given Γ_i, the following robust counterpart is formulated

$$a_i x + \max_{\{\Phi_i \cup t_i:\ \Phi_i \subseteq J_i, |\Phi_i| = \lfloor \Gamma_i \rfloor,\ t_i \in J_i \setminus \Phi_i\}} \sum_{j \in \Phi_i} \hat{a}_{ij} x_j + (\Gamma_i - \lfloor \Gamma_i \rfloor) \hat{a}_{t_i} x_{t_i} \leq b_i, \tag{3.5}$$

where $\lfloor \Gamma_i \rfloor$ parameters are permitted to take their worst possible values and one parameter (indexed by t_i) change by $(\Gamma_i - \lfloor \Gamma_i \rfloor)\hat{a}_{t_i}$, and J_i is the set of coefficients subject to uncertainty. Finally Φ_i is a sub-set of uncertain parameters. Note that without loss of generality, it is assumed that b_i is deterministic. Constraint (3.5) can be linearized in a tractable way using duality theorems. Bertsimas and Sim also introduce *probability bounds* depending on the value of Γ_i and represent the probability of violation of a constraint if more than Γ_i coefficients change at the same time. They show that the larger the number of uncertain coefficients in a constraint is, the more accurate the bounds are.

In the next three sections, we apply CCP, Ben-Tal and Nemirovski and Bertsimas and Sim RO approaches to the HVRPUD.

3.2 Chance-constrained model

In a chance-constrained model, constraints are required to be satisfied with some big probability. We start with the MTZ constraints (2.7–2.8), whose chance-constrained counterpart is as follows:

$$\Pr\left[u_j - u_i - Q_m \sum_{k \in K} x_e^k + Q_m \geq q_j\right] \geq 1 - \alpha, \quad e = (i, j) \in E_c, \tag{3.6a}$$

$$\Pr\left[q_i \leq u_i \leq \sum_{k \in K} Q_k \sum_{e \in \delta^+(i)} x_e^k\right] \geq 1 - \alpha, \quad i \in V_c, \tag{3.6b}$$

which mean that these constraints can be violated with probability at most α. In particular, given a cumulative distribution $\mathcal{F}_j$ for the demand parameter q_j, the above are equivalent to:

$$u_j - u_i - Q_m \sum_{k \in K} x_e^k + Q_m \geq \mathcal{F}_j^{-1}(1 - \alpha), \quad e = (i, j) \in E_c, \tag{3.7a}$$

$$\mathcal{F}_i^{-1}(1 - \alpha) \leq u_i \leq \sum_{k \in K} Q_k \sum_{e \in \delta^+(i)} x_e^k, \quad i \in V_c. \tag{3.7b}$$

Note that the chance-constrained counterpart (3.7a–3.7b) remains linear.

Remark: It is worth mentioning that constraints (3.7a–3.7b) guarantee that there is no sub-tour for $(1 - \alpha)100\%$ realisations of the demands. When we solve the problem, in the solution there will not be any sub-tour because the solution is guaranteed to be valid for $(1 - \alpha)100\%$ cases. Like validity of any constraints in the CCP context, these constraints may be violated with probability of α.

The chance-constrained counterpart of the capacity inequalities (2.10) involves *non-linear* constraints. To ease notation we let $x_i^k = \sum_{e \in \delta^+(i)} x_e^k$ and let x^k denote the (column) vector $(x_i^k : i \in V_c)$. The chance-constrained counterpart can be written as follows:

$$\Pr[q^T x^k \leq Q_k] \geq 1 - \alpha, \ k \in K. \tag{3.8}$$

When q follows a normal distribution $\mathcal{N}(\mu, \Lambda)$ with mean (vector) μ and covariance (matrix) Λ, the above chance constraint can be reformulated as the following second-order cone constraint:

$$\sqrt{(x^k)^T \Lambda x^k} \leq \frac{Q_k - \mu^T x^k}{\Phi^{-1}(1-\alpha)}, \ k \in K, \tag{3.9}$$

where Φ is the cumulative distribution function of the standard normal distribution. When demands are not correlated (i.e., $\lambda_{ij} = 0, \ i \neq j \in V_c$), we can rewrite (3.9) as:

$$\mu^T x^k + \Phi^{-1}(1-\alpha) \sqrt{\sum_{i \in V_c} \lambda_i^2 (x_i^k)^2} \leq Q_k, \ k \in K. \tag{3.10}$$

To obtain a linear formulation we can substitute the non-linear term on the left-hand side with the linear over-estimator $\Phi^{-1}(1-\alpha) \sum_{i \in V_c} \lambda_i x_i^k$, obtaining an approximated (linear) chance constraint (λ_i is the demand standard deviation for costumer i).

Next let us consider the chance-constrained counterpart of the sub-tour elimination inequalities (2.13), which is as follows:

$$\Pr\left[X(S:\bar{S}) \geq 2\left\lceil q(S)/Q_k \right\rceil\right] \geq 1-\alpha,\ k \in K,\ S \subseteq V_c. \tag{3.11}$$

If $\mathcal{F}_{q(S)}$ is the joint distribution function of the random variables $q_i,\ i \in S$, then the above is equivalent to:

$$X(S:\bar{S}) \geq 2\left\lceil \mathcal{F}^{-1}_{q(S)}(1-\alpha)/Q_k \right\rceil,\ k \in K,\ S \subseteq V_c, \tag{3.12}$$

where $\mathcal{F}^{-1}_{q(S)}(1-\alpha)$ can be calculated for some classes of distribution functions (e.g., *Normal*), when demands are independently distributed and follow the same distribution with different parameters. For example, when $q(S) \sim \mathcal{N}(\mu_S, \Lambda)$, where $\mu_S = \sum_{i \in S} \mu_i$ is the sum of the means and Λ is the covariance matrix, then we have a tractable case and (3.12) can be replaced by

$$X(S:\bar{S}) \geq 2\left\lceil q^*(S)/Q_k \right\rceil,\ k \in K,\ S \subseteq V_c,$$

where $q^*(S)$ is calculated as follows:

$$\Pr\left[q(S) \geq q^*(S)\right] = \Pr\left[\frac{q(S)-\mu_S}{\sqrt{|\Lambda|}} \geq \frac{q^*(S)-\mu_S}{\sqrt{|\Lambda|}}\right], \tag{3.13}$$

and

$$q^*(S) = \mu_S + \Phi^{-1}(1-\alpha)\sqrt{|\Lambda|}. \tag{3.14}$$

Similar to constraint (3.9), the chance-constrained counterpart of the lifted inequalities (2.23) also involves non-linear constraints.

$$\Pr\left[-u_j + u_i + Q_m x_{ij} + (Q_m - q_j - q_i)x_{ji} \leq Q_m - q_j\right] \geq 1 - \alpha,\ \ i, j \in V_c, \quad (3.15)$$

Again let assume q follow a normal distribution $\mathcal{N}(\mu, \Lambda)$, then the above chance constraint is equivalent to:

$$\sqrt{(x_{ji})^2\lambda_i^2 + (1 - x_{ji})^2\lambda_j^2} \leq \frac{F(u,x)}{\Phi^{-1}(1-\alpha)},\ \ i, j \in V_c, \quad (3.16)$$

where $F(u,x) = u_j - u_i + Q_m(1 - x_{ij}) - (Q_m - \mu_j - \mu_i)x_{ji} - \mu_j$. Note that $(1 - x_{ji})x_{ji} = 0$. Similar to (3.10) we can approximate (3.16) as follows:

$$\begin{aligned} -u_j + u_i + Q_m x_{ij} + \quad (Q_m & - \mu_j - \mu_i)x_{ji} \leq Q_m - \mu_j \\ & - \Phi^{-1}(1-\alpha)\left((x_{ji})\lambda_i + (1 - x_{ji})\lambda_j\right) \quad (3.17) \end{aligned}$$

The chance-constraint counterpart of the RLT inequalities of Section 2.4.2 retains linearity, since there is only one random variable which appears as a coefficient of one or more decision variables. In this case, we can apply the same idea used for the MTZ constraints. For example, considering the chance-constrained counterpart of the RLT inequalities (2.33b), we get

$$\Pr\left[u_j \geq z_{ij} + y_{ji} + q_j(1 - x_{ij} - x_{ji})\right] \geq 1 - \alpha, \quad (3.18)$$

which is equivalent to

$$u_j \geq z_{ij} + y_{ji} + \mathcal{F}_j^{-1}(1-\alpha)(1 - x_{ij} - x_{ji}). \tag{3.19}$$

The chance constraint counterpart of the depot capacity constraints presented in (2.37) can be formulated similar to (3.9):

$$\Pr\left[\sum_{i \in V_c} q_i a_{ij} \leq F_j\right] \geq 1 - \alpha, \ \forall j \in V_d \tag{3.20}$$

when $q \sim \mathcal{N}(\mu, \Lambda)$, the above chance constraint can be reformulated by

$$\sqrt{(a_j)^T \Lambda a_j} \leq \frac{F_j - \mu^T a_j}{\Phi^{-1}(1-\alpha)}, \ j \in V_d, \tag{3.21}$$

where a_j is the vector $a_{ij} : i \in V_c$.

3.3 Ben-Tal and Nemirovski robust model

In the Ben-Tal and Nemirovski (BN) model, the uncertain demand vector q belongs to a bounded uncertainty set U, which is constructed as a set of deviations around a fixed *expected value* q^0. In the following, we let s denote the number of *(demand) scenario vectors*: q^1, ..., q^s. The uncertainty set U consists of linear combinations of the scenario vectors with *weights* $\xi \in \Xi$:

$$U = \left\{ q \in \Re^n : \ q = q^0 + \sum_{l=1}^{s} \xi_l q^l, \ \xi \in \Xi \right\}. \tag{3.22}$$

In particular, we consider two uncertainty sets for Ξ:

$$\Xi_1 = \{\xi \in \Re^s : \ \|\xi\|_\infty \leq 1\}, \tag{3.23a}$$

$$\Xi_2 = \{\xi \in \Re^s : \ \|\xi\|_2 \leq \rho\}, \tag{3.23b}$$

which represent, respectively, a *box* and a *ball of radius* ρ. In this section, we present the robust counterparts for the above two sets and show that our formulation mainly results in linear robust counterparts for both sets.

Note that in the model of Section (2.2), only the right-hand side of the MTZ constraints (2.7–2.8) is subject to (demand) uncertainty. For such a case and the case where the left-hand side of each constraint contains only *one* coefficient of uncertainty, Sungur et al. [75] prove that the BN robust counterpart can be obtained simply by substituting q_j $(j = 1 \ldots n)$ with

$$q_j^0 + \textstyle\sum_{l=1}^s |q_j^l|, \tag{3.24a}$$

$$q_j^0 + \rho\sqrt{\textstyle\sum_{l=1}^s (q_j^l)^2}, \tag{3.24b}$$

for Ξ_1 (3.23a) and Ξ_2 (3.23b), respectively. Therefore, the BN robust counterpart of (2.7–2.8) retains the same structure, since only the right-hand side changes.

On the other hand, this is not true for all the inequalities presented in Chapter 2. In fact, while the box uncertainty set (3.23a) always retains linearity, the ball uncertainty set (3.23b) may lead to *conic quadratic* inequalities when the demand uncertainty is not restricted to the right-hand side of the constraints.

First, we consider the *capacity* inequalities (2.10). The BN robust counterpart corresponding to the box uncertainty set (3.23a) is the inequalities:

$$\sum_{i \in V_c} \sum_{a \in \delta^+(i)} q_i^0 x_e^k + \sum_{i \in V_c} \sum_{e \in \delta^+(i)} \sum_{l=1}^{s} |q_i^l| x_e^k \leq Q_k, \;\; k \in K, \tag{3.25}$$

whereas the BN-robust counterpart corresponding to the ball uncertainty set (3.23b) is a set of conic quadratic inequalities as follows:

$$\sum_{i \in V_c} \sum_{e \in \delta^+(i)} q_i^0 x_e^k + \rho \sqrt{\sum_{i \in V_c} \sum_{e \in \delta^+(i)} \sum_{l=1}^{s} (|q_i^l| x_e^k)^2} \leq Q_k, \;\; k \in K, \tag{3.26}$$

Now we consider the sub-tour elimination inequalities (2.13). Here, only the right-hand side is subject to uncertainty. To construct the BN robust counterpart, it suffices to substitute q_j with (3.24a) for Ξ_1 and (3.24b) for Ξ_2, respectively.

Next constraint to consider is the lifted inequalities (2.23), which leads to conic quadratic inequalities for the ball uncertainty set (3.23b), whereas for the box uncertainty set (3.23a) the BN-robust counterpart is:

$$\begin{aligned} -u_j + u_i + Q_m x_{ij} + (Q_m - q_j^0 - q_i^0) x_{ji} \\ + \sum_{l=1}^{s} \left| (-q_j^l - q_i^l) x_{ji} + q_j^l \right| \leq Q_m - \quad q_j^0 \quad , \; (i,j) \in E_c. \end{aligned} \tag{3.27}$$

The robust counterpart of the DL lifted inequalities for the ball uncertainty set

(3.23b) is

$$
\begin{aligned}
-u_j + u_i + Q_m x_{ij} + (Q_m - q_j^0 - q_i^0)x_{ji} & \\
+\rho\sqrt{\sum_{l=1}^{s}(q_j^l(1-x_{ji}))^2 + (q_i^l x_{ji})^2} & \leq \quad Q_m - q_j^0,\ (i,j) \in E_c \qquad (3.28)
\end{aligned}
$$

where E_c is the set of edges between only customers.

The robust counterpart of the RLT inequalities of Section (2.4.2) can be obtained similar to the MTZ constraint. These always retain linearity since there is only one uncertain (demand) parameter in each inequality, either in the right-hand side or in the left-hand side. So the BN-robust counterpart for Ξ_1 (3.23a) and Ξ_2 (3.23b) can again be obtained by substituting q_j with (3.24a) and (3.24b), respectively.

Finally, the robust counterpart of the depot capacity constraint for the box uncertainty set can be written as follows which retain their linearity.

$$\textstyle\sum_{i\in V_c} q_i^0 a_{ij} + \sum_{i\in V_c}\sum_{l=1}^{s}|q_i^l| a_{ij} \leq F_j,\ j \in V_d. \qquad (3.29)$$

However, similar with the capacity inequalities (2.10), the robust counterparts of the depot capacity constraints for the ball uncertainty set is conic quadratic inequalities:

$$\textstyle\sum_{i\in V_c} q_i^0 a_{ij} + \rho\sqrt{\sum_{i\in V_c}\sum_{l=1}^{s}(q_i^l a_{ij})^2} \leq F_j,\ j \in V_d. \qquad (3.30)$$

3.4 Bertsimas and Sim robust model

As explained, the robust counterpart developed by Bertsimas and Sim (BS) has two main features: It contains in each constraint a parameter Γ (the protection level) that controls the *degree of conservatism* of the robust solution; it is computationally tractable if the original problem is tractable. Regarding tractability, Bertsimas and Sim give a compact robust counterpart of a given nominal model by introducing a polynomial number of new variables and constraints. We will apply a similar approach and use the (strengthening) inequalities presented in the previous chapter.

According to BS-model of uncertainty set U, the uncertain demand vector q takes value of the interval $[q^0 - \hat{q}, q^0 + \hat{q}]$, symmetric around the nominal value q^0. The parameter Γ mentioned above denotes the maximum number of coefficients that are allowed to change simultaneously with respect to their nominal values in each constraint. In particular, at most $\lfloor \Gamma \rfloor$ q_is will change to their bounds $\hat{q}_j$s and one will change by $(\Gamma - \lfloor \Gamma \rfloor)$ portion of its bound.

Since the capacity inequalities (2.10) are more general types of inequalities we have, we will implement the Bertsimas and Sim method in more detail for them. To construct the BS-robust counterpart we denote, for each given $k \in K$, by $\Psi^k \subseteq V_c$ the subset corresponding to those coefficients q_i that are subject to uncertainty and by Γ^k the control parameter for the constraint. Recall (2.11) where $q_i \in [q^0 - \hat{q}, q^0 + \hat{q}]$

$$\sum_{i \in V_c} \sum_{e \in \delta^+(i)} q_i x_e^k \leq Q_k, \quad \forall k \in K \tag{3.31}$$

Following the Bertsimas and Sim idea of the robust solution, we would like to find a robust solution such that if up to $\lfloor\Gamma^k\rfloor$ parameters change, a set of routes is deterministically feasible and even if more than $\lfloor\Gamma^k\rfloor$ changes occur, then the robust solution will be feasible with very high probability. For this purpose let us consider the k-th constraint of the capacity inequality and reformulate it as follows.

$$\sum_{i\in V_c}\sum_{e\in\delta^+(i)} q_i x_e^k +$$
$$\max_{\{S^k\cup t_k:\ S^k\subseteq\Psi^k,|S^k|=\lfloor\Gamma^k\rfloor,t_k\in\Psi^k\setminus S^k\}} \sum_{i\in S^k}\sum_{e\in\delta^+(i)} \hat{q}_i w_e^k + (\Gamma^k - \lfloor\Gamma^k\rfloor)\hat{q_{t_k}} \sum_{e\in\delta^+(t_k)} w_e^k$$
$$\leq Q_k, \quad \forall k \in K \tag{3.32a}$$
$$-w_e^k \leq x_e^k \leq w_e^k,\ \forall e, \forall k \tag{3.32b}$$
$$w_e^k \geq 0 \tag{3.32c}$$

where $w_e^k \geq 0$ is a new variable and $t_k \in V_c$ is a customer's index.

