Decision Making under Uncertainty in Routing Problems for Reverse Logistics

Von der Fakultät für Mathematik der
Universität Duisburg-Essen
zur Erlangung des akademischen Grades eines
Dr. rer. nat.

genehmigte Dissertation

von
Nadine Wollenberg, M.Sc.
Geb. am 31.12.1987 in Oberhausen

Gutachter: Prof. Dr. Rüdiger Schultz
Prof. Dr. Ralf Borndörfer
Prof. Dr. Michel Gendreau

Tag der mündlichen Prüfung 23. November 2015

Acknowledgements

The research work presented in this thesis would have taken far longer or even would not have been possible to complete without my supervisor Prof. Dr. Rüdiger Schultz. I feel extremely lucky and privileged to be supervised by him. His supervision had a major influence not only on this thesis but also on my personal life. Giving me the opportunity to present my ongoing work at different conferences to get some feedback, to meet interesting people, and to connect to researchers all over the world, was one of the major advantages of doing my Ph.D. under Prof. Schultz‘s wing.

I wish to thank Prof. Dr. Michel Gendreau and Prof. Dr. Walter Rei for their fruitful discussion and valuable comments. Our cooperation in 2013 resulted in submitting a joint work to the journal Computers & Operations Research. I‘m very grateful to Prof. Dr. Jean-François Côté and Dr. Ralf Gollmer for their outstanding assistance with the implementation of the algorithm.

My brother, Tobias Wollenberg, inspired me to study math, which was of course not possible without being supported by my parents Ingrid and Detlef Wollenberg. Without their backup I would not have been able to come so far. I feel eternally thankful to have such a family.

Furthermore, I wish to thank all members of the working group for “Optimization and Discrete Mathematics” at Duisburg-Essen University for the friendly and amicable working atmosphere. I will always think with joy of this period and retain marvelous memories. I had a wonderful time.

I want to extend my gratitude to the Stiftung Mercator, Essen, who has supported my work on this thesis by a grant in their Mercur Research Program.

Abstract

We consider a stochastic extension of the vehicle routing problem with simultaneous delivery and pickup. While delivery amounts are assumed to be fixed and known in advance, pickup amounts are stochastic and revealed only after the determination of an a priori route. This may lead to arriving at a customer with insufficient capacity to collect the realized pickup demand. Such a situation is referred to as a failure. As corrective action an additional route is computed to collect the pickup amounts which are left at the failure points. The objective is to minimize the distance traveled in the first-stage with known delivery quantities plus the expected distance traveled along the corrective route. For the single vehicle case, we present a two-stage stochastic programming model with recourse as well as an exact algorithm to solve it. The proposed algorithm is based on an extension of the Integer L-Shaped method adapted for stochastic vehicle routing problems. Risk neutral and risk averse routing decisions are examined and compared.

Parts of this thesis have been submitted to the journal Computers & Operations Research.

Stochastic Integer Programming, Stochastic Vehicle Routing, Reverse Logistics, Integer L-Shaped.

Contents

List of Figures III
List of Tables V
List of Abbreviations IX

1 Introduction **1**

2 Stochastic Programming **5**
2.1 Decision Making under Uncertainty 5
2.2 Two-Stage Stochastic Linear Programs with Recourse 8

3 Routing under Uncertainty **13**
3.1 Deterministic Routing 13
3.1.1 Vehicle Routing with Simultaneous Delivery and Pickup 16
3.1.2 Generalized Traveling Salesman Problem 18
3.2 Capturing Uncertainty 19
3.2.1 VRP with Stochastic Demands 21
3.3 Solution Concepts and Algorithms 23
3.3.1 Exact Algorithms 23
3.3.2 Heuristics and Metaheuristics 28

4 Single-VRP with Simultaneous Delivery and Stochastic Pickup **31**
4.1 Literature Review 31
4.2 Modeling: Complete Recourse 32
4.2.1 Deterministic Equivalent 35
4.2.2 Risk Averse Strategies 36
4.3 Solution Framework: Integer L-Shaped Algorithm 39

4.3.1 Partial Routes 40
4.3.2 Lower Bounding Functionals 42
4.3.3 Computation of Lower Bounds on $\mathcal{Q}(x)$ 43
4.3.4 Global Lower Bounds 46

5 Computational Experiments **49**
5.1 Implementation Details 49
5.2 Computational Results 49
5.2.1 Risk Neutral Model 50
5.2.2 Averaged Value versus Averaged Data (Model) 56
5.2.3 Risk Averse Models 57

6 Conclusion **67**

A Appendix **69**

Bibliography **95**

List of Figures

Abb. 2.1: Block structure of constraint matrix 10

Abb. 2.2: Block structure of reformulation 10

Abb. 4.1: Two-stage scheme - Non-anticipativity 33

Abb. 4.2: Solution for Expected Value Model 37

Abb. 4.3: Solution for Risk Model: Excess Probability; $\rho = 1, 10$ 38

Abb. 4.4: Solution for Risk Model: Excess Probability; $\rho = 100$ 38

Abb. 4.5: Solution for Risk Model: Expected Excess; $\rho = 1$ 38

Abb. 4.6: Solution for Risk Model: Expected Excess; $\rho = 10, 100$ 39

Abb. 4.7: Partial Route 41

Abb. 4.8: General Partial Route 42

Abb. 5.1: 50-customer instance: Objective values for each of the hundred scenarios are rendered with bar charts for the solutions given by the EV-Model, EP-Model, and EE-Model (from top to bottom) for $\eta = 278.7135$ and $\rho = 50$. 61

Abb. 5.2: 50-customer instance: Probabilities together with objective values are plotted for each of the hundred scenarios and different risk measures; EV-Model, EP-Model, and EE-Model (from top to bottom) for $\eta = 278.7135$ and $\rho = 50$. 62

Abb. 5.3: 15-customer instance: Objective values for each of the hundred scenarios are rendered with bar charts for the solutions given by the EV-Model, EP-Model, and EE-Model (from top to bottom) for $\eta = 238.3877$ and $\rho = 50$. 63

Abb. 5.4: 15-customer instance: Probabilities together with objective values are plotted for each of the hundred scenarios and different risk measures; EV-Model, EP-Model, and EE-Model (from top to bottom) for $\eta = 238.3877$ and $\rho = 50$. 64

List of Tables

Tab. 4.1: Customer Data 37
Tab. 4.2: Solutions for Mean-Risk Models 39
Tab. 5.1: Computational results on instances with shuffled delivery demand 52
Tab. 5.2: Computational results on instances with 100% random customers, normally distributed pickup demand, and $\sigma = 0.2$ . . 52
Tab. 5.3: Computational results on instances with 50% random customers, normally distributed pickup demand, and $\sigma = 0.7$ . . 53
Tab. 5.4: Computational results on instances with 100% random customers, normally distributed pickup demand, and $\sigma = 0.7$ . . 54
Tab. 5.5: Computational results on instances with 100% random customers, uniformly distributed pickup demand, and 20% deviation 54
Tab. 5.6: Computational results on instances with 50% random customers, uniformly distributed pickup demand, and 70% deviation 55
Tab. 5.7: Computational results on instances with 100% random customers, uniformly distributed pickup demand, and 70% deviation 55
Tab. 5.8: Comparison of savings for instances with up to 50 customers . 57
Tab. 5.9: Average savings concerning the number of customers 57
Tab. 5.10: Solutions determined by EP-Model compared to solutions of EV-Model, EEV, and EE-Model with $\rho = 50$. 58
Tab. 5.11: Solutions determined by EP-Model compared to solutions of EV-Model, EEV, and EE-Model with $\rho = 100$. 59