To linearize the above nonlinear constraint we first solve the maximization problem within the constraint.

$$\max_{\{S^k\cup t_k:\ S^k\subseteq\Psi^k,|S^k|=\lfloor\Gamma^k\rfloor,t_k\in\Psi^k\setminus S^k\}} \sum_{i\in S^k}\sum_{e\in\delta^+(i)} \hat{q}_i y_e^k + (\Gamma^k - \lfloor\Gamma^k\rfloor)\hat{q_{t_k}} \sum_{e\in\delta^+(t_k)} y_e^k \tag{3.33a}$$

For a given x the above problem can be represented as follows:

$$\max \quad \sum_{i\in\Psi^k}\sum_{a\in\delta^+(i)} \hat{q}_i (x_e^k)^* \mu_{ik} \tag{3.34a}$$
$$\text{s. t.} \quad \sum_{i\in\Psi^k} \mu_{ik} \leq \Gamma^k \tag{3.34b}$$
$$0 \leq \mu_{ik} \leq 1,\ \forall i \in \Psi^k \tag{3.34c}$$

By strong duality, the optimal solution of the dual problem of (3.34) is feasible for (3.32). So, we can replace the non-linear term of (3.33) by the dual form of (3.34). Following Bertsimas & Sim construction, we obtain the following BS-robust counterpart with additional dual variables p_i^k and π^k:

$$\textstyle\sum_{i\in V_c} q_i^0 \sum_{e\in\delta^+(i)} x_e^k + \sum_{i\in\Psi^k} p_i^k + \Gamma^k\pi^k \le Q_k,\ \ k\in K \tag{3.35a}$$

$$\textstyle\pi^k + p_i^k \ge \hat{q}_i \sum_{e\in\delta^+(i)} x_e^k,\ \ i\in\Psi^k, k\in K \tag{3.35b}$$

$$\pi^k \ge 0,\ \ k\in K \tag{3.35c}$$

$$p_i^k \ge 0,\ \ i\in\Psi^k, k\in K. \tag{3.35d}$$

Next consider the sub-tour elimination inequalities (2.13), where the uncertainty only appears on the right-hand side of the constraints. For the constraint corresponding to $S\subseteq V_c$, Ψ^S denotes the sub-set of V_c that corresponds to those q_is that are subject to uncertainty and Γ^S the control parameter for the constraint. Clearly, in this case, we can simply sort $\hat{q}_i$ in non-increasing order and choose the first Γ^S demands where $\lfloor\Gamma^S\rfloor$ can change up to their bounds and the last of the selected demands can only change by $(\Gamma^S - \lfloor\Gamma^S\rfloor)$ portion of its bound.

Note that for the MTZ constraints (2.7–2.8), there is only one demand parameter in each constraint. Hence, the BS-robust counterpart can be simply obtained by substituting q_j with the quantity $q_j^0 + \Gamma\hat{q}_j$, where $0\le\Gamma\le 1$.

For the DL lifted inequalities (2.23), the BS construction is similar to the one used for the capacity inequalities (2.10) see (3.35a–3.35d).

In each of the RLT inequalities of Section (2.4.2), there is at most one demand coefficient. Hence, the BS-robust counterpart can be obtained by simply substituting q_j with the quantity $q_j^0 + \Gamma \hat{q}_j$ where $0 \leq \Gamma \leq 1$.

Finally, the robust counterpart of the depot capacity constraints (2.10) can be obtained similar with the capacity inequalities (2.10) as follows. Let Ψ^j be the set corresponding to those uncertain demands in j-th constraint and let Γ_j be its corresponding control parameter. The robust counterpart of the depot capacity constraint is formulated as follows:

$$\textstyle\sum_{i\in V_c} q_i^0 a_{ij} + \sum_{i\in\Psi^j} p_i^j + \Gamma^j \pi^j \leq F_j,\ j \in V_d \tag{3.36a}$$

$$\pi^j + p_i^j \geq \hat{q}_i a_{ij},\ i \in \Psi^j, j \in V_d \tag{3.36b}$$

$$\pi^j \geq 0,\ j \in V_d \tag{3.36c}$$

$$p_i^j \geq 0,\ i \in \Psi^j, j \in V_d, \tag{3.36d}$$

where analogous with the capacity inequalities' robust counterpart, we define new variables π and p.

After setting up the robust counterparts, we need to calculate the parameter Γ for each constraint. On the one hand, Γ controls the degree of conservatism of the robust solution, that is *guaranteed* to be feasible *up to* Γ simultaneous changes of the coefficients of a given constraint. On the other hand, Bertsimas and Sim also introduce *probability bounds* depending on the value of Γ and representing the *probability of violation* of a constraint if more than Γ coefficients change at the same time. They show that the larger the number of uncertain coefficients in a constraint is, the more accurate the bounds are. Recall the capacity inequalities, the below bound is the best bound proposed in this work:

$$\Pr\left(\sum_{i\in V_c}\sum_{e\in\delta^+(i)} q_i x_e^k \leq Q_k\right) \leq \mathrm{B}(n,\Gamma^k), \quad \forall k\in K \tag{3.37}$$

where

$$\mathrm{B}(n,\Gamma^k) = \frac{1}{2^n}\left((1-\gamma)\begin{pmatrix} n \\ \lfloor\nu\rfloor \end{pmatrix} + \sum_{l=\lfloor\nu\rfloor+1}^{n}\begin{pmatrix} n \\ l \end{pmatrix}\right) \tag{3.38}$$

where $n = |\Psi^k|$ and $\nu = (\Gamma^k + n)/2$ and $\gamma = \nu - \lfloor\nu\rfloor$.

However, since in many of our inequalities only a few uncertainty coefficients appear, these bounds are not very helpful for deciding the value of Γ. For instance, if we change q_i in the k-th constraint of the MTZ inequalities to its bound, it is expected that the constraint is never violated. But the probability bound provided by Bertsiams and Sim is $\mathrm{B}(1,1) = \frac{1}{2}$ which is a very poor bound.

For this reason, we give the following two propositions that allow us to calculate *exactly* the value of Γ corresponding to a given probability of violation for two specific types of constraints. Note that the concept of probability of violation for a given constraint is strictly related to the chance-constrained models that were previously presented

Proposition 4 applies when *only one* uncertainty coefficient is present as, for example, in the MTZ constraints(2.7–2.8).

Proposition 4. *If q_j ($j \in V_c$) is a uniformly distributed random variable in $[q_j^0 - \hat{q}_j, q_j^0 + \hat{q}_j]$, then any constraint where q_j is its only uncertainty coefficient has a* *probability α of violation for $\Gamma = 1 - 2\alpha$.*

Proof. Since q_j follows a uniform distribution, we can easily calculate the corresponding cumulative distribution function. Hence, by setting $q_j^* = q_j^0 + \hat{q}_j(1 - 2\alpha)$, we can guarantee that $\Pr[q_j \leq q_j^*] \leq \alpha$. Therefore, $\Gamma = 1 - 2\alpha$ provides the desired probability of violation. □

Remark. The above proposition also applies to inequalities (2.33b) as well as to (2.33a) since $1 - x_{ij} - x_{ji} = 0$ or 1 due to the integrality condition.

Proposition 5 applies to sub-tour elimination inequalities (2.13).

Proposition 5. *Given any $S \subset V_c$, if q_j for any $j \in V_c$ is an independently and symmetrically distributed random variable in $[q_j^0 - \hat{q}_j, q_j^0 + \hat{q}_j]$ with cumulative distribution function $\mathcal{F}_j$ and joint distribution $\mathcal{F}_{q(S)}$, then*

$$\Pr\left[x(S : \bar{S}) \geq 2\left\lceil q(S)/Q_k \right\rceil\right] \geq 1 - \alpha,\ k \in K,$$

for Γ computed as follows:

$$\min \Gamma \tag{3.39a}$$

$$s.t. \qquad \textstyle\sum_{i \in S} \xi_i \leq \Gamma \tag{3.39b}$$

$$\textstyle\sum_{i \in S} \hat{q}_i \xi_i = \mathcal{F}_{q(S)}^{-1}(1 - \alpha) \tag{3.39c}$$

$$0 \leq \xi_i \leq 1,\ i \in S. \tag{3.39d}$$

Proof. Since the inverse joint distribution function $\mathcal{F}_{q(S)}^{-1}(1-\alpha)$ can be calculated for some classes of distribution functions (e.g., Normal), the LP (3.39a–3.39d) selects the uncertainty coefficients such that the sum of their deviations gives the desired value and Γ is minimized. $\square$

3.5 Concluding remarks

This chapter addressed the second research question i.e., which methods in stochastic programming and mixed-integer programming can be used to improve VRPUD solution algorithms. For the first time, we formulated the HVRPUD via three single-stage approaches within stochastic optimization: chance constraint programming and Ben-Tal and Nemirovski robust optimization approach and Bertsimas and Sim adjustable robust optimization approach. We developed the formulations for the HVRPUD on the basis of the models introduced in the previous chapter.

The deterministic counterpart of the HVRPUD models are tractable via mixed-integer programming, so standard techniques within mixed-integer programming can be employed to solve the models. We use a branch-and-cut method to solve the proposed models. The separation procedures will be explained in the next chapter and an extensive computational results and experiments will be presented in Chapter 7.

In addition, we proposed two probability bounds to calculate the protection parameter for Bertsimas and Sim adjustable robust optimization approach.

Chapter 4

BRANCH-AND-CUT METHOD FOR HVRPUD

4.1 Introduction

One of the most successful methods for solving a wide range of (mixed) integer programs is the Branch-and-Cut (B&C) method [56]. Indeed, branch-and-cut based methods e.g., Lysgaard et al. [54] were the best solution methods for the CVRP for a long time. Wolsey [78] defines a B&C algorithm as follows. *A B&C algorithm is a branch-and-bound algorithm in which cutting planes are generated throughout the branch-and-bound tree.* A cutting plane is generated when a solution at a node of branch-and-bound tree violates a valid inequality. To find out if a valid inequality is violated we employ some algorithms called *separation algorithms.* According to Applegate et al. [3] a separation algorithm is defined as follows: *A separation algo-*

rithm for a class $\mathcal{C}$ of linear inequalities is an algorithm that, given any x^*, returns either an inequality in $\mathcal{C}$ that is violated by x^* or a failure message. There exist two types of separation algorithms: *exact algorithms* in which a failure message will be returned only if x^* satisfies all valid inequalities, and *heuristic algorithms* where a failure message may be returned even when there still exist some violated valid inequalities in $\mathcal{C}$.

In this chapter, the separation algorithms which identify the violated constraints of the SEC and the comb inequalities introduced in Chapter 2 will be discussed.

4.2 Separation algorithms for SEC

In the literature of the TSP, the VRP and their variants, different separation algorithms have been proposed for the Sub-tour Elimination Constraints (SEC). As the VRP is an extension of the TSP, all exact separation algorithms for the SEC introduced for the TSP are only heuristic algorithms for the VRP and its variants. Usually methods within Graph Theory and Network Optimization are deployed to identify the violated constraints. Bard et al. [12] and Lysgraad et al. [54] review and introduce various types of separation algorithms for the SEC. Recall the SEC as follows for a given LP solution x^* at some node on the branch-and-bound tree:

$$\sum_{i \in S} \sum_{j \in \bar{S}} x^*_{ij} \geq 2\lceil \frac{q(S)}{Q_m} \rceil \tag{4.1}$$

where $S \subset V_c$ and $|S| > 2$, and $\bar{S} = V \setminus S$.

We first briefly review the existing separation algorithms and then a greedy separation algorithm will be presented to separate the SEC.

Shrunk Support Graph In this class of algorithms, the vertices presenting the customers iteratively one by one are shrunk and create a new vertex which is called *supernode* or *supervertex*. The weight assigned to the supervertex is $q(S) = \sum_{i \in S} q_i$ and an edge $\{s, j\}, s \in S,\ j \in \bar{S}$ is given the weight of $\sum_{s \in S} \sum_{j \in \bar{S}} x^*_{sj}$. Since the vertices are shrunk into a supervertex, to make sure that we do not miss any violated constraint because of using this procedure the notion of safe shrinking has been introduced and characterised as follows [54]: *Whenever there is a violated capacity inequality in G, there exists a set of supervertices in the shrunk graph whose union defines a capacity inequality with at least the same violation.* Lysgraad et al. [54] generalise the conditions under which a shrinking is safe as follows.

For separation of the SEC, it is safe to shrink a customer set S if $\sum_{i \in S} \sum_{j \in \bar{S}} x^*_{ij} \leq 2$ *and* $\sum_{i \in R} \sum_{j \in \bar{R}} x^*_{ij} \geq 2$, $\forall R \subset S$.

The next step is to identify a vertex and shrink it into a supervertex. Different strategies have been proposed to answer this question. A greedy heuristic introduced by Lysgraad et al. [54] considers each supervertex as a seed and shrink vertex j to the supervertex that minimizes the slack of the SEC for $S \cup \{j\}$. When we cannot expend S, we select another seed and continue the procedure.

Bard et al. [12] suggest another strategy based on the following equivalent form of the SEC.

$$\sum_{i \in S} \sum_{j \in S} x^*_{ij} \leq |S| - k(S) \tag{4.2}$$

where $S \subset V_c$ and $|S| > 2$, and $k(S)$ is the minimum number of vehicles needed to serve set S.

The weight of each supervertex is equal to the sum of the weights of those edges that connect vertices inside S. The shrinking procedure is shown in Figure 4.1. The shrinking procedure is repeated until the resulting graph consists of one disconnected supervertex or more. Recall that if $x_{ij} = 0$, then there is no edge between i and j which may lead to a disconnected graph. During this procedure, $k(S)$ for each supervertex is calculated and is compared with the weight of the supervertex to identify violated constraints.

Figure 4.1: Shrinking procedure.

Min-Cut problem For a special case when $x(E(S : V_c \setminus S)) < 2$, we can apply algorithms for the the min-cut problem to find violated SECs. For any $S \subset V_c$, if $x(E(S : V_c \setminus S))$ is less that 2, then S violates a constraint of the SEC. Several algorithms with different complexities have been proposed for this heuristic. The best know algorithm is by Frank [35] whose complexity is $O(|V||E|)$. For a more general setting, Lysgaard et al. [54] adopt a heuristics based on the max-flow problem for the so-called fractional capacity inequalities. The fractional capacity inequalities are the sub-tour elimination constraint (4.1) where the right-hand side is replaced by

$q(S)/Q_m$. If a violated fractional capacity inequality is identified, then constraint (4.1) is violated, too. By solving the max-flow problem for $S \subset V_c$, we minimize the slack of the fractional capacity inequalities. Moreover, these heuristic can be run on the shrunk graph obtained from safe shrinking procedures.

A greedy algorithm Motivated by the shrinking procedure, we here introduce a greedy separation algorithm to identify violated SEC. A new approach to find the violated constraints can be based on calculating and ordering a set of values for each customer. Let S be a sub-set of customers, we would like to find vertex $j \in V_c \setminus S$ so that by adding it, new set $S \cup \{j\}$ violates a constraint of the fractional SEC. So, the following integer programme can find the vertex if the optimal solution value is strictly positive.

$$\max \quad \alpha \tag{4.3}$$

$$\text{s.t.} \quad \frac{q(S)}{Q_m}(1 + y_j) - x^*(\delta(S))$$
$$-x^*(E(j : \bar{S}))y_j + x^*(E(S : j))y_j \geq \alpha, \ \forall j \in \bar{S}_c \tag{4.4}$$

$$\sum_{j \in \bar{S}_c} y_j = 1 \tag{4.5}$$

$$y_j \in \{0, 1\}, \ \forall j \in \bar{S}_c \tag{4.6}$$

where constraint (4.4) evaluates the slack if customer $j \in V_c \setminus S$ is contracted into set S. Also, y_j is a binary variable which takes value one if j is added to S, and zero if otherwise. If $\alpha > 0$, then $S \cup \{j\}$ violates a constraint. But if for all $j \in V_c \setminus S$, the objective function is non-positive ($\alpha \leq 0$), then it is better to contract the vertex with highest value into S. The larger the objective function is, the more likely it will be to have violated constraints in next iterations. The above problem identifies only one vertex to be added to set S, but it can identify k vertex by simple changes.

The above problem can be reduced to a simple ordering as follows. Let $f = \frac{q(S)}{Q_m} - x^*(\delta(S))$ which is a fixed value for all customers in one iteration and, let $q(j) = \frac{q_j}{Q_m}y_j - x^*(E(j : \bar{S}))y_j + x^*(E(S : j))y_j$ for $j \in V_c \setminus S$. Now we can sort $f + q(j)$ for $j \in \bar{S}_c$ in a descending order. If $f + q(j) > 0$, then violates the SEC, hence we add the corresponding customer to set S and the corresponding constraint to the problem. The process will be repeated until there is no strictly positive value for $f + q(j)$. Then as mentioned the vertex with highest $f + q(j)$ will be added to S and run the algorithm again until $S = V_c$. In each iteration of the B&C method, we run the above heuristic 10 times selecting randomly a vertex as the seed (S).

4.3 Separation algorithm for comb inequalities

Several methods have been proposed to identify violated comb inequalities on the support graph for the TSP and these works have been later adapted to the VRP (see [3],[12],[40],[51],[54] and [62]). As mentioned in Chapter 2, the blossom valid inequality is a special case of the comb inequality where each tooth can contain only two vertices (one in common with the handle and one outside of the handle). There are also a few algorithms for separating this class of valid inequalities (see [3] and [50]). Recall the comb valid inequalities for a LP solution vector x^*:

$$x^*(E(H)) + \sum_{t=1}^{s} x^*(E(T_t)) \leq |H| + \sum_{t=1}^{s}(|T_t| - 1) - \lceil \frac{1}{2}s \rceil. \tag{4.7}$$

as mentioned in Chapter 2, the comb inequalities can be stated in the below form:

$$x^*(\delta(H)) + \sum_{t=1}^{s} x^*(\delta(T_t)) \geq 3|T| + 1 \tag{4.8}$$

We first briefly explain two existing algorithms and then a new separation algorithm will be proposed.

Connected component method This method identifies bi-connected components of the ϵ-support graph. A ϵ-support graph is obtained by deleting edges with weights less than ϵ or greater than $1 - \epsilon$. Let us start this heuristics with two definitions we need later.

Definition 5. *k-connected graph: A graph is k-connected if k vertices (along with their adjacent arcs) must be removed to disconnect the graph.*

Two equivalent definitions of the above definition for the bi-connected graph are as follows. A connected graph is bi-connected if there are two paths between each two vertices or a connected graph is biconnected if the removal of any single vertex (and all edges incident on that vertex) cannot disconnect the graph.

Definition 6. *Articulation points (cut vertex): any vertex whose removal (together with removal of any incident edges) results in a disconnected graph.*

To identify candidate handle and teeth, the ϵ-support graph is first constructed, then each bi-connected component with at least three vertices is a candidate handle. The teeth are comprised of any vertices on the original support graph which are connected to the vertices in the bi-connected components. To find bi-connected components of the ϵ-support graph, the Depth First Search (DFS) algorithm can be applied (see [25] for the DFS algorithm).

Shrinking method The above procedure provides us configurations in the support graph to set up the blossom or 2-matching inequalities. Combining algorithms developed for blossom inequalities with shrinking procedures can lead to effective algorithms for general comb inequalities on the original graph. For instance, if there is a path P from vertex s to vertex t such that $x_e = 1$ for $e \in E(P)$ (where $E(P)$ is the set of edges on path P), then we can shrink path P and replace the whole path with an edge having weight $x_{st} = 1$ ([3]). Several types of shrinking procedures have been suggested based on this simple idea by Padberg and Grotschel [60], Grotschel and Holland [40] and Padberg and Rinaldi [61]. Here we present Grotschel and Holland's procedure as their procedure has been reported to be one of the most successful procedures and also easy in implementation ([3] and [50]).

Grotschel and Holland [40] propose five rules for shrinking as follows:

1. Given a path P of 1-edges (i.e. $x_e = 1$ for $e \in E(P)$) between s and t, replace P with a single edge $(s - t)$ with $x_{st} = 1$.

2. Given $\{v, u, w\} \subseteq V$ with $x_{u,v} = 1$ and $x(\{u, v\}\{w\})$, then shrink u and v. See Figure 4.2.

3. Given $S \subseteq V$ so that $|S| = 4$ and $x(\delta(S)) = 2$, then shrink S to a single vertex.

4. Given $\{s, t, u, v\} \subseteq V$ so that $x_{st} = x_{vu} = 1$ and $x(\{s, t\}\{u, v\}) = 1$, then shrink $\{s, t\}$ to $\{u, v\}$. See Figure 4.3.

5. Given $\{u, v, w\} \subseteq V$ so that $x_{uv} = 1$ and $x(\{s, t\}\{u, v\}) \geq 0.5$, then shrink $\{u, v\}$.

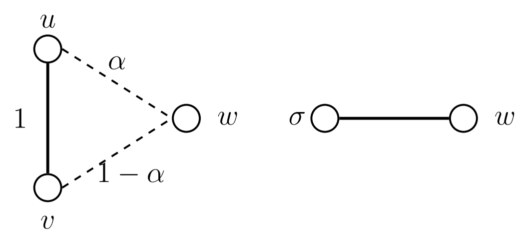

Figure 4.2: 1-edge shrinking in a triangle

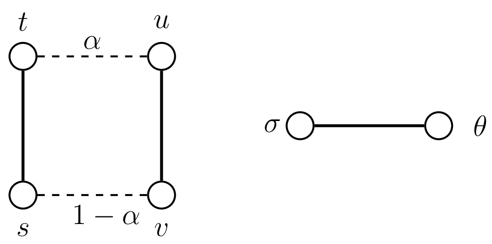

Figure 4.3: 1-edge shrinking in a square

A greedy approach In this section, we propose a new greedy procedure to identify violated comb inequalities for the HVRP. The main idea of this procedure is to find out which configuration of a given set of vertices violates a valid inequality or is most likely to violate a valid inequality. Similar to our proposed heuristic for the SEC, this heuristic is also motivated by shrinking procedure. In the proposed method, shrinking process is guided in the sense that even if it does not identify any violated inequality in an iteration, it forms a setting of a comb which is more likely to be violated in next steps. In standard shrinking procedures, as mentioned, we need to have 1-edges, whereas in our proposed method, there is no need to necessarily have 1-edges.

The heuristic starts with an initial comb. An initial comb configuration can be set up using the connected component method or algorithms developed for blossoms. Let C be the set of vertices forming the comb. Then, we would like to find out moving a given vertex $i \in V_c \setminus C$ to which sub-set ($H \setminus T$, $T \setminus H$ or $T \cap H$) can lead to a violated valid inequality. If no violated constraint is found in this step, the vertex is added to a subset which is most likely to lead to a violated inequality in next steps. The procedure can be explained with a simple example.

Example Let assume the following initial comb configuration is given and let assume that the weight of each edge is one (see Figure 4.4):

$$H = \{2, 4, 6\},\ T_1 = \{1, 2\},\ T_2 = \{3, 4\},\ T_3 = \{5, 6\},\ V_c \setminus C = \{7\}.$$

We would like to find out moving vertex 7 to which sub-set leads to the highest reduction in the left-hand side(lhs) of (4.8). The smaller the lhs is, the more likely it

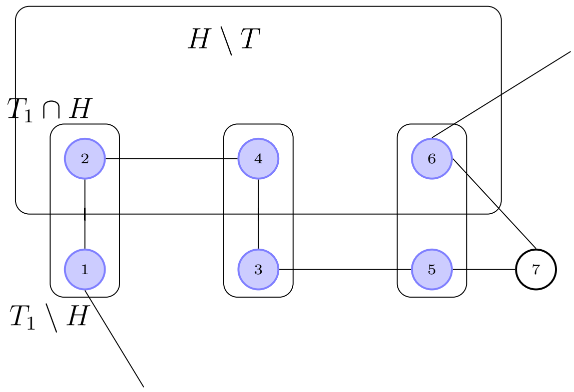

Figure 4.4: A comb consists of a handle and three teeth

is to lead to a violated inequality. Vertex 7 can be moved to either of following sets:

$$H \setminus T,\ T_1 \setminus H,\ T_2 \setminus H,\ T_3 \setminus H,\ T_1 \cap H,\ T_2 \cap H \text{ or } T_3 \cap H$$

For instance, if the vertex is moved to $T_3 \setminus H$, the value of the comb will be reduced by 2, because the edge (6-7) now intersects only one sub-set border and the edge (5-7) does not intersect any sub-set border any more.

4.4 Concluding remarks

We studied several separation procedures for the SEC and comb inequalities for the HVRPUD and proposed two greedy algorithms for these two types of inequalities. This study addresses the second research question. Indeed, we employ methods within mixed-integer programming to solve the HVRPUD.

The computational results suggest that while the proposed separation algorithm for the SEC is relatively efficient, the comb inequalities do not do anything good for our problem. In addition to the proposed algorithm, we implemented and tried the other separation algorithms in the literature but no improvement was achieved. This is in contrast with performance of the comb inequalities for the CVRP. But as the problem complexity increases, the performance of the comb valid inequalities significantly reduces. Belenguer et al. [13] report the similar performance for the comb inequalities.

Chapter 5

CVRPUD WITHOUT RECOURSE ACTION: COLUMN GENERATION

5.1 Introduction

In Chapter 3, we presented three single-stage approaches: chance constrained programming, BS robust approach and BN robust approach, and applied them to the heterogeneous vehicle routing problem with uncertain demand (HVRPUD). These approaches have been mainly applied to (mixed) integer problems with uncertainty within branch-and-cut algorithms. Although branch-and-cut methods are very successful in solving deterministic VRPs, in recent years column-generation based methods are reported to perform better in some specific problems. In particular, using column generation based methods, Baldacci and Mingozzi [10] and Fukasawa et al [36] solve instances of the CVRP that had never been solved by branch-and-cut methods. Despite this fact, there are very few works on solving the VRPUD to optimality. These works focus on solving stochastic vehicle routing problems with recourse cost ([22]). However, there are few issues with these works which will be explained in the next chapter where we study stochastic vehicle routing problems with recourse cost. To the best of our knowledge there is no work applying column generation based methods to the VRPUD without recourse cost. In this chapter we first present an overview of the column generation based methods for the deterministic vehicle routing problem. Then, we implement the three single-stage approaches presented in Chapter 3 to the VRPUD within the column generation based method's framework.

5.2 Overview of column generation methods for CVRP

The first work on solving deterministic VRPs via column-generation based methods dates back to early 1960s with Balinski and Quandt [11]. Since then, there has been a considerable attention on solving different variants of the VRP using column-generation based methods, most notably Agarwal et al [1] and Hadjiconstantinou et al. [42]. These attempts were not as successful as branch-and-cut algorithms until Fukasawa et al [36] which was a major breakthrough in solving the CVRP. They propose a branch-and-cut-and-price framework which enables them to solve some instances which were unsolved at that time. In addition to their very efficient implementation, the key element of their success is to combine the branch-and-cut approach with column generation methods originated from a *q-route approach.* Later, Beldacci et al [8] and Beldacci and Mingozzi [10] propose another framework to solve variants of the VRP mainly concentrating on the Lagrangean relaxation and finding feasible routes more efficiently.

In order to present a general framework based on a column-generation method, we first need to reformulate the CVRP into a set-covering formulation. From now on we study capacitated vehicle routing problems with homogenous vehicles, hence we define the problem as follows. Let $G = (V_0, E)$ be a directed graph with vertices $V_0 = \{0, 1, ..., n\}$ and edges $(i, j) = e \in E$ where $i, j \in V_0$. Vertex 0 is the depot, and $i \in V = V_0 \backslash \{0\}$ represents a customer with an associated positive random demand q_i. Each edge $e \in E$ has a non-negative length c_e. Recall the set-partitioning formulation presented in Chapter 1.

$$
\begin{aligned}
(\text{SP}): \quad \mathcal{Z}(\text{P}) = \min \quad & \sum_{r \in \mathcal{R}} f_r z_r && (5.1)\\
\text{s.t.} \quad & \sum_{r \in \mathcal{R}} z_r \leq m, && (5.2)\\
& \sum_{r \in \mathcal{R}(i)} z_r \geq 1, \ \forall i \in V && (5.3)\\
& z_r \in \{0, 1\}, \ \forall r \in \mathcal{R} && (5.4)
\end{aligned}
$$

where $\mathcal{R}$ is the index set of all feasible routes. Let $\mathcal{V}(r) = \{r_0 = 0, r_1, ..., r_{n_r}, r_{n_r+1} = 0\}$ be the set of vertices on route $r \in \mathcal{R}$. Then, a feasible route is defined as follows: route $r \in \mathcal{R}$ is feasible if $q(\mathcal{V}(r)) = \sum_{i \in \mathcal{V}(r)} q_i \leq Q$. Let $\mathcal{R}(i)$ be all routes that contain vertex $i \in V$ i.e., $\mathcal{R}(i) = \{r \in \mathcal{R} : i \in \mathcal{V}(r)\}$, and f_r be the cost of route

$r \in \mathcal{R}$. The decision variable z_r is a binary variable which takes value one if route r is chosen in the solution, and zero otherwise. In the literature, constraint (5.3) is also represented by

$$\sum_{r\in\mathcal{R}}\sum_{e\in\delta(i)} a_r^e z_r \geq 1, \ \forall i \in V \tag{5.5}$$

where a_r^e is the number of times route r visits customer i. There are other presentations of the set-partitioning formulation of the CVRP in the literature. Due to computational issues which will be explained in Chapter 7, we use the following presentation. The Edge based Set-Partitioning model (SP_E) for the CVRP is reformulated by (a) introducing a new integer variable x_e for edge $e \in E$, and (b) converting z_r to a continuous variable. Moreover, let $\mathcal{E}(r) \subset E$ be the set of edges visited by route r and let $\mathcal{R}(e) = \{r \in \mathcal{R} | e \in \mathcal{E}(r)\}$ be the set of routes which visit edge $e \in E$. Then, we add constraint $-x_e + \sum_{r\in\mathcal{R}(e)} z_r \leq 0, \ \forall e \in E$ to connect z_r to x_e. Fukasawa et al. [36] provide an interesting connection between the standard CVRP formulation and the set-partitioning formulation using $-x_e+\sum_{r\in\mathcal{R}(e)} z_r \leq 0, \ \forall e \in E$ and refer to as the Dantzig-Wolfe Master problem (DWM). Let SP_E be formulated by

$$
\begin{aligned}
(\mathrm{SP}_E): \quad \mathcal{Z}(\mathrm{P}_E) = \min \quad & \sum_{e \in E} c_e x_e && (5.6) \\
\text{s.t.} \quad & \sum_{r \in \mathcal{R}(i)} z_r \geq 1, \ \forall i \in V, && (5.7) \\
& -x_e + \sum_{r \in \mathcal{R}(e)} z_r \leq 0, \ \forall e \in E, && (5.8) \\
& x(\delta(i)) = 2, \ \forall i \in V, && (5.9) \\
& z_r \in [0,1], \ \forall r \in \mathcal{R}, && (5.10) \\
& x_e \in \{0,1,2\}, \ \forall e \in E. && (5.11)
\end{aligned}
$$

where $\delta(S)$ is the cut-set defined by S: $\delta(S) = \{(i,j) \in E | i \in S \ \& \ j \notin S \text{ or } i \notin S \ \& \ j \in S\}$.

As (SP) and (SP_E) suggest, there are exponentially many possible routes and consequently, exponentially many variables of type z_r. Identifying all possible routes at the beginning is impractical and also unnecessary. There are two main successful approaches to handle this difficulty. Firstly, Fukasawa et al. [36] and Pessoa et al. [63] suggest to set up the initial problem with an initial sub-set of routes ($\bar{\mathcal{R}} \subset \mathcal{R}$) and solve its corresponding LP-relaxation. Then, using the dual variables and heuristic methods, they identify those routes which can improve in each iteration and add them to the initial problem. Once no improving routes are found, they branch on integer variables and repeat the algorithm until there is no improving route and the integrality condition is satisfied for all integer variables. They invoke cuts in each node of the branch-and-bound tree when a constraint is violated. They run separation procedures for framed capacity, strengthened comb, multistar, partial multistar, generalized multistar and hypotour contraints. Their approach is known as *branch-and-cut-and-price algorithm.* In the second approach which was proposed by Baldacci et al [9] and Baldacci and Mingozzi [10], they use a series of bounding procedures to find near optimal solutions for the LP-relaxation of the problem. They use *q-route approach* and combine it with the Lagrangian relaxation. Then, a set of routes whose *reduced costs* are smaller than the gap between upper and lower bounds generated through their method are found and added to the LP-relaxation of the problem. Finally, the resulting problem is solved using an integer programming solver. Although both the approaches are capable of solving the same problems and present the same performance, we follow Fukasawa et al.'s method as their method is more flexible and suitable for the framework we propose for the VRPUD in this chapter and the next chapter.

Let the initial problem with an initial set of routes be $\overline{\mathrm{SP}_E}$. As mentioned, the next step is to identify feasible routes that improve the current solution. To do so, a set of routes called *q-routes* are identified. As it is difficult to find feasible routes, we relax one of the condition of feasible routes i.e., in a route a vertex may be visited more than once. This type of routes are called q-routes ([36]). Let α_i, β_e and π_i be the dual variables corresponding to constraints (5.7), (5.8) and (5.9), respectively. The LP-relaxation of $\overline{\mathrm{SP}_E}$ is denoted by ($\overline{\mathrm{LPSP}_E}$). Let $(\bar{\mathbf{z}}, \bar{\mathbf{x}})$ be the optimal solution to $\overline{\mathrm{LPSP}_E}$. Using the dual problem, we can assess if the current solution is optimum for $\overline{\mathrm{LPSP}_E}$. The dual of $\overline{\mathrm{LPSP}_E}$ is

$$\text{D-}\overline{\mathrm{LPSP}}_E: \quad \max \quad \sum_{i \in V} \alpha_i + 2\pi_i \tag{5.12}$$

$$\text{s.t.} \quad \beta_{(i,j)} + \pi_i \leq c_{(i,j)}, \ \forall (i,j) = e \in E \tag{5.13}$$

$$\sum_{i \in \mathcal{V}(r)} \alpha_i - \sum_{e \in \mathcal{E}(r)} \beta_e \leq 0, \ \forall r \in \mathcal{R} \tag{5.14}$$

In the above problem if $(\bar{\mathbf{z}}, \bar{\mathbf{x}})$ satisfies all constraints then, the current solution is optimum for the dual problem and as a result for $\overline{\mathrm{LPSP}_E}$. The idea of the pricing problem is to form a set of routes which improve the current solutions. Constraint (5.12), however, does not help us to form any constraints. Therefore, we focus on constraint (5.14) by which we can form new constraints improving the current solutions.

The current solution is infeasible if constraint (5.14) is violated. Note that each constraint (5.14) corresponds to a route. The procedure of identifying routes which violate (5.14) is called *the column-generation problem* (CG). In fact, the CG problem identifies feasible routes $r \in \mathcal{R}$ which are violated for a given solution $(\bar{\mathbf{z}}, \bar{\mathbf{x}})$. This problem can be formulated as:

$$\text{CG:} \ \ \pi = \min\{\pi_r = \textstyle\sum_{e \in \mathcal{E}(r)} \beta_e - \sum_{i \in \mathcal{V}(r)} \alpha_i | \ \sum_{i \in \mathcal{V}(r)} q_i \leq Q\} \tag{5.15}$$

Several approaches have been suggested to deal with the CG problem. Bramel and Simchi-Levi [19] review some popular methods. As mentioned, Fukasawa et al. [36] on the basis of the previous works developed an efficient method for solving the

CG using the q-route notion. Here we briefly explain their method. The reduced cost of an edge e is calculated by

$$\bar{c}_e = \begin{cases} \beta_e - \frac{1}{2}\alpha_j & \text{for} \quad i = 0 \ \& \ j \neq 0 \\ \beta_e - \frac{1}{2}\alpha_i - \frac{1}{2}\alpha_j & \text{for} \quad i \neq 0 \ \& \ j \neq 0 \\ \beta_e - \frac{1}{2}\alpha_i & \text{for} \quad i \neq 0 \ \& \ j = 0 \end{cases} \tag{5.16}$$

where $e = (i, j) \in E$. Due to the existence of negative cycles, finding q-routes is known to be NP-hard but it is doable in pseudo-polynomial time. Using a data structure and dynamic programming they find q-routes with negative reduced costs. Let M be a $Q \times n$ matrix whose entities $M(q, v)$ represent the least costly walk that reaches vertex $v \in V$ using a total of demands exactly q. Each entity contains a label consisting of a vertex (v), the reduced cost of the shortest q-route ($\bar{c}(M(q, v))$) and a pointer to the previous vertex on the q-route. All entities are initialized with an empty q-route and an infinite cost. Then, the contents of each entity will be updated using dynamic programming by extending the walk to its neighbours as follows. If w is a neighbour of v and if $\bar{c}(M(q, v)) + \bar{c}_{(v,w)} < \bar{c}(M(q + q_w, w))$, then $M(q + q_w, w)$ will be updated: the new reduced cost is $\bar{c}(M(q, v)) + \bar{c}_{(v,w)}$ and the predecessor is v.

In matrix M, at most n q-routes with negative reduced cost will be identified. As there are nQ entities and each one is processed in $O(n)$, therefore, the total running time is $O(Qn^2)$. Since the reduced cost can be negative, having cycles is very likely. All algorithms developed to find the shortest path on a graph are valid when there are no negative cycles. Eliminating these negative cycles is a strongly NP-hard problem. Fukasawa et al. propose to look for s-cycle-free q-routes where

s is small. In Chapter 7, we discuss more on this issue and other tricks to avoid negative cycles as much as possible.