Tab. 5.12: Cross-Check for random 15-customer instance with shuffled demand and 100 scenarios; $\rho = 50$ 65
Tab. A.1: Computational results on instances with shuffled delivery demand using F-Cuts, PR, and artificial nodes. 70
Tab. A.2: Computational results on instances with shuffled delivery demand using F-Cuts, PR, and GTSP. 71
Tab. A.3: Computational results on instances with shuffled delivery demand using F-Cuts, GPRs, and artificial nodes. 72
Tab. A.4: Computational results on instances with shuffled delivery demand using F-Cuts, GPRs, and GTSP. 73
Tab. A.5: Computational results on instances with 100% random customers, normally distributed pickup demand, and $\sigma = 0.2$ using F-Cuts, PRs, and GTSP. 74
Tab. A.6: Computational results on instances with 100% random customers, normally distributed pickup demand, and $\sigma = 0.2$ using F-Cuts, GPRs, and GTSP. 75
Tab. A.7: Computational results on instances with 50% random customers, normally distributed pickup demand, and $\sigma = 0.7$ using F-Cuts, PRs, and artificial nodes. 76
Tab. A.8: Computational results on instances with 50% random customers, normally distributed pickup demand, and $\sigma = 0.7$ using F-Cuts, PRs, and GTSP. 77
Tab. A.9: Computational results on instances with 50% random customers, normally distributed pickup demand, and $\sigma = 0.7$ using F-Cuts, GPRs, and artificial nodes. 78
Tab. A.10: Computational results on instances with 50% random customers, normally distributed pickup demand, and $\sigma = 0.7$ using F-Cuts, GPRs, and GTSP. 79
Tab. A.11: Computational results on instances with 100% random customers, normally distributed pickup demand, and $\sigma = 0.7$ using F-Cuts, PRs, and artificial nodes. 80

Tab. A.12: Computational results on instances with 100% random customers, normally distributed pickup demand, and $\sigma = 0.7$ using F-Cuts, PRs, and GTSP. 81
Tab. A.13: Computational results on instances with 100% random customers, normally distributed pickup demand, and $\sigma = 0.7$ using F-Cuts, GPRs, and artificial nodes. 82
Tab. A.14: Computational results on instances with 100% random customers, normally distributed pickup demand, and $\sigma = 0.7$ using F-Cuts, GPRs, and GTSP. 83
Tab. A.15: Computational results on instances with 50% random customers, uniformly distributed pickup demand, and 70% deviation using F-Cuts, PRs, and artificial nodes. 84
Tab. A.16: Computational results on instances with 50% random customers, uniformly distributed pickup demand, and 70% deviation using F-Cuts, PRs, and GTSP. 85
Tab. A.17: Computational results on instances with 50% random customers, uniformly distributed pickup demand, and 70% deviation using F-Cuts, GPRs, and artificial nodes. 86
Tab. A.18: Computational results on instances with 50% random customers, uniformly distributed pickup demand, and 70% deviation using F-Cuts, GPRs, and GTSP. 87
Tab. A.19: Computational results on instances with 100% random customers, uniformly distributed pickup demand, and 70% deviation using F-Cuts, PRs, and artificial nodes. 88
Tab. A.20: Computational results on instances with 100% random customers, uniformly distributed pickup demand, and 70% deviation using F-Cuts, PRs, and GTSP. 89
Tab. A.21: Computational results on instances with 100% random customers, uniformly distributed pickup demand, and 70% deviation using F-Cuts, GPRs, and GTSP. 90
Tab. A.22: Computational results for the EP-Model with $\rho = 50$. 91
Tab. A.23: Computational results for the EP-Model with $\rho = 100$. 92

Tab. A.24: Computational results for the EE-Model with $\rho = 50$. 93
Tab. A.25: Computational results for the EE-Model with $\rho = 100$. 94