5.3 Column generation method for CVRP with uncertain demands

In this section, we apply CCP, BN and BS robust approaches to the CVRPUD without recourse actions within the aforementioned branch-and-price framework. Regarding the uncertainty set of customers' demand, similar to Chapter 3, we consider two types of uncertainty sets (box and ellipsoid) for Ben-Tal and Nemirovski's approach and intervals for Bertsimas and Sim's approach. For the CVRP with chance constraints, in addition to probability distribution functions we can consider scenarios for uncertain parameters (data-driven chance constraints). In the following sub-sections we explain these approaches within the branch-and-price framework in more detail.

5.3.1 CVRP with chance constraints

As explained, in the set-partitioning formulation, we look for feasible routes identified by the CG problem that minimize the total cost. Recall the definition of a feasible route: route r is feasible if it begins from the depot, visits a set of customers ($\mathcal{V}(r)$) at most once and returns to the depot while it maintains $\sum_{i\in\mathcal{V}(r)} q_i \leq Q$. In CCP, the last condition of the route feasibility changes as follows:

$$\Pr\left[\sum_{i\in\mathcal{V}(r)} q_i \le Q\right] \ge 1-\epsilon, \quad r\in\mathcal{R}. \tag{5.17}$$

In the CG problem, it is sufficient to make sure that for a route, condition (5.17) is held. As there is no decision variable in (5.17), holding the condition in each step is not very difficult but it requires to have access to all the vertices on a path for each entity of matrix M. In matrix M defined in the previous section, we store a pointer to the previous vertex on the route and therefore, we can retrieve the list of vertices on the route. Then, we can check condition (5.17) when we extend the walk to neighbours of vertices given that we have access to the joint probability distribution of demands. When only the historical data (a set of scenarios) is available, we assess the condition as follows. Let S be the set of scenarios available for the uncertain demands and p_s be the corresponding probability of scenario s. Also, let g_s be a binary indicator which takes value one if $\sum_{i\in\mathcal{V}(r)} q_i^s \le Q$ is held for scenario s, and zero otherwise i.e.,

$$g_s = \begin{cases} 1 & \text{for} \quad \sum_{i\in\mathcal{V}(r)} q_i^s \le Q \\ 0 & \text{otherwise.} \end{cases} \tag{5.18}$$

Let $\mathcal{W}_v$ be the least costly walk ending at vertex v. Then, at entity $M(q,v)$ if we extend the walk to vertex w, we need to make sure that the extended walk ($\mathcal{W}_w$) satisfies $\sum_{s\in S} p_s g_s \ge 1-\epsilon$ where $g_s = 1$ if $\sum_{i\in\mathcal{W}_w} q_i^s \le Q$. Note that in $M(q,v)$, $q = \sum_{i\in\mathcal{W}(r)} q_i^0$ for $q_i^0 = \mathbb{E}[q_i]$.

5.3.2 CVRP with BN RO

In this section, we apply Ben-Tal and Nemirovski's approach to the CVRP with uncertain demands within the branch-and-price framework. Similar to the CVRP with chance constraints, here we focus on the feasibility of routes. The difference is that the last condition of route feasibility changes as we consider the worst cases of demands. The condition on the vehicle's capacity is re-stated by $\max \sum_{i \in \mathcal{V}(r)} q_i \leq Q$. Recall the uncertainty sets from Chapter 3 as follows. The uncertainty set U consists of linear combinations of the scenario vectors with *weights* $\xi \in \Xi$:

$$U = \left\{ q \in \Re^n : \; q = q^0 + \sum_{l=1}^{|S|} \xi_l q^l, \; \xi \in \Xi \right\}. \tag{5.19}$$

In particular, we consider two uncertainty sets for Ξ:

$$\Xi_1 = \{\xi \in \Re^s : \; \|\xi\|_\infty \leq 1\}, \tag{5.20a}$$

$$\Xi_2 = \{\xi \in \Re^s : \; \|\xi\|_2 \leq \rho\}, \tag{5.20b}$$

As the domain of demands is positive, the robust counterpart of the above condition is

$$\sum_{i \in \mathcal{V}(r)} q_i^0 + \sum_{i \in \mathcal{V}(r)} \sum_{l=1}^{|S|} q_i^l \leq Q, \quad r \in \mathcal{R}. \tag{5.21}$$

For each demand if an interval is defined ($q_i \in [\underline{q}_i, \bar{q}_i]$), then we can simply replace q_i with its upper bound and assess the following condition in our CG problem:

$$\sum_{i \in \mathcal{V}(r)} \bar{q}_i \leq Q, \quad r \in \mathcal{R}. \tag{5.22}$$

As constraint (5.22) suggests, we do not need to retrieve the list of vertices on a route and re-calculate any probability at each iteration for each entity of matrix M. In the CG problem, it is sufficient to consider the upper bound of the demands and extend walks based on the upper bounds.

5.3.3 CVRP with BS RO

In this section, we adopt Bertsimas and Sim's robust optimization approach to the CVRP with uncertain demand. Therefore, we can sort $\hat{q}_i$ in non-increasing order and choose the first $\Gamma^{\mathcal{V}_r}$ in the list where $\lfloor \Gamma^{\mathcal{V}_r} \rfloor$ can change up to their bounds and the last of the selected demands can only change by $(\Gamma^{\mathcal{V}_r} - \lfloor \Gamma^{\mathcal{V}_r} \rfloor)$ of its bound. To compute $\Gamma^{\mathcal{V}_r}$, we can use Proposition 5 in Chapter 3 or the bounds defined in Bertsimas and Sim [17].

To implement the above condition within the branch-and-price algorithm, similar to CCP, we need to re-evaluate the condition at each step for each entity of matrix M to make sure the feasiblity condition is held. Let $\mathcal{W}_v$ be the set of vertices on the least costly walk starting from the depot and ending at vertex v. Then, the following condition must be held when we extend the walk to vertex w:

$$\sum_{i \in \mathcal{W}_w} q_i^0 + \sum_{i \in \Psi_w} \hat{q}_i + (\Gamma^{\mathcal{V}_r} - \lfloor \Gamma^{\mathcal{V}_r} \rfloor)\hat{q}_k \leq Q, \tag{5.23}$$

where Ψ_w is the set of vertices consisting of $\lfloor \Gamma^{\mathcal{V}_r} \rfloor$ first vertices of the non-increasing order of sorted $\hat{q}_i$ and $\hat{q}_k$ is the next vertex in this list.

5.4 Concluding remarks

In this chapter, we first studied general issues on column-generation based methods for the deterministic CVRP. For first time, we applied the three single-stage approaches of stochastic optimization to the CVRPUD within column generation based methods.

We defined and studied the master and the pricing problems to formulate the CVRPUD. The definition of feasible routes is critical in defining the pricing problem. Using the definition of feasible routes, we developed the pricing problem for each approach.

In Chapter 7 we present the computational results comparing the models developed in this chapter.

Chapter 6

The VRPSD WITH RECOURSE ACTION

6.1 Introduction

As explained in Chapter 1, different types of recourse actions have been proposed within different policies. Due to its advantages, among the existing policies, we chose to study the *a priori policy* in which the routing is pre-planned and the replenishment is reactive. In this chapter, we address the models of the VRPUD with recourse actions within the a priori policy in greater detail. The stochastic vehicle routing problem with recourse action is one of the most well-studied variants of the vehicle routing problems with uncertain parameters.

A common recourse action modelled within this policy is that if a vehicle fails to serve a customer on its pre-planned route, then it must make a return trip for a replenishment and resume the pre-planned route. This recourse action is known as *traditional recourse action*. In addition, other recourse actions have been suggested in the literature among which we here briefly describe four more popular ones. The first one is *preventive action*. In the preventive action, some strategic points are defined on the planned route to make the return trips before a failure actually occurs. For example, the closer the vehicle gets to the depot, the higher the chance of failure is. The second recourse action is that when a failure occurs, the route for the remaining customers are re-optimized. The third one is described as follows. When a vehicle makes a return trip to the depot, it usually can serve more customers than only those on the failed pre-planned route. So a new route can be planned in this case.

Finally, Dror and Trudeau [31] propose another type of recourse actions. When a failure occurs, the remaining customers on the failed route will have to be served by a series of single customer trips.

Despite the diversity of recourse actions, due to difficulty of modelling, the traditional recourse action has received more attention. The traditional recourse action has been implemented within different frameworks and in different forms. Laporte et al. [48] and Gendreau et al. [37] are the first to use the idea of optimality and feasibility cuts from the two-stage stochastic programming to model the traditional recourse action within the integer L-shaped method. Following Laporte et al.'s method, Hjorring and Holt [44] introduce a new framework on the basis of using partial routes to generate tighter approximation for optimality cuts. They improve the previous works by approximating the expected cost and adding cuts dynamically. However, all attempts by that time were to solve single vehicle SVRPs. Most notable study after Hjorring and Holt is Laporte et al. [49] wherein they use Hjorring and Holt optimality cuts and solve large problems with more than one vehicle. They solve the SVRP for two different types of distribution functions for the customers' demand: Normal and Poisson. Later, this stream becomes the dominant stream in solving different variants of SVRPs. But the difficulty with this framework is that the SVRP's polyhedral is very complex. Hence, several methods within MIP are borrowed to solve SVRP more efficiently. For instance, Rei et al. [66] use local branching to solve the single vehicle routing problem with stochastic demands.

On the other hand, due to the performance of column generation based methods for the deterministic CVRP, set-partitioning based formulations and column-generation based methods have become a new promising stream in modelling and solving the SVRP. Novoa et al. [59] is one of the earliest works studying the SVRP within this framework. They model the SVRP using a set-partitioning formulation as a two-stage stochastic program. In addition to the traditional recourse action, they suggest another recourse action, namely *extended recourse action*. In the extended recourse action, vehicles that have completed their routes with available capacity can serve additional customers from failed routes before returning to the depot or when a vehicle returns to the depot for a replenishment due to failure, this vehicle can perform extra trips to serve unserved customers on its own failed route and also other vehicles' failed routes. They use a set of scenarios to capture the uncertainty of demands. As enumerating all feasible routes is computationally expensive, they suggest a heuristic by which they limit their search space to routes with specific sizes. Having tested their route generation process on deterministic CVRP instances, they conclude that their route generation process provides good enough solution although it does not find the optimal solutions for the CVRP. As they do not solve their generated instances for the SVRP to optimality as well, the performance of their results and the comparison of the recourse actions may not be very much reliable. Note that this study remains unpublished.

The most notable work after Novoa et al. is Christiansen and Lysgaard [22]. They propose a set-partitioning formulation for the SVRP. In their work, the total expected cost of a route consists of two elements: the deterministic cost of the route and the expected extra distance traveled due to failures (the expected cost of recourse costs). They use the fact that the probability that the total demand on a path (starting from the depot and ending at vertex i) does not exceed uQ, depends only on the total expected demand on the path, not on the order of the customers. Note that u is an integer. They assume that demands are independently distributed and follow the same distributions. In their study, they consider those distributions which have an accumulative property i.e. if $q_i \sim \Psi$ then $\sum_i q_i \sim \Psi$. They first calculate the expected number of failure ($\text{Fail}(\mu, \sigma^2, i)$) for a path ending at vertex i with given mean and variance (μ and σ^2), then the expected failure cost is calculated as follows:

$$\text{EFC}(\mu, \sigma^2, i) = 2c_{i0}\text{Fail}(\mu, \sigma^2, i). \tag{6.1}$$

Therefore, a path is charactrized with three arguments μ, σ^2 and i. They use a data structure of 3-dimensional matrix based on these arguments to find shortest paths. However, in their computational study, they assume that demands follow

Poisson distribution. This results in reducing their data structure to a 2-dimensional matrix as the mean and the variance of Poisson distribution are equal. Using the algorithm described above, they solve 19 test problems out of 40 to optimality with the largest being a test problem with 60 customers and 16 routes. As we will explain later, due to dependency of the probability of failures on the vertices on paths, domination of a path over another path is not as simple as the domination of a path in the deterministic CVRP. Not taking into account appropriate domination rules may lead to non-optimal solutions even if the optimality gap is zero.

Christiansen et al. [23] study a set partitioning formulation for the Capacitated Arc Routing Problem with Stochastic Demands (CARPSD) when the demands follow Poisson distributions. They solve the CARPSD via a branch-and-price algorithm for the CARPSD. The capacitated arc routing problem is defined on a network in which demands are associated with edges rather than vertices. This problem can be transferred into the standard CVRP. In this work, a sub-set of edges have stochastic demands which follows Poisson distribution. The rest of edges have no demands and may or may not be visited. Similarly to Christiansen and Lysgaard [22], they consider two types of costs: traveling cost and expected failure cost. They adapt Christiansen and Lysgaard's method to the CARPSD.

6.2 Problem setting and models

In this section, we state the problem setting and present a set-partitioning formulation for the vehicle routing problem with stochastic demands and recourse costs. We assume that demands are revealed on the vehicle arrival. Therefore, it is possible that the actual demands which are realized on vehicles arrival exceed the vehicle capacity. In the pervious chapter, we studied the case wherein we do not consider any action or cost upon failures. Instead, we formulated the capacitated vehicle routing problem with uncertain demand (the CVRPUD) so that the system (the CVRP) is valid with a high probability. In this chapter, we formulate and embed a recourse action in the CVRP model such that the total expected cost is minimized. We consider the traditional recourse action i.e., the vehicle returns to the depot for a replenishment when a failure occurs and then resumes the pre-planned route. The cost of the return trip is usually the penalty considered for the traditional recourse action. The output of the aforementioned approach is a set of routes which is guaranteed to have the minimum expected cost covering the routing cost and the recourse cost.

We define the vehicle routing problem on a graph as follows. Let $G = (V_0, E)$ be a complete directed graph where $V_0 = \{0, .., n\}$ is the set of vertices. Vertex 0 represents the depot and other vertices ($V = \{1, .., n\}$) represent customers. The cost of traveling from vertices i to j is $c_e \geq 0$ where $e = ij$. A homogeneous fleet of m vehicles with capacity Q is available at the depot. Customer i is assigned uncertain demand q_i ($q_0 = 0$) with $\mathbb{E}[q_i] = \bar{q}_i$ and $\mathrm{Var}[q_i] = \sigma_i^2$ such that $\Pr(q_i \leq 0, q_i \geq Q) = 0$. Let $\mathcal{R} = \{1, ..., R\}$ be the index set of all feasible routes. As defined in the previous chapter, route $r \in \mathcal{R}$ charactrized by $\mathcal{V}(r) = \{r_0 = 0, r_1, ..., r_{n_r}, r_{n_r+1} = 0\}$ is feasible if $\bar{q}(\mathcal{V}(r)) = \sum_{i \in \mathcal{V}(r)} \bar{q}_i \leq Q$. Also let f_r be the cost of route $r \in \mathcal{R}$ which consists of two elements: the transportation cost without any failure costs plus the cost of recourse actions in case of failure ($f_r = f_r^1 + f_r^2$). Recall the edge-based set partitioning formulation in the previous chapter. We modify the model as follows:

$$
\begin{aligned}
(\mathrm{SP}_{EC}): \quad \mathcal{Z}(\mathrm{P}_E) = \min \quad & \mathbb{E}\left(\sum_{r\in\mathcal{R}} f_r z_r\right) && (6.2)\\
\text{s.t.} \quad & \sum_{r\in\mathcal{R}(i)} z_r \geq 1, \ \forall i \in V, && (6.3)\\
& -x_e + \sum_{r\in\mathcal{R}(e)} z_r = 0, \ \forall e \in E, && (6.4)\\
& x(\delta(i)) = 2, \ \forall i \in V, && (6.5)\\
& z_r \in [0,1], \ \forall r \in \mathcal{R}, && (6.6)\\
& x_e \in \{0,1,2\}, \ \forall e \in E. && (6.7)
\end{aligned}
$$

The difference between the above model and the model presented in the previous chapter is in the objective function. Here, we minimize the expected total cost.

6.3 Calculating expected costs

The next challenge is to calculate the expected total cost. The cost of traveling from vertices i to j on a given route r can be considered as a statistics and calculated by

$$
\tilde{c}_{ij} = \begin{cases} c_{ij}, & \rho_j \\ c_{ij} + c_{j0} + c_{0j}, & \theta_j \end{cases} \tag{6.8}
$$

When there is no failure to serve vertex j i.e., there is no need to make a return trip to the depot for serving vertex j, the cost of traveling from vertex i to vertex j is equal to c_{ij}. The probability of serving vertex j after vertex i on route r without visiting the depot is ρ_j which can be calculated by

$$\rho_j = \sum_{l=0}^{u} \Pr\left(\sum_{t=0}^{u} q_{r_t} \leq lQ \text{ and } \sum_{t=0}^{u+1} q_{r_t} \leq lQ\right) \tag{6.9}$$

where l is the number of failures before visiting vertex i, and $r_u = i$ and $r_{u+1} = j$. Since $\Pr(q_i \geq Q) = 0$, at most u failures are possible until visiting u-th vertex on route r. Once a failure occurs the vehicle must return to the depot to replenish. This recourse action imposes an extra cost of $c_{j0} + c_{0j}$. Since at most one failure is considered at each edge, the probability of failing to serve vertex j after visiting vertex i is calculated by

$$\theta_j = \sum_{l=0}^{u} \Pr\left(\sum_{t=0}^{u} q_{r_t} \leq lQ \leq \sum_{t=0}^{u+1} q_{r_t}\right) \tag{6.10}$$

Figure 6.1 illustrates the cost of serving a vertex on route r when the vehicle fails to serve vertex e_n in its first visit.

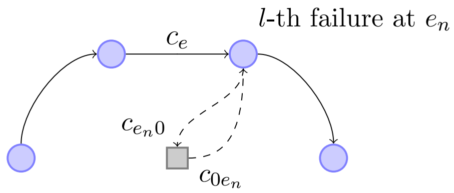

Figure 6.1: An illustrative example for the recourse action

Therefore, the total expected cost of route r is calculated by

$$\mathbb{E}[f_r] = \textstyle\sum_{u=0}^{n_r} \mathbb{E}[\tilde{c}_{r_u r_{u+1}}] \tag{6.11}$$

$$\Rightarrow \mathbb{E}[f_r] = \textstyle\sum_{u=0}^{n_r} \left(\rho_{r_{u+1}}(c_{r_u r_{u+1}}) + \theta_{r_{u+1}}(c_{r_u r_{u+1}} + c_{r_{u+1}0} + c_{0 r_{u+1}})\right) \tag{6.12}$$

$$\Rightarrow \mathbb{E}[f_r] = \textstyle\sum_{u=0}^{n_r} \left(c_{r_u r_{u+1}} + \theta_{r_{u+1}}(c_{r_{u+1}0} + c_{0 r_{u+1}})\right). \tag{6.13}$$

The above expected cost can be calculated when the distribution functions are given. However even when the distribution functions are known, calculating the above expected cost is not always tractable. In many cases, distributions are discretized with a desired accuracy. The output is considered as a set of scenarios. When a set of scenarios S for the demands is given, the above expected cost can be re-stated as follows:

$$\mathbb{E}[f_r] = \sum_{s \in S} p_s f_r^s. \tag{6.14}$$

where f_r^s is the total cost of serving all vertices on route r under scenario s and p_s is the probability assigned to scenario s. For a given scenario s if $\sum_{i \in \mathcal{R}_r} q_i \leq Q$, then $f_r^s = \sum_{e \in E(r)} c_e$, whereas if $\sum_{i \in \mathcal{R}_r} q_i^s > Q$ and $\sum_{i \in \mathcal{V}(r)} q_i^s \leq 2Q$, then $f_r^s = \sum_{e \in E_r} c_e + 2c_{0 e_{n_r}^*}$. Let $e_{n_r}^*$ be the vertex at which failure occurs. For the sake of simplicity of the notation we assume that only one failure occurs on route r but our method can be easily extended to the case when more than one failure occurs.

6.4 Pricing problem

After calculating the expected cost, the next step is to identify those routes which can improve the solution in each iteration. Recall the general steps of the branch-and-price framework in Chapter 5, we write the dual problem of Problem SP_{EC} and set up the column generation problem as follows. Let α_i, β_e be the dual variables corresponding to constraints (6.3) and (6.4) then the dual problem is:

$$\text{DP}: \quad \bar{\mathcal{Z}}(DP) = \max \quad \sum_{i \in V} \alpha_i \tag{6.15}$$

$$\text{s.t.} \quad \sum_{i \in V(r)} \alpha_i - \sum_{e \in \mathcal{E}(r)} \beta_e \leq \bar{f}_r, \; r \in R \tag{6.16}$$

$$\alpha_i \geq 0, i \in V \text{ and } \beta_e \geq 0, \; e \in E. \tag{6.17}$$

where to simplify the notation we denote by $\bar{f}_r = \mathbb{E}[f_r]$. In the above problem, we look for those routes which violate constraint (6.16). To identify them, we define the reduced cost of each edge as follows:

$$\bar{c}_e = \begin{cases} c_{0e_n} + \beta_{0e_n} - \frac{1}{2}\alpha_{e_n} \\ c_{e_s e_n} + \beta_{e_s e_n} - \frac{1}{2}\alpha_{e_s} - \frac{1}{2}\beta_{e_n} \quad \text{when} \quad e_s \neq 0 \quad \text{or} \quad e_n \neq 0 \\ c_{e_s 0} + \beta_{e_s 0} - \frac{1}{2}\beta_{e_s} \end{cases} \tag{6.18}$$

Given the above reduced cost we re-arrange the constraint (6.16) and define the route's reduced cost:

$$\bar{c}_r = \sum_{e \in E(r)} c_e + (c_{0e^*_{n_r}} + c_{e^*_{n_r}0}) \sum_{s \in S^*} p_s + \sum_{e \in \mathcal{E}(r)} \beta_e - \sum_{i \in V(r)} \alpha_i \quad (6.19)$$

$$\bar{c}_r = \sum_{e \in \mathcal{E}(r)} \bar{c}_e + (c_{0e^*_{n_r}} + c_{e^*_{n_r}0}) \sum_{s \in S^*} p_s \quad (6.20)$$

where S^* is the set of scenarios in which failures occur. The next step is to find routes whose reduced cost is negative and then we add them to the master problem. We cannot follow the pricing problem described in the previous chapter as the reduced cost of each edge depends on the fact that using which route we visit an edge and a vertex. To find the shortest route, we need to keep not only the best path and its cost at each vertex but also all paths ending at a vertex and their costs. To clarify this issue we explain it with a simple example. Let us assume there are two paths (P_1 and P_2) ending at vertex i with their corresponding reduced costs ($\bar{c}_1(i)$ and $\bar{c}_2(i)$) such that $\bar{c}_1(i) \leq \bar{c}_2(i)$. We visit vertex j after vertex i. Let us assume that because of the total demands of pervious vertices on path 1, a failure occurs to serve vertex j, so a replenishment trip has to be made to the depot whereas we can serve vertex j using path 2 without any failure. Now at vertex j, due to the failure we had on path 1, $\bar{c}_1(j) \geq \bar{c}_2(j)$. As a conclusion, we cannot eliminate a path with a larger reduced cost as it may turn out to be the cheaper route.

Therefore, we cannot use existing algorithms to find routes with least negative reduced cost. A greedy algorithm is to save all possible paths at each entity of matrix M but it is computationally very expensive and is not practical. As the aim of the pricing problem is to identify routes with negative reduced cost not necessarily least negative reduced cost, we can modify the pricing algorithm described in the previous chapter to identify routes with negative reduced cost. Of course, this modification may lead to not optimal solutions. To make sure we do not miss out any optimal route, if no routes with negative cost was found, then we run another pricing procedure in which we introduce a new eliminating rule. We call this procedure the extensive search. Therefore, we run two different pricing algorithms: the first one is the modified version of the the pricing problem described the previous chapter and will be explained in a greater detail here, and the second one is an extensive search and contains a new eliminating criterion to reduce the original search space and at the same time makes sure we do not miss out the optimal solution.

6.4.1 Modified pricing procedure

The pricing subproblem consists of finding q-routes of minimum reduced cost. This problem is NP-hard when the cost of each edge is independent, however, it can be solved if all demands are integer [36]. As our primary aim is to identify routes with negative reduced cost, here we follow standard procedure (plus some modifications) to identify q-routes without edge cost dependency. The output of this procedure may not be the minimum reduced cost but it is quicker to find a q-route with negative reduced cost.

Similar to matrix M described in Chapter 5, the data structure is a $Q \times n$ matrix M where each element of this matrix $(M(q, v))$ is the least costly walk that reaches vertex v using total demand exactly q. In addition to a reference to the previous vertex, we update the available data in a label by saving all vertices on the path. Given the available data now we can calculate the reduced cost of each path. The matrix is filled using dynamic programming. For each entity of matrix ($M(d, v)$), we extend the walk to its neighbours $w \in V \setminus \{v\}$ so that the total nominal demand of the walk does not exceed Q. $M(q + q_w, w)$ will be updated if $\bar{c}(M(q, v)) + \bar{c}_{v,w} < \bar{c}(M(q + q_w, w))$. In our implementation, once an entity with negative reduced cost is found we terminate the pricing problem and add the corresponding variable/column/route to the master problem.

As explained, due to the negative reduced cost for edges there is a possibility of having negative cycles. As we save all vertices on a path we can eliminate routes with cycles. Moreover, an ordered-queue list is used to sort the matrix entities so that a matrix entity which is more likely to lead to a violated constraint is selected. The ordered-queue list is filled by the labels of matrix M and is a triple containing demand, reduced cost and vertex. The queue list is ordered based on demand, reduced cost and vertex, respectively. In each step, an entity with least demand is chosen. If there are two entities with the same demands, then the one with the least reduced cost is chosen. If two entities have the same demands and the same reduced costs, then the one with smaller vertex number is chosen. And finally, we use the concept of dominance so that instead of having one label in each entity of matrix M, a bucket of labels reaching to vertex v through different paths are calculated. So, a set of alternative walks are considered rather than only one walk. A label is added to the bucket of entity $M(q, v)$ if it is not dominated by labels which have been already added to the bucket. A label is dominated if its reduced cost and its demand (at the same time) are greater than the reduced costs and the demands of existing labels in the bucket. Likewise, a label in the bucket will be deleted if it is dominated by a new label.

6.4.2 Extensive pricing procedure

Since the failure cost is embedded in the edge cost, the reduced cost of each edge depends on the fact that using which route we visit an edge/vertex. So, as mentioned to identify the shortest route, we need to keep the cost and the elements of all possible paths from the depot to vertex i. This leads to check exponentially many possible paths to each vertex. Using the Modified Pricing Procedure (MPP) we eliminate paths which are less likely to be part of the optimum solution to make the search space as small as possible. The MPP eliminates a label from the bucket of an entity or does not add a label to the bucket of an entity if its demand and its reduced cost at the same time are larger than another label's demand and reduced cost. Therefore, the MPP may eliminate some part of the optimum solution. To avoid this problem, we add another rule to our eliminating rules.

We explain the new rule using a simple example. Let $S_1(v)$ and $S_2(v)$ be two sets representing two paths starting from the depot and ending at vertex v. Let us assume $S_2(v) \subseteq S_1(v)$ such that both paths have the same vertices with the same order but in path 2 we skip one vertex or more (Figure 6.2). If $\bar{c}_2 \leq \bar{c}_1$ and $\mathbb{E}[q_2] \leq \mathbb{E}[q_1]$, then path 2 dominates path 1. We can eliminate path 1 with the knowledge that there is a cheaper way to reach vertex i. But in general, we cannot eliminate a path just because its reduced cost and its demand is larger than others'.

Figure 6.3 illustrates the later case that we cannot eliminate path 1. The MPP eliminates path 1 in the both examples. We add the new rule that we can eliminate a path like path 1, if $S_2(v) \subseteq S_1(v)$, given that the other rules apply.

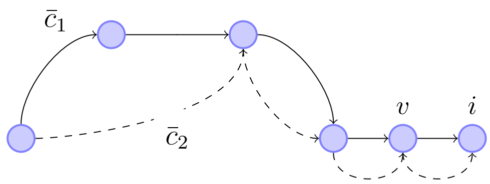

Figure 6.2: Path 2 dominates path 1

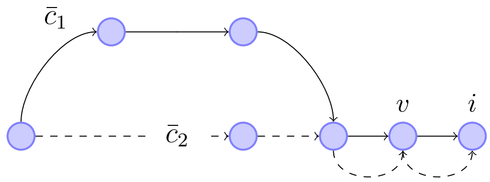

Figure 6.3: Path 2 does not dominate path 1

6.5 Concluding remarks

In this chapter, we addressed the SCVRP with recourse action. This problem is probably the most well-studied variant of the VRP with uncertain parameters. We proposed a new formulation within the column generation framework. Our proposed method overcomes with other methods problems which have been proposed within the column generation framework. The issue lies in the pricing problem where the search space should be narrowed to those routes which are more likely to be in the optimal solution. If elimination rules are not defined appropriately, there is a possibility to eliminate some routes which may be part of the optimal solution. We updated the elimination rules by introducing a new rule.

We defined two pricing procedures. The first one is quick and may eliminate some routes of the optimal solution. But we run the second pricing procedure when the first procedure fails to find routes with negative reduced cost.

The recourse action we considered in this study was the popular recourse action that if a failure occurs, the vehicle must return to the depot for replenishment and resume the pre-planned route.

Chapter 7

COMPUTATIONAL EXPERIMENTS

In this chapter,we first investigate the performance of the different valid inequalities and lifting techniques studied in Chapter 2 for the deterministic HVRP. Then, we present the computational analysis on the four approaches within stochastic optimization for two variants of the VRP with uncertain demands by conducting different experiments.

Similar with the theoretical chapters, this chapter is also categorized into two main parts. In the first part we study all issues on the deterministic HVRP and the HVRP with uncertain demands presented in Chapters 2, 3 and 4. To assess the performance of the valid inequalities and the lifting techniques, we compare the gap between the lower bound obtained by solving LP relaxations and the upper bound obtained from Yaman[79]. As will be explained in more detail, we use simulation to investigate the impact of CCP, BN robust optimization approach and BS optimization approach on optimal solutions and objective functions for different risk scenario levels. In the second part, we study the CVRP with uncertain demands where we investigate the performance of the four approaches within stochastic optimization presented in Chapters 5 and 6. Analogous with the first part, we use simulation to compare the results for the CVRP with uncertain demands.

7.1 Computational experiment for HVRP

In Section 7.1.1, we present percentage gaps for the lower bounds corresponding to the LP relaxation of different formulations for the deterministic model. In Section 7.1.2, we present three performance measures, by which we analyse the solutions of the three uncertainty models considered in Section 3 (i.e., BN, BS and CC).

Regarding the experiments on the HVRP, our computational experiments use two sets of benchmark instances: Golden et al [39] and Prins & Prodhon `http://prodhonc.free.fr/`. We denote them by G and P, respectively. G instances correspond to *single-depot* HVRP with unlimited fleet size and fixed costs. P instances were originally generated for the *homogeneous location routing problem*, so we modify them to obtain *multi-depot* HVRP with *limited fleet size*. In particular, according to the solutions presented in `http://prodhonc.free.fr/`, we limit the number of vehicles to that needed to serve the customers. We change the capacity of vehicles to define a heterogenous fleet (Q_k). We assign a coefficient (OC_k) as operational (traveling) cost for each type, so that the matrix c_a^k is calculated taking the

Instance	NO. Veh.	Cap.		k=1	2	3	4	5
P-20-5-5-1a	5	70	OC_k	1	1.2	1.4	1.6	2
			Q_k	70	100	130	160	190
P-20-5-3-1b	3	150	OC_k	1	1.2	1.4		
			Q_k	150	200	250		
P-20-5-5-2a	5	70	OC_k	1	1.2	1.4	1.6	2
			Q_k	70	100	130	160	190
P-20-5-3-2b	3	150	OC_k	1	1.2	1.4		
			Q_k	150	200	250		

Table 7.1: Vehicle type details

distance between nodes and multiplying it by OC_k. In Table 7.1 we report for each instance of type P, the number of vehicles (NO. Veh.), the original capacity (Int. Cap.) and for each type ($k = 1, \ldots, 5$) the corresponding operational cost (OC_k) and capacity (Q_k).

7.1.1 Lower bounds for the deterministic model

Table 7.2 shows percentage gaps between the lower bounds and the upper bounds for different formulations of the deterministic HVRP. The lower bounds are obtained relaxing the integrality conditions, the upper bounds obtained from Yaman [79]. Although we tried to follow Yaman's work step by step to produce the same lower bound, but we failed to achieve the bound presented in [79]. The single-depot HVRP with fixed cost is considered. We do not claim that these bounds are the best known bounds. Although Instead we would like to compare and assess the advantage of the valid inequalities and the lifting techniques we studied in Chapter 2. The first column represents the instances e.g., G-n20-m5 has 20 vertices, 5 types of vehicles and unlimited number of vehicles of each type. The second column (MTZ) corresponds to the LP relaxation of the standard MTZ formulation (2.1-2.9). The third column (Cap.) corresponds to the LP relaxation of the standard MTZ formulation after adding the capacity inequalities (2.10) which are kept in the succeeding column. The fourth column (DL) is obtained substituting (2.7, 2.8) with (2.23, 2.24) in (2.1-2.9). The fifth column (RLT) is obtained by replacing (2.7, 2.8) with (2.31 - 2.34b). The big-M method can be used to linearize the non-linear term in the RLT (2.30) as follows. The gap for the RLT^M is provided in its corresponding column.

$$y_{ij} \leq u_i, \; i, j \in V_c, \tag{7.1a}$$

$$y_{ij} \geq u_i - M(1 - \textstyle\sum_{k \in K} x_{ij}^k), \; i, j \in V_c, \tag{7.1b}$$

$$y_{ij} \leq M \textstyle\sum_{k \in K} x_{ij}^k, \; i, j \in V_c, \tag{7.1c}$$

As the numerical results suggest, the RLT^M formulation dominants the other formulations and lifting techniques.

7.1.2 Experiments

We start describing how the data uncertainty is constructed, then we explain the performance measures used and finally we analyze the computational results.

Uncertain Data To build demand uncertainty sets for the BS and BN robust models, we allow q_i to vary up to a fixed percentage of its nominal value so that $q_i \in [q_i^0 - \upsilon q_i^0, \; q_i^0 + \upsilon q_i^0]$ where q_i^0 is the demand nominal value and $\upsilon = 0.1$ or 0.2. To build uncertainty sets for the CC model, it is quite common to consider a normal

Instance	MTZ	Cap.	DL	RLT	RLT^M
G-n20-k5	76.78	13.46	11.30	11.16	9.93
G-n20-k3	96.56	3.50	3.17	3.16	3.04
G-n20-k5	77.84	18.09	17.09	16.96	13.25
G-n20-k3	96.60	5.01	4.81	4.78	4.27
G-n50-k6	84.75	9.55	9.03	9.01	7.88
G-n50-k3	95.99	5.91	5.74	5.73	5.51
G-n50-k3	84.87	13.72	12.85	12.79	11.06
G-n50-k3	85.70	10.06	9.18	9.15	7.39
G-n75-k4	71.95	12.18	9.80	9.79	8.17
G-n75-k6	79.55	15.30	13.69	13.68	12.74
G-n100-k3	93.31	6.53	6.06	6.06	5.59
G-n100-k3	85.53	12.94	12.09	12.06	10.35

Table 7.2: Gap on percentage for the deterministic models

distribution based on the mean and the variance calculated for a sample. Hence, we assume that the demand of each customer follows the normal distribution $\mathcal{N}(\mu_i, \lambda_i^2)$ where $\mu_i = q_i^0$, and $\lambda_i^2 = \frac{0.16}{12} q_i^0$. Notice that we set the variance equal to the variance of the uniform distribution we calculated for the RO cases. In this case, 91% of the interval defined previously is covered by the normal distribution function.

Performance measures We compare our solutions according to three performance measures.

First, we compute the *extra cost* which is required to pay to achieve a certain level of validity for routes:

$$\mathsf{E}^a := \frac{z^a - z^{det}}{z^a} \times 100$$

where E^a denotes the extra cost value, z^a denotes the optimal value of the uncertain model (a can be bs, bn and cc for BS, BN and CC models, respectively) and z^{det} is the optimal value for the deterministic case.

In case of failure, there are two possible strategies. On the one hand, one may assume that vehicles return to the depot and do not resume the interrupted (failed) route, so the remaining customers on the failed route are left unserved. This is known as *allowed lost sales* (ALS). The second performance measure represents the *number of unmet customers* (and the corresponding unmet demand). On the other hand, if lost sale is not allowed (NALS), the vehicle returns to the depot for a replenishment and then resumes the route starting from the first customer which was left unserved. The third performance measure calculates the *recourse cost*.