List of Abbreviations

AS Ant Systems

B&B Branch&Bound

B&C Branch&Cut

B&P Branch&Price

CCP Chance Constrained Programming

CG Column Generation

CVRP Capacitated Vehicle Routing Problem

DA Deterministic Annealing

EE Expected Excess

EE-Model Expected Excess Model

EEV Expected Value of Expected Solution

EP Excess Probability

EP-Model Excess Probability Model

EV Expected Value

EV-Model Expected Value Model

GA Genetic Algorithm

GPR General Partial Route

GTSP Generalized Traveling Salesman Problem

LBF Lower Bounding Functional

NN Neural Networks

PR Partial Route

RMP Restricted Masterproblem

RO Robust Optimization

SA Simulated Annealing

SD Stochastic Dominance

SPPRC Shortest Path Problem with Resource Constraints

SPR Stochastic Programming with Recourse

SVRP Stochastic Vehicle Routing Problem

TS Tabu Search

TSP Traveling Salesman Problem

VRP Vehicle Routing Problem

VRPSD Vehicle Routing Problem with Stochastic Demands

VRPSDP Vehicle Routing Problem with Simultaneous Delivery and Pickup

VRPSDSP Vehicle Routing Problem with Simultaneous Delivery and Stochastic Pickup

1. Introduction

The Vehicle Routing Problem (VRP) with various extensions, e.g. time windows, inhomogeneous vehicle fleets, pickup and delivery, has been addressed since 1959 where it was introduced by Dantzig and Ramser [17]. It consists of determining a set of routes starting and ending at a depot, servicing with a fleet of identical vehicles of finite capacity a set of customers each having a known demand that minimizes the total travel cost. Deterministic VRPs belong to the NP-hard complexity class, since they generalize the Traveling Salesman Problem (TSP) and are thus difficult to solve. Many solution approaches involving both heuristic and exact methods have been published to tackle this type of problems, for detailed surveys see [38] and [72], an annotated bibliography is given by Laporte in [39].

In recent years, the study of Stochastic Vehicle Routing Problems (SVRP) has gained popularity. The three most common causes of randomness regarding problem data are stochastic customers, which means the presence or absence of a customer is not known with certainty, stochastic demands, and stochastic times. Most solution strategies are based on the determination of an a priori route and the consideration of some corrective policy. A survey on SVRPs is presented by Gendreau, Laporte and Séguin in [28]. The most studied SVRP of all is the Vehicle Routing Problem with Stochastic Demands (VRPSD) (cf. [28]). The first exact algorithm using an Integer L-Shaped method to solve the VRPSD was proposed by Séguin [64]. In [22] Dror, Laporte, and Trudeau outline a variety of operating and service policies, properties and models for the VRPSD.

In this thesis we concentrate on a stochastic extension of the Vehicle Routing Problem with Simultaneous Pickup and Delivery (VRPSPD). We assume that the quantities to be delivered are fixed and known in advance, whereas the quantities to be picked up are given as random variables, following finite discrete probability distributions. For the stochastic model the recourse point of view is chosen. As corrective action an additional route is determined to collect the pickup amounts that have been left at failure nodes. The distance traveled according to the route determined in the first stage by known delivery quantities plus the expected distance traveled according to the additional corrective route is subject to minimization.

The Integer L-Shaped method, introduced 1993 by Laporte and Louveaux [40], is adjusted to solve the present specific SVRP. In general terms the Integer L-Shaped method which is an extension of the L-Shaped method of Van Slyke and Wets [74] for continuous stochastic programs is a branch-and-cut procedure where lower optimality cuts are generated to eliminate feasible but non-optimal solutions. Further, lower bounding functionals (LBFs) are used to improve the efficiency of the algorithm by strengthening the lower bound on the recourse cost associated to partial routes encountered throughout the solution process. The construction of the LBF is due to Jabali, Rei, Gendreau and Laporte [36]. The concept of partial routes originally introduced by Hjorring and Holt in [34] is compared to the generalized counterpart presented in [36] by Jabali, Rei, Gendreau, and Laporte. Different strategies for computing a lower bound on the compensation cost given a partial route are presented and examined. Furthermore risk neutral and risk averse routing decisions are compared.