Since the probability of failure (*risk level*) and the cost are conflicting goals, we would like to find a proper threshold. Risk level is an important parameter in CC and BS models (denoted by α in Sections 3.4 and 3.2) by which we can adjust the conservativeness of the solutions. From the sensitivity analysis for MIP, we know that for small perturbations of parameters, the optimal solution may remain unchanged and from some point, the optimal solution will change. However, the behaviour of the optimal solution in respect to changes in MIP's parameters is not quite predictable and the value function in MIP is in general non-convex. This topic has been widely studied, see [18]. By changing the risk level in fact we can measure the sensitivity of the optimal solution and find the thresholds at which the optimal solution will change. In many cases in MIPs, a full description of the convex hull of the feasible region is not available and constraints may not define facets of the convex hull, hence changing the parameters may not affect neither the optimal solution nor the objective value. On the other hand, if an optimal solution is cut off as a result of varying parameters, the effect can be dramatic from changing the optimal solution to infeasible solution. Therefore, in practice in particular when resources are limited it is vital to define appropriate risk levels so that not only the solution is feasible but also unnecessary extra costs are not imposed. One way of identifying the threshold is to define different scenarios for the risk level. Here in addition to the nominal case which represents $\alpha_0 = 0.5$, we consider 9 scenarios for the risk level ($\alpha_1 = 0.40,\ \alpha_2 = 0.30,\ \alpha_3 = 0.25,\ \alpha_4 = 0.20, \alpha_5 = 0.10,\ \alpha_6 = 0.05,\ \alpha_7 = 0.03,\ \alpha_8 = 0.01,\ \alpha_9 = 0.001$). Note that the larger risk level, the higher is the probability of violating a constraint. We solve the CCP and BS RO deterministic counterpart of the instances for all these scenarios and calculate the aforementioned performance measures for each scenario. As formulated in the previous section, the protection level of the BS RO (Γ) is calculated for each risk level. Then, among the risk level scenarios, the optimal one can be suggested.

Computational results In this experiment, we consider the variable routing cost without fixed routing cost for the data sets. All experiments are carried out on a Dell Precision T1600 computer with a 3.4 GHz Intel Xeon Processor and 16 GB RAM running Ubuntu Linux 12. Also note that we use our B&C method for the nominal problem and the BN RO and the default CPLEX solver for the BS RO. When a user defined B&C method is run in CPLEX, by default CPLEX uses only one thread.

Tables 7.3 and 7.4 present E^{bs} and E^{bn} values when $\upsilon = 10\%$ and $\upsilon = 20\%$, respectively. Table 7.5 presents E^{cc} values when $\upsilon = 20\%$. All running times are in seconds. Note that, when the BS optimal value equals the BN optimal value, we do not need to run other risk levels since they will give the same results. When this happens, we use bold numbers in Tables 7.3 and 7.4 for the corresponding percentage of extra cost.

Inst.		Nom.	BS							BN
			Γ=1.14	2.34	3.01	3.77	5.74	7.35	8.76	
			α=0.40	0.30	0.25	0.20	0.10	0.05	0.03	
UL. Veh.										
G-n20-k5	E	623.22	1.07	1.33	1.45	1.91	3.19	3.19	**4.30**	4.30
	T		4147	2199	1264	859	2103	1366	4429	2183
G-n20-k3	E	387.18	0.82	1.04	1.04	1.04	1.92	**3.31**	-	3.31
	T		24561	5208	6317	1889	2470	29387		43392
G-n20-k5	E	742.87	N[a]	N	N	N	4.97	4.97	N	6.06
	T						5632	1079		1206
G-n20-k3	E	415.03	0.00	0.00	1.96	2.20	2.35	**2.59**	-	2.59
	T			1967	6755	4168	2528	2364		21450
L. Veh.										
P-20-5-5-1a	E	234.36	0.65	N	N	**0.77**	-	-	-	0.77
	T		30806			11679				37518
P-20-5-3-1b	E	217.58	0.00	0.00	0.00	0.00	**0.55**	-	-	0.55
	T		1354	626	956	733	793			3427
P-20-5-5-2a	E	194.46	0.00	**3.06**	-	-	-	-	-	3.06
	T		1124	1714						2228.71
P-20-5-3-2b	E	180.48	0.00	0.00	0.00	0.00	**3.99**	-	-	3.99
	T					14	117			417

Table 7.3: The deterministic optimal objective value and the first performance measure for BS RO and BN RO ($\upsilon = 0.1$)

[a]Not be able to solve due to out-of-memory error.

In order to calculate the second and third performance measures, we generate random demands for each customer from their defined distribution functions to simulate the actual situations. Table 7.6 reports the results for the average of the second and the third performance measures for 100 simulations when $\upsilon = 20\%$. For each instance, we use abbreviations as follows U (Unmet Demands), N (Number of

Inst.	Nom.	BS									BN
		Γ=1.14	2.34	3.01	3.77	5.74	7.35	8.76	10.39	16.99	
		α=0.40	0.30	0.25	0.20	0.10	0.05	0.03	0.01	0.001	
UL. Veh.											
G-n20-k5	E 623.22	1.31	3.09	4.12	4.12	6.72	7.23	7.87	8.55	8.55	9.11
	T	2102	1879	3324	1439	915	1225	1187	959	806	1163
G-n20-k3	E 387.18	1.03	1.89	3.20	3.20	4.15	**4.56**	-	-	-	4.56
	T	4355	5025	3269	1245	1269	1409	-			494
G-n20-k5	E 742.87	N	5.49	7.39	N	N	10.35	10.35	10.35	N	11.07
	T		3921	3679			17680	48302	10205		545
G-n20-k3	E 415.03	2.15	2.53	2.5	N	7.62	8.15	8.89	**10.49**	-	10.49
	T	20488	9939	4446		15668	6528	25907	38829		38794
L. Veh.											
P-20-5-5-1a	E 234.36	0.77	0.76	0.76	1.04	**4.65**	-	-	-	-	4.65
	T	20929	4903	-	14132	21130					88165
P-20-5-3-1b	E 217.58	0.00	**0.55**	-	-	-	-	-	-	-	0.55
	T	467	645								122
P-20-5-5-2a	E 194.46	2.97	2.97	2.97	2.97	**4.59**	-	-	-	-	4.59
	T	3114	835	1682	950	1229					
P-20-5-3-2b	E 180.48	0.00	3.84	3.84	3.84	3.84	5.08	**5.73**	-	-	5.73
	T	22	158	156	146	114	64	250			104

Table 7.4: The deterministic optimal objective value and the first performance measure for BS RO and BN RO ($\upsilon = 0.2$)

	Nom.	CCP								
Inst.		$\alpha = 0.40$	0.30	0.25	0.20	0.10	0.05	0.03	0.01	0.001
UL. Veh.										
G-n20-k5	E 623.22	2.72	6.84	8.80	10.01	18.81	24.79	29.21	34.79	41.73
	T	278	190	69	37	102	40	21	25	6
G-n20-k3	E 387.18	2.56	4.27	5.09	5.16	11.63	14.65	17.51	19.87	28.92
	T	1287	864	402	132	312	172	212	118	38
G-n20-k5	E 742.87	5.87	10.21	14.01	14.01	24.17	32.43	38.08	43.16	58.12
	T	2622	293	2096	264	1674	242	532	53	22
G-n20-k3	E 415.03	2.20	8.52	10.23	11.41	17.75	24.70	27.89	31.09	38.73
	T	509	3435	2472	1044	389	408	547	257	38
L. Veh.										
P-20-5-5-1a	E 234.36	N	N	N	N	N	N	N	N	N
	T									
P-20-5-3-1b	E 217.58	0.00	0.55	0.55	0.55	0.98	6.62	6.62	12.03	13.23
	T	786	408	195	139	162	399	295	1197	694
P-20-5-5-2a	E 194.46	3.06	3.06	4.81	4.81	7.10	7.10	15.41	18.19	21.95
	T	1722	1702	2357	566	1003	439	4366	7845	1808
P-20-5-3-2b	E 180.48	0.00	3.99	3.99	5.35	10.27	11.44	12.35	12.35	21.67
	T	21	134	139	58	398	411	895	428	4194

Table 7.5: The deterministic optimal objective value and the first performance measure for CCP ($\upsilon = 0.2$)

Unmet demands) and R (Recourse Cost). As the numerical result suggests, we do not need to set a very low risk level to achieve 100% valid routes. Table 7.7 also presents these measures for the CCP models.

Figure 7.1 illustrates the actual costs, the optimal costs the BS RO for the defined scenarios of the risk level and also the optimal cost of the BN RO. The actual cost is calculated based on the BS RO solution for each scenario as follows. For each scenario of the risk level, the routes are set according its BS RO solution and then the customer demands are generated from their pre-defined intervals assuming that they follow a uniform distribution (100 realizations for each demand).There is a possibility of failure for the scenarios. So, having assumed that no lost sales are allowed, the recourse actions are performed to serve unmet customers and the related cost (the recourse costs) is calculated and added to the optimal cost obtained by the BS RO. We call this total cost as the *actual cost with recourse action*. Figure 7.1 illustrates these three cost graphs for instance G-n20-k5. One can observe that if the risk level is set to a big value ($\alpha = 0.40$), the actual cost is even less than when the risk level is very small ($\alpha \geq 0.10$). It suggests that the extra cost paid to prevent the route validity for certain level is not necessary. In this specific problem, if the risk level is set to $\alpha = 0.20$, the total cost will be minimum. We can conclude that a lower risk level does not necessarily lead to a better result. Some unnecessary costs may be imposed without any significant outcome for the system. Figure 7.1 and Table 7.8 provide the optimal level of risk for each problem for the BS RO and the CCP when no lost sales are allowed. Obviously, the BN RO is too conservative and imposes unnecessary costs.

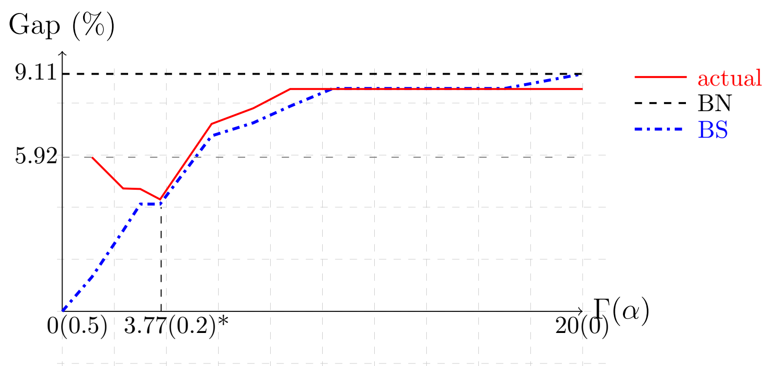

Figure 7.1: Risk levels, optimal costs and actual costs for Instance G-n20-k5

On the other hand, we can assume that lost sales/unmet customers are allowed in some cost. This means when a failure occurs the vehicle returns to the depot and does not resume the route, so the remaining customers on the failed route will be left unserved. To identify the optimal risk level in this case, let us assume a simple case where all lost sales have the same cost of f. We construct a pair comparison between each two scenarios to find out under which condition one is better than the other one. Let C_1, C_2, n_1 and n_2 be the optimal cost and the number of unmet customers for two risk scenarios 1 and 2, respectively. When $f \leq \frac{C_2-C_1}{n_1-n_2}$, then scenario 1 is better than scenario 2 and if $f \geq \frac{C_2-C_1}{n_1-n_2}$ then scenario 2 is better than scenario 1. Therefore, a risk level can be the best scenario for a specific range of lost sale costs. Table 7.9 presents intervals for f in which a risk level is optimal when lost sales is allowed (ALS). For instance, for Instance G-n20-k5, for the BS RO when $f \in [0,\ 19.97]$ and $f \in [19.97,\ 40.23]$, then the best risk levels are $\alpha = 0.4$ and $\alpha = 0.3$, respectively. However, $\alpha = 0.25$ cannot be the optimal risk level for any interval as it has the same cost of $\alpha = 0.2$ while there are unmet customers. Therefore, when $f \in [40.23,\ \infty)$, then $\alpha = 0.2$ is the optimal risk level. As this analysis also suggests, the smaller risk levels are not necessarily the best options.

Inst.	Γ	1.14	2.34	3.01	3.77	5.74	7.35	8.76	10.39	16.99
	α	0.40	0.30	0.25	0.20	0.10	0.05	0.03	0.01	0.001
UL. Veh.										
G-n20-k5	U	18.55	5.93	2.42	0	0	0	0	0	0
	N	8.30	2.5	0.80	0	0	0	0	0	0
	R	28.58	9.55	2.51	0	0	0	0	0	0
G-n20-k3	U	3.93	N	0.25	0	0	0	0	0	0
	N	0.25		0.01	0	0	0	0	0	0
	R	0.97		0.36	0	0	0	0	0	0
G-n20-k5	U	N	3.07	4.31	N	N	0	0	0	0
	N		0.27	0.09			0	0	0	0
	R		12.93	2.40			0	0	0	0
G-n20-k5	U	0.48	0.73	0	0	0	0	0	0	0
	N	0.04	0.03	0	0	0	0	0	0	0
	R	1.32	1.40	0	0	0	0	0	0	0
L. Veh.										
20-5-5-1a	U	N	0	0	0	0	0	0	0	0
	N		0	0	0	0	0	0	0	0
	R		0	0	0	0	0	0	0	0
20-5-3-1b	U	0.65	0	0	0	0	0	0	0	0
	N	0.03	0	0	0	0	0	0	0	0
	R	0.40	0	0	0	0	0	0	0	0
20-5-5-2a	U	0	0	0	0	0	0	0	0	0
	N	0	0	0	0	0	0	0	0	0
	R	0	0	0	0	0	0	0	0	0
20-5-3-2b	U	0.30	0	0	0	0	0	0	0	0
	N	0.02	0	0	0	0	0	0	0	0
	R	0.16	0	0	0	0	0	0	0	0

Table 7.6: Second and third performance measures for the BS ($\upsilon = 0.2$)

Inst.	Γ	1.14	2.34	3.01	3.77	5.74	7.35	8.76	10.39	16.99
	α	0.40	0.30	0.25	0.20	0.10	0.05	0.03	0.01	0.001
UL. Veh.										
G-n20-k5	U	46.96	52.55	43.72	43.01	20.62	4.32	1.89	0	0
	N	1.37	1.25	0.99	1.00	0.37	0.09	0.02	0	0
	R	42.88	46.96	31.93	35.41	13.07	3.12	0.59	0	0
G-n20-k3	U	19.12	16.05	13.16	16.99	1.62	1.43	0	0	0
	N	0.75	0.38	0.25	0.31	0.03	0.03	0	0	0
	R	20.51	11.87	7.63	8.9	1	1.33	0	0	0
G-n20-k5	U	49.21	37.07	29.38	25.38	5.48	4.52	1.20	1.19	0
	N	1.64	0.97	0.64	0.55	0.1	0.05	0.02	0.03	0
	R	50.79	39.3	23.29	23.05	4.85	1.89	1.14	1.63	0
G-n20-k5	U	40.8	4.56	6.03	17.46	3.24	0	0	0	0
	N	0.82	0.17	0.20	0.30	0.06	0	0	0	0
	R	34.36	6.91	9.04	9.26	1.90	0	0	0	0
UL. Veh.										
20-5-5-1a	U	N	N	N	N	N	N	N	N	N
	N									
	R									
20-5-3-1b	U	1.05	0	0	0	0	0	0	0	0
	N	0.05	0	0	0	0	0	0	0	0
	R	0.67	0	0	0	0	0	0	0	0
20-5-5-2a	U	0	0	0	0	0	0	0	0	0
	N	0	0	0	0	0	0	0	0	0
	R	0	0	0	0	0	0	0	0	0
20-5-3-2b	U	0.32	0	0	0	0	0	0	0	0
	N	0.02	0	0	0	0	0	0	0	0
	R	0.16	0	0	0	0	0	0	0	0

Table 7.7: Second and third performance measures for the CCP ($\upsilon = 0.2$)

α	0.40	0.30	0.25	0.20	0.10	0.05	0.03	0.01	0.001	Best Scen.
Inst.					BS					
UL. Veh.										
G-n20-k5	5.92	4.72	4.70	4.30	7.21	7.80	8.55	8.55	8.55	0.20
G-n20-k3	3.55	4.65	3.40	3.31	4.33	4.78	4.78	4.78	4.78	0.20
G-n20-k5	N	7.24	7.72	N	N	10.35	10.35	10.35	N	0.30
G-n20-k3	2.52	2.93	2.59	N	8.25	8.87	8.89	10.49	10.49	0.40
L. Veh.										
20-5-5-1a	0.77	0.77	0.77	1.05	4.88	4.88	4.88	4.88	4.88	0.25
20-5-3-1b	0.30	0.55	0.55	0.55	0.55	0.55	0.55	0.55	0.55	0.40
20-5-5-2a	3.06	3.06	3.06	3.06	4.81	4.81	4.81	4.81	4.81	0.25
20-5-3-2b	0.09	3.99	3.99	3.99	3.99	5.35	6.08	6.08	6.08	0.40
					CCP					
UL. Veh.										
G-n20-k5	0.10	0.14	0.14	0.16	0.21	0.25	0.29	0.35	0.42	0.40
G-n20-k3	0.08	0.07	0.07	0.07	0.12	0.15	0.18	0.20	0.29	0.20
G-n20-k5	0.13	0.16	0.17	0.17	0.25	0.33	0.38	0.43	0.58	0.40
G-n20-k3	0.10	0.10	0.12	0.14	0.18	0.25	0.28	0.31	0.39	0.30
L. Veh.										
20-5-5-1a	N	N	N	N	N	N	N	N	N	N
20-5-3-1b	0.00	0.01	0.01	0.01	0.01	0.07	0.07	0.12	0.13	0.40
20-5-5-2a	0.03	0.03	0.05	0.05	0.07	0.07	0.15	0.18	0.22	0.30
20-5-3-2b	0.00	0.04	0.04	0.05	0.10	0.11	0.12	0.12	0.22	0.40

Table 7.8: Best scenario for the risk level when lost sales are not allowed for BS and CCP

7.2 Computational experiment for CVRPUD

In this section we report the computational results of our experiments on the CVRPUD solved via the column generation methods. Similar to the previous section, we use simulation to analyze the performance of the models and also use the same performance measures.

To carry out our experiments, we use the standard instances for the CVRP

	α=0.40	0.30	0.25	0.20	0.10	0.05	0.03	0.01	0.001
					BS				
UL. Veh.									
G-n20-k5	[0,19.97]	[19.97,40.23]	-	[40.23,∞)	-	-	-	-	-
G-n20-k3	[0,36.57]	-	[36.57,∞)	-	-	-	-	-	-
G-n20-k5	N	[0,78.22]	N	N	-	-	-	-	-
G-n20-k3	[0,163.2]	-	[163.2,∞)		-	-	-	-	-
L. Veh.									
20-5-5-1a	[0,∞)	-	-	-	-	-	-	-	-
20-5-3-1b	[0,39.8]	[39.8,∞)	-	-	-	-	-	-	-
20-5-5-2a	[0,∞)	-	-		-	-	-	-	-
20-5-3-2b	[0,360.3]	[360.3,∞)	-	-	-	-	-	-	-
					CCP				
UL. Veh.									
G-n20-k5	[0,100.25]	-	-	-	[100.25,133.19]	[133.19, 392.81]	[392.81,1739.85]	[1739.85,∞)	-
G-n20-k3	[0,17.84]	[17.84, 24.60]	-	[24.6,115.11]	-	[115.11, 758.1]	[758.1, ∞)	-	-
G-n20-k5	[0, 48.11]	[48.11,85.64]	-	[85.64, 167.61]	[167.61,1227.3]	[12273,1399]	-	[1399, 7445]	[1399,∞)
G-n20-k3	[0, 40.36]	[40.36, 348.05]	-	-	[348.05, 481.01]	[481.01,∞)	-	-	-
L. Veh.									
20-5-5-1a	N	-	-	-	-	-	-	-	-
20-5-3-1b	[0,23.88]	[23.88,∞)	-	-	-	-	-	-	-
20-5-5-2a	[0,∞)	-	-	-	-	-	-	-	-
20-5-3-2b	[0,360.3]	[360.3, ∞)	-	-	-	-	-	-	-

Table 7.9: Best intervals of the lost sale cost for each scenario for the BS and the CCP

available in `http://branchandcut.org`. To build demand uncertainty sets for the robust models, similar to the previous section for each demand, we consider an interval of v percent around its nominal value i.e., $q_i \in [q_i^0 - vq_i^0, q_i^0 + vq_i^0]$. In our experiments we assume $v = 0.20$. Since we assume demands are integer, if the bounds are not integer values then we round them down to the next integer number. But for converted TSPLIB instances which are instances originally generated for the TSP problem, each customer has a unit demand ($q_i^0 = 1$). We set $v = 100$. As the demand's values are integer, the possible values of the converted TSPLIB instances are 0, 1, 2.

We use a set of scenarios to model the uncertainty for the CCP models. The reason is that even when the distribution functions are known, not always their joint distribution is tractable and usually its calculation is not easy. In many cases, distributions are approximated and discretized with any desired accuracy. The output is considered as a set of scenarios. We generate 10 scenarios from the above defined intervals. For converted TSPLIB instances, we define demands differently. To generate the scenarios, we assume three possible values i.e., 0, 1, 2 for each customer.

The number of vehicles which are listed for the standard CVRP instances in `http://branchandcut.org` are the minimum number of vehicles needed to serve all customers for deterministic cases. Given that we assume uncertain demands, these numbers of vehicles may not be sufficient to serve all customers and it may result in infeasible solutions. Thus, we drop limitations of the number of vehicles and assume that an unlimited number of homogenous vehicles are available.

We implement our proposed branch-and-price algorithms introduced in Chapters 5 and 6 in SCIP (Solving Constraint Integer Programs). SCIP is a non-commercial mixed integer programming (MIP) solvers available at `http://scip.zib.de`. SCIP provides us with a framework in which we can implement our branch-and-price algorithms. As mentioned in Section 5.2, SCIP has a limitation which forces us to change our formulation to SP_E. This limitation is that if variables have tight bounds e.g., $\{0, 1\}$ then the pricer module may find and add a variable more than once into the master problem. Therefore, the tight bounds of variables must be imposed via constraints rather than in their definitions. It is why we defined new variables (x_e) and new constraints to overcome this issue.

All experiments are run on a Dell Precision T1600 computer with a 3.4 GHz Intel Xeon Processor and 16 GB RAM running Ubuntu Linux 12. Notice that SCIP does not provide us with parallel computing, therefore, we can use only one thread out of eight available threads.

7.2.1 Implementation

In order to improve the running time and the efficiency of our algorithms, we make two main modifications. The first modification is concerned with the initial set of routes. The second one is concerned with the pricing problem.

A good initial set of routes is very important in the overall efficiency of column generation based methods ([19]). Therefore, we implemented and tested three different heuristics to find an initial good solution. In the first one, we simply sort the customers based on their numbers in an ascending order. Unassigned customers are assign to a route until the vehicle's capacity is reached. Then, we add another route and repeat this procedure until no customer is left unassigned. We use the nominal values for the customers demand in the assignment procedure for the CVRP with recourse actions while for the CVRP without recourse actions, we hold the route feasibility conditions for each approach e.g., in the BN robust approach we consider the upper bounds of demands to construct the initial routes. Finally, the total (expected) costs of routes are calculated for the initial routes for each approach and the routes and their costs are added to the master problem as the initial set of routes. In the second heuristic, we sort the customers in an ascending order based on their distance to the depot. Then we assign the customers to routes/vehicles similar to the first heuristic.

The last heuristic is more complicated than the first two ones and is similar to Solomon [72]. In this heuristic, vertices are inserted between two adjacent vertices on a route based on two criteria. First, we sort the customers in a descending order on the basis of their distance to the depot. We choose the furthest vertex (i) and initialize a route between the depot and vertex i. Then, we calculate the best feasible insertion place for each unassigned customer on the route. Among all unassigned customers we pick the one which leads to a new route with the least cost. Once the capacity of a route is reached then we start a new route and repeat the procedure until all customers are assigned.

The second modification is concerned on the pricing problem. As mentioned in Chapters 5 and 6, we defined a queue list to sort the entities of matrix M. We now add another element to the defined triple and make it a quadruple. The new entry is called *ratio*. This element will be placed as the first criterion in the quadruple, so that the entity with smallest ratio will be first in the queue. The ratio is defined as follows:

$$r := \frac{M(v,q)}{l(M(v,q))}, \tag{7.2}$$

where $M(v,q)$ and $l(M(v,q))$ are the reduced cost of reaching vertex v with total demand q and the actual length of this entity, respectively. Using this ratio will rank an entity with smaller reduced cost and smaller actual length higher in the queue. Such an entity probably has more chance of being a part of the optimal solution. We need to either re-calculate the length in each iteration or save the actual length of each entity's path. In our implementation we save the actual length to each entity of matrix M and update it in each iteration.

7.2.2 Experiments

In this section, the computational results of our experiments will be reported. Table 7.12 presents the description of the instances we will use for our experiments. The first column refers to the instance's name which indicates the number of vertices and the number of vehicles required to serve all vertices. For instance E-n13-k4 has 13 vertices (one depot and 12 customers) and 4 vehicles. The second column shows the capacity tightness which is equal to the total demand divided by mQ. Finally, the last two columns denote the total expected cost and the percentage of increase in comparison with the objective value of the deterministic cases. To calculate the total expected cost, we use the solution (routes) of deterministic problem and simulate 1000 realizations of the demands from the intervals defined above.Then, having assumed that lost sales are not allowed, we calculate the total expected cost which consists of the routing cost plus the cost of return trips to the depot.

Instance	Obj	tightness	expected cost	inc. (%)
E-n13-k4	247	0.76	277	12.15
E-n22-k4	375	0.94	393.76	5
E-n23-k3	569	0.75	596.948	4.91
E-n30-k3	534	0.94	574.554	7.59
E-n76-k14	1021	0.97	1107.16	8.44
P-n16-k8	450	0.88	461.99	2.66
P-n20-k2	216	0.97	223.3	3.38
P-n22-k8	603	0.94	700.172	16.11
P-n22-k2	216	0.96	221.144	2.38
P-n23-k8	529	0.98	605.77	14.51
P-n40-k5	458	0.88	461.6	0.79
P-n55-k15	989	0.99	1143.9	15.66
P-n60-k15	968	0.95	1042.6	7.71
ulysses-n16-k3	85	1	99.6	13.3

Table 7.10: CVRP instances and the expected cost

We solve instances for Ben-Tal and Nemirovski robust optimization, chance constraint programming and stochastic programming with recourse action. We do not provide computational results for Bertsimas and Sim robust approach due to mainly the following issue. The first issue is that the probability bounds by which the protection parameter (Γ) is calculated for each constraint is not tight when the number of uncertain parameters is not large in a constraint. Therefore the value calculated for (Γ) is usually equal to the number of vertices in small α which is equivalent with Ben-Tal and Nemitovski robust optimization. In the standard CVRP instances, the number of vertices on a route are normally 3, 4 or 5.

Let us start with CCP and BN RO models. Analogous with the previous section, we solve instances corresponding to CCP and RO models and obtain their optimal solutions. As these two approaches are single stage and no recourse costs or recourse actions are suggested by their optimal solution, using simulation we generate demands from corresponding intervals for each customer. In our experiment, we consider 10 scenarios for the probability of failure ($\alpha = 0.40,\ 0.30,\ 0.20,\ 0.10,\ 0$). Table 7.11, the first column refers to the instance name. In the second column, the objective values for the corresponding deterministic instances are provided. In the third, each character refers to a performance measure as follows:

(A) refers to the percentage of increase in the expected cost of the CCP solution obtained from simulation for each scenario in comparison with the nominal objective value.

(R) indicates the expected recourse cost to serve all customers.

(N) refers to the expected number of failures.

(U) refers to the expected amount of demands that vehicles fail to serve in the first visit.

The forth column to the eighth column report the above measures for each scenario of α. The ninth column reports the percentage increase in the objective value in BN robust approach in respect to the nominal objective value. The final column is the best α which has the least expected cost.

This table also confirms our finding in the previous section that not necessarily the higher protection level results in a better overall solution. Notice that all instances in Table 7.11 have been solved to optimality within 2 hours time limit.

Instance	Obj. N.		0.40	0.30	0.20	0.10	0.00	RO	Best Scen.
E-n13-k4	247	A	17.82	15.69	15.69	15.69	12.16	4.05	0
		R	43.01	35.75	35.75	35.75	0.05		
		N	0.89	0.70	0.70	0.70	0.00		
		U	1433.23	1264.42	1264.42	1264.42	1.76		
E-n22-k4	375	A	4.79	8.05	8.72	7.71	6.99	13.07	40
		R	17.97	23.18	20.71	9.90	7.21		
		N	0.65	0.69	0.64	0.28	0.15		
		U	938.87	975.68	991.33	437.91	140.65		
P-n16-k8	450	A	9.53	8.51	7.58	5.43	8.44	5.78	10
		R	42.88	38.30	23.13	6.43	15.97		
		N	0.82	0.82	0.45	0.20	0.32		
		U	12.37	10.97	6.94	5.32	7.84		
P-n22-k8	603	A	9.54	6.68	7.91	12.13	7.22	10.45	30
		R	70.55	51.29	58.70	60.15	10.54		
		N	1.06	1.07	1.07	0.84	0.19		
		U	1102.22	802.12	1467.65	732.51	235.81		
P-n23-k8	529	A	17.61	14.50	14.50	7.69	9.09	14.56	10
		R	93.15	64.69	64.69	25.68	9.06		
		N	1.72	1.04	1.04	0.43	0.22		
		U	26.12	20.60	20.60	5.17	2.75		
ulysses	85	A	19.00	18.44	20.53	32.58	27.90	60.00	30
-n16-k3		R	8.15	4.67	5.45	2.70	0.71		
		N	0.62	0.41	0.30	0.17	0.04		
		U	1.01	0.68	0.51	0.29	0.08		

Table 7.11: Results for CCP and RO models for the CVRPUD

In Tables 7.12 and 7.13 we summarize the details of our computational experiments for the stochastic vehicle routine problem with recourse action. The second column in Table 7.12 reports the increase in the objective value in comparison with the objective values provided in `http://branchandcut.org`. In the third and fourth columns, we present the number of routes required to serve all customers and the number of routes on which a failure(s) occur and have to return to the depot for the solution of the SVRP with recourse action. The instances in which there is an increase in the number of routes are marked by $(*)$ in the third column. If an instance is solved to optimality then the optimality gap is zero. The optimality gap is provided in the fourth column. Finally the fifth column reports running time. We generate 1000 demand scenarios from the intervals defined for each customer's

Instance	Obj. Inc.(%)	No. R	No. Fail. R.	gap(%)	time(s)
E-n13-k4	6.2753	4	0	0	110.66
E-n22-k4	2.3467	5*	2	0	27.2
E-n23-k3	7.6977	3	0	0	109.73
E-n51-k5	1.7274	5	2	1.17	48600
E-n76-k14	0.6856	15*	0	0.75	
E-n76-k29	3.687	30*	5	0	214.37
P-n60-k15	4.4628	16*	3	0.25	29292.07
P-n16-k8	3.0222	8	1	0	0.22
P-n20-k2	0.4630	2	0	0	430.48
P-n40-k5	1.3100	5	1	0	17088.8
P-n22-k8	2.2886	9*	2	0	0.82
P-n22-k2	0.0000	2	0	0	500.36
P-n23-k8	5.3686	9*	2	0	4.89
P-n55-k15	2.8055	16*	4	0	205.91
P-n50-k10	2.4425	10	2	0.88	48948.67
P-n50-k8	1.4263	9*	0	0.78	124310
ulysses-n16-k3	12.56	3	3	0	1.02
P-101-37	4.410	39*	4	0	5167.35

Table 7.12: Results for SCVRP with recourse actions I

demand, then, calculate the total expected cost and the total expected recourse cost for the solution obtained from the deterministic CVRP (indicated by "Nominal") and the SVRP with recourse action (indicated by "2-stage"). Similar to Table 7.11, in Table 7.13, we report the two measures previously defined. As the table suggests in all cases the SVRP with recourse action provide better solutions even for those which we could not solve to optimality.

Instance		Nominal	2-stage	Instance		Nominal	2-stage
E-n13-k4	A	12.30	9.98	E-n76-k29	A	37.75	35.94
	R	30.38	2.65		R	27.40	24.59
E-n22-k4	A	5.00	4.79	P-n22-k8	A	16.11	6.43
	R	18.76	17.95		R	97.17	40.77
E-n23-k3	A	4.91	3.51	P-n22-k2	A	2.38	1.77
	R	27.95	17.00		R	5.14	3.83
E-n76-k14	A	8.44	7.07	P-n23-k8	A	14.51	6.41
	R	86.16	66.16		R	76.77	14.90
P-n60-k15	A	7.71	5.80	P-n55-k15	A	15.66	2.24
	R	74.60	21.15		R	154.90	61.11
P-n16-k8	A	2.66	2.66	ulysses-n16-k3	A	13.38	12.56
	R	11.99	11.99		R	11.37	10.67
P-n20-k2	A	3.38	1.76	P-101-k37	A	39.67	38.38
	R	7.30	2.80		R	28.40	25.92

Table 7.13: Results for SCVRP with recourse actions II

In addition, we increased the number of demands scenarios from 10 to 100 and solve three instances of the SVRP with recourse actions. The results of this experiment is summarised in Table 7.14. All instances are solved to optimality and the number of routes are the same as the number of routes when we used 10 scenarios reposted in Table 7.12. There are small changes in the objective function and also in the performance measures (A and R) in comparison with the case when we had 10 scenarios. Also the routes are the same but there are more failures on the routes as we have increased the number of scenarios and provided more information on the uncertain demands. As the results suggest all cases provide better expected objective value in comparison with the nominal models.

Instance	Obj. Inc.(%)	No. Fail. R.	time(s)	A	R
P-n23-k8-s100	5.82	3	6.12	25.81	23.28
P-n55-k15-s100	3.12	12	342.42	19.55	19.72
P-101-37-s100	6.2	17	3710.79	38.91	26.00

Table 7.14: Results for SCVRP with recourse actions for 100 demand scenarios

In Table 7.15, we compare the best solution of the CCP models with the solution of the SVRP with recourse action models for those instances we solved to optimality of the CCP approach. As this table suggests in our experiment the solutions of the SVRP with recourse action dominate the solution of the CCP models.

instance		CCP	Nominal	2-stage
E-n13-k4	A	12.16	12.30	9.98
	R	0.05	30.38	2.65
E-n22-k4	A	4.79	5.00	4.79
	R	17.97	18.76	17.95
P-n16-k8	A	5.43	2.66	2.79
	R	6.43	11.99	12.56
P-n22-k8	A	6.68	16.11	6.43
	R	51.29	97.17	40.77
P-n23-k8	A	7.69	14.51	6.41
	R	25.68	76.77	14.90
gr-n17-k3	A	75.29	28.87	62.77
	R	89.64	209.62	40.68
ulysses-n16-k3	A	18.44	13.38	12.56
	R	4.67	11.37	10.67

Table 7.15: Comparison between CCP (for the best scenario), Nominal and two stage stochastic models

Chapter 8

CONCLUSIONS AND FUTURE RESEARCH

This dissertation has investigated different formulations within mixed integer programming and different approaches within stochastic optimization for two variants of the vehicle routing problem with uncertain demands: heterogeneous vehicle routing problems and capacitated vehicle routing problems. We have addressed the following two main research questions:

- What are specific properties of our proposed VRPUD formulations?
- Which methods in stochastic programming and mixed-integer programming can be used to improve VRPUD solution algorithms?

First we presented a comprehensive polyhedral study of the heterogeneous vehicle routing problem which have led us to a relatively efficient formulation for the HVRP. This formulation is introduced based on Miller-Tucker-Zemlin (MTZ) formulation.

The computational experiment presented in Table 7.2 reports the efficiency of each type of key valid inequalities and lifting techniques. Among all valid inequalities, the capacity inequalities are most effective inequalities and our proposed Reformulation-Linearization technique dominates the existing lifting techniques in the literature. We have also extended the models to the multi depot HVRP.

However, the advantage of our proposed formulation does not end here. The main advantage of the proposed formulation is that the corresponding counterparts of uncertainty remain tractable via mixed integer linear programming (MILP). Thus, we could apply approaches within stochastic optimization to our models and solve the resulting problem via a mixed integer solver. In particular, we have applied three main single-stage approaches within stochastic programming to the HVRP with uncertain demands: chance constrained programming, Ben-Tal and Nemirovski robust approach as well as Bertsimas and Sim robust approach. We finally plugged the proposed models into a branch-and-cut method. We have conducted an extensive experiment for the models where we have tested the several separation algorithms and have compared the stochastic optimization approaches. Among the separation algorithms, the separation algorithm for the DFJ SEC inequalities is the most efficient one. Although we tried several separation algorithms for comb valid inequalities, it turned out to be not a very efficient type of inequalities of the HVRP.