The thesis is structured as follows. Chapter 2 summarizes basic ideas and models in context of stochastic programming. Risk neutral as well as risk averse models are presented. In Chapter 3 the VRP is addressed. Section 3.1 deals with the definition of the classical deterministic VRP. Two different basic formulations are given. Further the Vehicle Routing Problem with Simultaneous Delivery and Pickup (VRPSDP) and the Generalized Traveling Salesman Problem (GTSP) are introduced. These two problems arise partly in our specific problem and the GTSP needs to be solved several times during the solution procedure. In section 3.2 the stochastic counterpart is formulated. Solution concepts and algorithms for deterministic and stochastic VRPs are recapitulated in section 3.3. The VRP with Simultaneous Delivery and Stochastic Pickup (VRPSDSP) is presented in chapter 4. Section 4.1 precises a literature review on the problem studied. In section 4.2, for the single vehicle case, the VRPSDSP is defined and formulated for the risk neutral as well as for the risk averse routing strategies. In section 4.3 the Integer L-Shaped method and its adaptation to the Single-VRPSDSP is described. Furthermore, the construction of the LBF based on partial routes is displayed. Different aggregation policies for the computation of local and global lower bounds on the compensation costs are presented. Chapter 5 treats of the performance of the implemented algorithm. Extensive tests on several sets of randomly generated scenarios are performed and presented for the risk neutral and the risk averse strategies. This thesis is concluded in section 6 and an outlook on future research is given.

2. Stochastic Programming

In deterministic settings, complete information on problem data has to be available upon solving the problem. However, in most real world applications one is challenged to make decisions under incomplete information. For instance some input data is assumed to be uncertain and parts of the decisions have to be taken prior to the observation of this uncertain data. To model phenomena like this, stochastic programming concepts are applied, since solving a deterministic problem where stochastic parameters are replaced by their expected values can yield poor solutions (cf. [47]). Stochastic programming models are characterized by the way they capture the interplay of making decisions and gaining information. For a detailed introduction into stochastic programming and an overview of basic models, methods and applications of stochastic programming see [62].

2.1. Decision Making under Uncertainty

Including random data enforces the decision maker to call in to question how to define feasibility and how to rank the arising random variables.

For a wide class of stochastic programming problems, also for those in the present thesis, it is assumed that each decision variable $x \in X$ causes an individual random variable $f(x, \omega) : \Omega \to \mathbb{R}$, where Ω is a given probability space, such that the decision maker is forced to select the “best” random variable out of the family

$$\{f(x, \omega) : x \in \mathcal{X}\}. \tag{2.1}$$

Selecting an “optimal“ x is dependent on the choice of evaluation concept. In [52] different possibilities for ranking random variables are introduced. We consider ranking by statistical parameters, more specifically ranking by expectation

$$\min\{\underbrace{\mathbb{E}_\omega[f(x, \omega)]}_{Q_{\mathbb{E}}(x)} : x \in \mathcal{X}\}. \tag{2.2}$$

To base the optimization on the expected value, only, completely neglects the variability of the random variable and risk is not taken into account. With regard to scenarios having small probability but high costs, one can base the minimization on a weighted sum of expected value and some risk measure $Q_{\mathcal{R}}(x)$

$$\min\{Q_{\mathbb{E}}(x) + \rho \cdot Q_{\mathcal{R}}(x) : x \in \mathcal{X}\}, \tag{2.3}$$

where $\rho \geq 0$ is a fixed parameter. Models in form of (2.3) are called Mean-Risk Models. Risk measures pertinent to the present thesis are the Excess Probability (EP) and the Expected Excess (EE). They are chosen since being practically meaningful and algorithmically tractable at the same time. The EP is the probability of exceeding a prescribed threshold $\eta \in \mathbb{R}$, i.e.

$$Q_{\mathbb{P}^\eta}(x) := \mathbb{P}[\{\omega : f(x, \omega) > \eta\}]. \tag{2.4}$$

The EE also takes into account the amount of exceeding $\eta \in \mathbb{R}$:

$$Q_{\mathbb{D}^\eta}(x) := \mathbb{E}[\max\{f(x, \omega) - \eta, 0\}]. \tag{2.5}$$

In particular in banking and finance there exist many more measures quantifying risk, such as the “Value-at-Risk"and the "Conditional Value-at-Risk. For detailed descriptions, properties and applications see [58] and [60].