Ben-Tal and Nemirovski's robust optimization approach is very conservative whereas chance constrained programming and Bertsimas and Sim robust optimization approach provide us with a parameter which is called the protection level and gives us ability of controlling the level of conservativeness. Using simulation we have conducted a scenario based experiment to find out which protection level results in the least costly set of routes. As the single-stage approaches do not suggest any recourse action or cost, we consider two possible actions if a failure occurs: first, returning trip to the depot and resuming the pre-planned route and second, leaving the remaining customers on the failed route unserved with a penalty. Since these methods do not take into account any recourse action, it is common knowledge that the higher protection level results in a better solution. On the contrary, as our experiment confirms, the high protection level is not always good and may impose unnecessary extra costs to the problem. Having calculated the total expected cost for different scenarios for the chance constraint models and Bertsimas and Sim models, we could observe from Table 7.8 that not always the higher scenario results in the routes with least expected cost. Table 7.8 reports which scenario leads to solutions with least expected costs when the lost sales are not allowed i.e., the vehicle must make a return trip to the depot for replenishment. Also for the second action, leaving the customers on the failed route unserved, Table 7.9 presents for which range of penalty costs a scenario of the protection level is the optimum. Notice that we have used intervals to model uncertainty for the robust optimiztaion approaches and the Normal distribution for the chance constrained programming.

As column generation based methods are reported to be the most successful methods for variants of the deterministic vehicle routing problem, we have formulated the capacitated vehicle routing problem with uncertain demands within this framework. In addition to the three single-stage approaches, we have applied a two-stage stochastic approach known as stochastic vehicle routing problem with recourse action to the capacitated vehicle routing problem with uncertain demands within the column generation based framework. We have developed two master problems and call them edged based set-partitioning (SP_E) and edged based set-partitioning with route costs (SP_{EC}) formulations. The first master problem is used to model the single-stage approaches while the second one is used to model the stochastic vehicle routing problem with recourse action. The main issue to employ the stochastic optimization approach within the column generation framework lies in the pricing problem where routes are generated and their corresponding costs are calculated. In Chapter 5, using the definition of feasible route we have proposed pricing problems which model the single-stage approaches. This development is new in the literature of the CVRP with uncertain demands. In Chapter 7, we studied the stochastic vehicle routing problem with recourse action where the recourse action considered is the traditional recourse action i.e., if a failure occurs on a route, the vehicle must return to the depot for a replenishment and resume the pre-planned route to serve the rest of customers on the route. We define a new way of calculating the expected cost for a route in the pricing problem. Using standard pricing procedures may result in eliminating routes which are part of the optimal solution as the expected cost does also depend on the fact that using which route we visit an edge and a vertex. Thus, we have suggested a new rule to eliminate those routes which are dominated by other routes. In addition, we have tried three different heuristics to identify the initial routes.

To model the demand uncertainty, for chance constrained programming and the stochastic vehicle routing problem with recourse action, we have used data driven approaches and have defined a set of scenarios for the demands. Similar to the HVRP with uncertain demands, using simulation we have conduced an extensive experiment to test the four approaches for the CVRP with uncertain demands within the column generation framework. The results for the HVRP with uncertain demands were confirmed by similar experiments on the CVRP, i.e., not always the higher protection level results in a better solution. Our experiment also suggests the stochastic vehicle routing problem with recourse action provides better solutions (in terms of the total expected cost) for the CVRP with uncertain demands in comparison with the single-stage methods.

We finish this thesis with discussion of future research. The possible research in this field can be summarized as follows:

- In this research, we studied the HVRP and CVRP with uncertain demands. A possible research is to study other variants of vehicle routing problems such as VRP with time windows and uncertain parameters.

- In this work we studied single-stage approaches for the HVRP with uncertain demands. In the literature is no work on stochastic heterogenous vehicle routing problem with recourse action.

- the only uncertain parameter studied in this dissertation was demand. Also, in the literature customers' demand is the main uncertain parameter. In practice, other parameters such as routing cost, traveling time and availability of vehicles are sometimes subject to uncertainty. In particular, our proposed branch-and-price method provides a good framework to study uncertainty of other parameters.

- As described in Chapter 6, several types of recourse actions have been suggested in the literature. A possible research direction is to implement them within the proposed branch-and-price framework and investigate their advantages and disadvantages.

- In our proposed branch-and-price framework, our assumption on the type of demand uncertainty was limited to a date driven approach while distribution functions such as Poisson can be considered for uncertain demands.

- While data correlation is an important issue in stochastic optimization, taking data correlation into account usually results in intractable models for the VRP with uncertain parameters. Our proposed branch-and-price framework is flexible so that data correlation may be modelled while the problem's complexity remains the same. In addition, adjustable decision making has recently received a great deal of attention where dependency of decision variables and random variables can be modelled. Considering data and decision variable correlation in the vehicle routing problems with uncertain demands could be a possible research direction.

- Finally, embedding efficient cutting plane procedures into our branch-and-price framework can improve the framework efficiency so that larger instances can be solved to optimality.

Bibliography

[1] Y. Agarwal, K. Mathur, and H. M. Salkin, *A set-partitioning-based exact algorithm for the vehicle routing problem*, Networks, 19 (1989), pp. 731–749.

[2] S. Ahmed and A. Shapiro, *Tutorials in operations research*, INFORMS, 2008, ch. Solving Chance-Constrained Stochastic Programs via Sampling and Integer Programming.

[3] D. L. Applegate, R. E. Bixby, V. Chvátal, and W. J. Cook, *The traveling salesman problem*, Princeton University Press, 2006.

[4] J. Araque, L. Hall, and T. Magnanti, *Capacitated trees, capacitated routing and associated polyhedra.* Discussion paper, Center for Operations Research and Econometrics, Catholic University of Louvain, Belgium, 1990.

[5] E. Balas, *Intersection cuts - a new type of cutting planes for integer programming*, Operations Research, 19 (1971), pp. 19–39.

[6] E. Balas, *Disjunctive programming: properties of the convex hull of feasible points*, Discrete Applied Mathematics, 89 (1998), pp. 3–44.

[7] E. Balas, S. Ceria, and G. Cornuejols, *A lift-and-project cutting plane algorithm for mixed 0-1 programs*, Math. Program., 58 (1993), pp. 295–324.

[8] R. Baldacci, N. Christofides, and A. Mingozzi, *An exact algorithm for the vehicle routing problem based on the set partitioning formulation with additional cuts*, Math. Program., 115 (2008).

[9] R. Baldacci, E. Hadjiconstantinou, and A. Mingozzi, *An exact algorithm for the capacitated vehicle routing problem based on a two-commodity network flow formulation*, Operations Research, 52(5) (2004), pp. 723–738.

[10] R. Baldacci and A. Mingozzi, *A unified exact method for solving different classes of vehicle routing problems*, Math. Program., 120(2) (2009), pp. 347–380.

[11] M. L. Balinski and R. E. Quandt, *On an integer program for a delivery problem*, Operations Research, 12 (1964), pp. 300–304.

[12] J. F. Bard, G. Kontoravdis, and G. Yu, *A branch-and-cut procedure for the vehicle routing problem with time windows*, Transportation Science, 36 (2002), pp. 250–269.

[13] J. M. Belenguer, E. Benavent, C. Prins, C. Prodhon, and R. W. Calvo, *A branch-and-cut method for the capacitated location-routing problem*, Computers& Operations Research, 38 (2011), pp. 931 – 941.

[14] A. Ben-Tal, S. Boyd, and A. Nemirovski, *Extending scope of robust optimization: comprehensive robust counterparts of uncertain problems*, Math. Program., B (2006).

[15] A. Ben-Tal and A. Nemirovski, *Robust solutions of uncertain linear programs*, Operations Research Letters, 25 (1999), pp. 1–13.

[16] A. Ben-Tal and A. Nemirovski, *Robust solutions of linear programming problems contaminated with uncertain data*, Math. Program., 88 (2000), pp. 411–424.

[17] D. Bertsimas and M. Sim, *The price of robustness*, Operations Research, 52(1) (2004).

[18] C. E. Blair and R. G. Jeroslow, *The value function of an integer program*, Math. Program., 23 (1982), pp. 237–273.

[19] J. Bramel and D. Simchi-Levi, *The vehicle routing problem*, Society for Industrial and Applied Mathematics Philadelphia, PA, USA, 2001, ch. Set-covering-based algorithms for the capacitated VRP, pp. 85–108.

[20] A. Charnes and W. Cooper, *Chance-constrained programming*, Management Science, 6 (1959), pp. 73–79.

[21] ——, *Deterministic equivalents for optimizing and satisficing under chance-constrained*, Operations Research, 11 (1963), pp. 18–39.

[22] C. H. Christiansen and J. Lysgaard, *A branch-and-price algorithm for the capacitated vehicle routing problem with stochastic demands*, Operations Research Letters, 35 (2007), pp. 773 – 781.

[23] C. H. Christiansen, J. Lysgaard, and S. Wøhlk, *A branch-and-price algorithm for the capacitated arc routing problem with stochastic demands*, Operations Research Letters, 37 (2009), pp. 392 – 398.

[24] G. Clarke and J. W. Wright, *Scheduling of vehicles from a central depot to a number of delivery points*, Operations Research, 12 (1964), pp. pp. 568–581.

[25] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, Massachusetts Institute of Technology, 2009.

[26] G. Dantzig, R. Fulkerson, and S. Johnson, *Solution of a large-scale traveling-salesman problem*, Operations Research, 2 (1954), pp. 393–410.

[27] M. Desrochers and G. Laporte, *Improvements and extensions to the miller-tucker-zemlin subtour elimination constraints*, Operations Research Letters, 10 (1991), pp. 27–36.

[28] S. S. Dey, *A note on the split rank of intersection cuts*, Math. Program., 130 (2011), pp. 107–124.

[29] M. Dror, *Modeling Uncertainty: An Examination of Stochastic Theory, Methods, and Applications. International Series in Operations Research and Management Science*, Kluwer, Boston, 2002, ch. Vehicle routing with stochastic demands: Models and computational methods, pp. 625–649.

[30] M. Dror, *Vehicle routing with stochastic demands: Models and computational methods*, in Modeling Uncertainty, M. Dror, P. L'Ecuyer, F. Szidarovszky, F. S.

Hillier, and C. C. Price, eds., vol. 46 of International Series in Operations Research and Management Science, Springer New York, 2005, pp. 625–649.

[31] M. Dror and P. Trudeau, *Stochastic vehicle routing with modified savings algorithm*, European Journal of Operational Research, 23 (1986), pp. 228–235.

[32] J. Edmonds, *Maximum matching and a polyhedron with 0, 1-vertices*, Journal of Research of the National Bureau of Standards, B 69 (1965), pp. 125–130.

[33] EEA, *Laying the foundations for greener transport – term 2011: transport indicators tracking progress towards environmental targets in europe*, tech. rep., European Environment Agency, 2011.

[34] A. L. Erera, J. C. Morales, and M. Savelsbergh, *The vehicle routing problem with stochastic demand and duration constraints*, Transportation Science, 44 (2010), pp. 474–492.

[35] A. Frank, *On the edge-connectivity algorithm of nagamochi and ibaraki*, tech. rep., Egervary Research Group of MTA-ELTE, Department of Operations Research, Eotvos University, Pazmany P. s. 1/c. Budapest, Hungary, 1994.

[36] R. Fukasawa, H. Longo, J. Lysgaard, M. P. d. Aragão, M. Reis, E. Uchoa, and R. F. Werneck, *Robust branch-and-cut-and-price for the capacitated vehicle routing problem.*, Math. Program., 106 (2006), pp. 491–511.

[37] M. Gendreau, G. Laporte, and R. Seguin, *An exact algorithm for the vehicle routing problem with stochastic demands and customers*, Transportation Science, 29 (1995), pp. 143–155.

[38] M. Gendreau, G. Laporte, and R. Seguin, *Stochastic vehicle routing*, European Journal of Operational Research, 88 (1996), pp. 3–12.

[39] B. Golden, A. Assad, L. Levy, and F. Gheysens, *The fleet size and mix vehicle routing problem*, Computers & Operations Research, 11 (1984), pp. 49–66.

[40] M. Grotschel and O. Holland, *Solution of large-scale symmetric travelling salesman problems*, Math. Program., 51 (1991), pp. 141–202.

[41] M. Grotschel and M. W. Padberg, *On the symmetric travelling salesman problem i: inequalities*, Math. Program., 16 (1979), pp. 265–280.

[42] E. Hadjiconstantinou, N. Christofides, and A. Mingozzi, *A new exact algorithm for the vehicle routing problem based on $q-$paths and $k-$shortest paths relaxations*, Annals of Operations Research, 61 (1995), pp. 21–43.

[43] M. Hesse and J.-P. Rodrigue, *The transport geography of logistics and freight distribution*, Journal of Transport Geography, 12 (2004), pp. 171 – 184.

[44] C. Hjorring and J. Holt, *New optimality cuts for a single-vehicle stochastic routing problems*, Annals of Operations Research, 86 (1999), pp. 569–584.

[45] R. Jagannathan, *Chance-constrained programming with joint constraints*, Operations Research, 22 (1974), pp. 358–372.

[46] G. Laporte and F. Louveaux, *Formulations and bounds for the stochastic capacitated vehicle routing problem with uncertain supplies.*, tech. rep., Ecole des Hautes Etudes Commerciale, University of Montreal, Montreal, Canada, 1987.

[47] G. Laporte, F. Louveaux, and H. Mercure, *Models and exact solutions for a class of stochastic location-routing problems*, European Journal of Operational Research, 39 (1989), pp. 71–78.

[48] G. Laporte, F. Louveaux, and H. Mercure, *The vehicle routing problem with stochastic travel times*, Transportation Science, 26 (1992), pp. 161–170.

[49] G. Laporte, F. V. Louveaux, and L. van Hamme, *An integer l-shaped algorithm for the capacitated vehicle routing problem with stochastic demands*, Operations Research, 50(3) (2002), pp. 415–423.

[50] A. Letchford, D. Theis, and G. Reinelt, *A faster exact separation algorithm for blossom inequalities*, in Integer Programming and Combinatorial Optimization 10, Lecture Notes in Computer Science, vol. 3064, Springer, 2004, pp. 196–205.

[51] A. N. Letchford, *Separating a superclass of comb inequalities in planar graphs*, Mathematics of Operations Research, 25 (2000), pp. 443–454.

[52] A. N. Letchford, R. W. Eglese, and J. Lysgaard, *Multistars, partial multistars and the capacitated vehicle routing problem*, Math. Program., 94 (2002), pp. 12–25.

[53] A. N. Letchford and J.-J. Salazar-Gonzalez, *Projection results for vehicle routing*, Math. Program., 105 (2006), pp. 251–274.

[54] J. Lysgaard, A. N. Letchford, and R. W. Eglese, *A new branch-and-cut algorithm for the capacitated vehicle routing problem*, Math. Program., 100 (2004), pp. 423–445.

[55] C. Miller, A. Tucker, and R. Zemlin, *Integer programming formulations and traveling salesman problems*, J. ACM, 7 (1960), pp. 326–329.

[56] J. Mitchell, *Handbook of Applied Operations Research*, Oxford University Press, 2000, ch. Branch-and-Cut Algorithms for Combinatorial Optimization Problems.

[57] G. L. Nemhauser and L. A. Wolsey., *Integer and combinatorial optimization*, Wiley, 1988.

[58] A. Nemirovski and A. Shapiro, *Convex approximations of chance constrained programs*, SIAM Journal on Optimization, 17 (2006), pp. 969–996.

[59] C. Novoa, R. Berger, J. Linderoth, and R. Storer, *A set-partitioning-based model for the stochastic vehicle routing problem*, tech. rep., Lehigh University, 2006.

[60] M. Padberg and M. Grotschel, *The traveling salesman problem*, John Wiley & Sons, 1985, ch. Polyhedral computations, pp. 307–360.

[61] M. Padberg and G. Rinaldi, *Facet identification for the symmetric traveling salesman polytope*, Math. Program., 47 (1990), pp. 219–257.

[62] ——, *A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems*, SIAM Review, 33 (1991), pp. 60–100.

[63] A. Pessoa, M. P. de Aragao, and E. Uchoa, *A robust branch-cut-and-price algorithm for the heterogeneous fleet vehicle routing problem*, lecture Notes in Computer Science, 4525 (2007), pp. 150–160.

[64] V. Pillac, *Dynamic vehicle routing: solution methods and computational tools*, 4OR, 11 (2013), pp. 395–396.

[65] W. Powell, *Approximate Dynamic Programming*, John Wiley and Sons, 2nd edition ed., 2011.

[66] W. Rei, M. Gendreau, and P. Soriano, *A hybrid monte carlo local branching algorithm for the single vehicle routing problem with stochastic demands*, Transportation Science, 44 (2010), pp. 136–146.

[67] N. Secomandi, *A rollout policy for the vehicle routing problem with stochastic demands*, Operations Research, 49 (2001), pp. 796–802.

[68] N. Secomandi and F. Margot, *Reoptimization approaches for the vehicle-routing problem with stochastic demands*, Operations Research, 57 (2009), pp. 214–230.

[69] S. Shen, *Risk optimization in probabilistic programs with single or multiple chance constraints*, tech. rep., University of Michigan, Ann Arbor, 2012.

[70] H. Sherali and W. Adams, *A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems*, Siam Journal of Discrete Mathematics, 3 (1990), pp. 411–430.

[71] H. Sherali and P. Driscoll, *On tightening the relaxations of miller-tucker-zemlin formulations for asymmetric traveling salesman problems*, Operations Research, 50 (2002), pp. 656–669.

[72] M. M. Solomon, *Algorithms for the vehicle routing and scheduling problems with time window constraints*, Operations Research, 35 (1987), pp. 254–265.

[73] A. L. Soyster, *Convex programming with set-inclusive constraints and applications to inexact linear programming*, Operations Research, 21 (1973), pp. 1154–1157.

[74] W. R. Stewart and B. L. Golden, *Stochastic vehicle routing: A comprehensive approach*, European Journal of Operational Research, 14 (1983), pp. 371 – 385.

[75] I. Sungur, F. Ordóñez, and M. Dessouky, *A robust optimisation approach for the capacitated vehicle routing problem with demand uncertainty*, IIE Transactions, 40 (2008).

[76] F. A. Tillman, *The multiple terminal delivery problem with probabilistic demands*, Transportation Science, 3 (1969), pp. 192–204.

[77] P. Toth and D. Vigo, eds., *The Vehicle Routing Problem*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2001.

[78] L. A. Wolsey, *Integer programming*, Wiley, 1998.

[79] H. Yaman, *Formulations and valid inequalities for the heterogeneous vehicle routing problem*, Math. Program., A 106 (2006).

[80] W.-H. Yang, K. Mathur, and R. H. Ballou, *Stochastic vehicle routing problem with restocking*, Transportation Science, 34 (2000), pp. 99–112.