Concerning feasibility, the two main solution concepts in stochastic programming are Chance Constrained Programming (CCP) and Stochastic Programming with Recourse (SPR). The difference between these two approaches mainly lies in the fact that in some applications one is able to compensate for first-stage decisions to maintain feasibility and in others, corrective actions cannot be modeled reasonably or they simply do not exist. In circumstances described the latter CCP is used to guarantee feasibility 'as often as possible'. Unexpected extreme events may cause constraint violation, that can almost never be avoided. In CCP the problem is solved under the condition that constraints are satisfied with some probability, i.e. constraint violation is allowed up to a given tolerance. SPR is taken into consideration if the problem involves a two-stage decision scheme, often consisting of some first-stage decision before and a recourse action after realization of the random data. Second-stage variables ensuring feasibility are defined, such that first-stage decisions can be compensated. SPR then aims at a first-stage decision such that the expected value of the costs caused by the decisions in both stages becomes minimal.

Robust Optimization (RO) could be resorted to if one is interested in worst-case analysis. Using concepts based on RO result in more restrictive and pessimistic optimization models. In RO feasibility has to be guaranteed for all realizations of the random variable and no constraint violation is allowed. For a comprehensive account of RO see Ben-Tal, El-Ghaoui, and Nemirovski [6]. Another conceptual framework is Stochastic Dominance (SD), where the arising random variables are compared to some benchmark random variables. An accessible introduction, deeper insights, and algorithmic experiments are given in [21].

In the present thesis the recourse point-of-view is adopted, i.e., parts of the decisions must be taken with incomplete information and the remaining ones are serving as corrective actions after the complete information has been revealed.

2.2. Two-Stage Stochastic Linear Programs with Recourse

Modeling stochastic programs in a two-stage framework results in an alternating decision and observation scheme. Prior to knowing the realizations of some random variables first-stage or here-and-now decisions x must be taken. After observing the randomness second-stage or wait-and-see decisions y can be made to compensate for possible infeasibility. The goal of two-stage stochastic programming is to choose a single set of actions for the first-stage that minimizes expected costs for all possible realizations of the random variables.

The general formulation of a two-stage stochastic linear program with recourse is given by:

Definition 2.2.1. *Two-Stage Stochastic Linear Programs with Recourse*

$$\min\{c^T x + q(\omega)^T y : T(\omega)x + W(\omega)y = z(\omega), x \in \mathcal{X}, y \in \mathcal{Y}\} \tag{2.6}$$

where $\mathcal{X} \subseteq \mathbb{R}^{m_1}$, $\mathcal{Y} \subseteq \mathbb{R}^{m_2}$ *are nonempty polyhedra and* $(q, T, W, z)(\omega)$ *is a random vector on some probability space* $(\Omega, \mathcal{A}, \mathbb{P})$ *with values in* $\mathbb{R}^{m_2} \times \mathbb{R}^{m_1 s} \times \mathbb{R}^{m_2 s} \times \mathbb{R}^s$*, whose distribution does not depend on* x*.*

Regarding stochastic programs it is crucial, that the non-anticipativity constraint is fulfilled, which means, that the decision vector x must be independent on the realizations of $\xi := (q, T, W, z)(\omega)$. Note that X and Y might have integer requirements.
The structural analysis of the stochastic programming models is mainly based on the value function ϕ, which is defined as:

$$\phi(s_1, \mathcal{A}, s_2) := \min_{y \in \mathcal{Y}}\{s_1^T y : \mathcal{A}y = s_2\} \tag{2.7}$$

Using the value function ϕ, (2.6) can be rewritten as:

$$\min\{c^T x + \phi(q(\omega), W(\omega), z(\omega) - T(\omega)x) : x \in \mathcal{X}\} \tag{2.8}$$

The value function ϕ can be seen as given by a parametric optimization problem with parameters in the objective function, the constraint matrix, and in the right-hand-side of the constraints. For the problem studied in the present thesis we just have randomness in the right-hand-side. Therefore, with $z(\omega)$ as the only random variable the value function ϕ reduces to:

$$\phi(z(\omega) - Tx) := \min_{y \in \mathcal{Y}} \{q^T y : Wy = z(\omega) - Tx\} \tag{2.9}$$

Basic properties of ϕ depending on the random elements of vector ξ are summarized and discussed by Ruszczyński and Shapiro in chapter 2 of [65].

Defining the function $f(x, \omega)$ from (2.1) as:

$$f(x, \omega) := c^T x + \phi(z(\omega) - Tx) \tag{2.10}$$

and minimizing the expected value

$$\min\{\mathbb{E}_\omega[f(x, \omega)] : x \in \mathcal{X}\} \tag{2.11}$$

one obtains a well defined optimization problem. The computation of the expected value requires to calculate a multi-dimensional integral, where also, in general, the integrand is discontinuous. However, when assuming a discrete probability distribution with finitely many realizations z_ω and probabilities $\pi_\omega, \omega \in \Omega, |\Omega| = S$, the minimization in (2.11) can be equivalently expressed by:

$$\begin{aligned} \min\{c^T x + \sum_{\omega \in \Omega} \pi_\omega q^T y_\omega \ : \ & Tx + Wy_1 && && = z_1 \\ & Tx && + Wy_2 && = z_2 \\ & \vdots && \ddots && \vdots \\ & Tx && \quad + Wy_S && = z_S \\ & && x \in \mathcal{X}, y_\omega \in \mathcal{Y}, \omega = 1, ..., S\} \end{aligned} \tag{2.12}$$

Algorithms typically exploit the special form of the constraint matrix of problem (2.12). The block structure is depicted in figure (2.1).

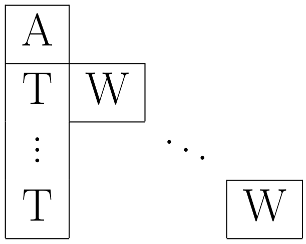

Figure 2.1.: Block structure of constraint matrix

Decomposition turns out an adequate method for solving these problems. For example, there is the L-shaped decomposition by Van Slyke and Wets [74]. In the course of the algorithm, cutting planes are generated to approximate both the objective and the constraints. Decomposition occurs by the scenario-wise creation of the cuts.

Another algorithmic advantage can be exploited when writing the non-anticipativity constraints in explicit fashion. More specifically, for each scenario a set of copies of the first-stage decision variables is established and an additional constraint is added stating that the copies have to be all equal. The resulting block structure is depicted in figure (2.2).

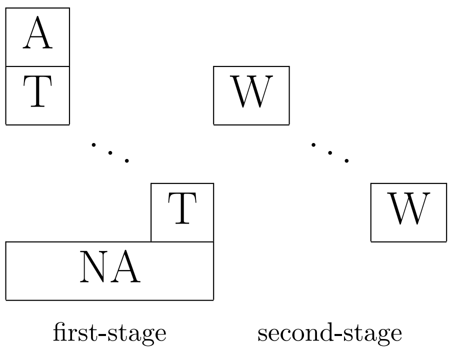

Figure 2.2.: Block structure of reformulation

The additional constraint is treated by Lagrangean relaxation, such that problem (2.12) decomposes into S single-scenario problems. An implementation of an algorithm based on dual decomposition can be found in [49], for theoretical details see [11].

Regarding Mean-Risk Models using EP and EE as risk measures and still concerning a finite discrete probability distribution, the resulting optimization models are equivalent to the following mixed integer linear programs.

For the EP model additional binary variables Θ_ω for each scenario $\omega \in \Omega$ may be introduced. Choosing a constant M, such that

$$M > \sup\{c^T x + \phi(z_\omega - Tx) : x \in \mathcal{X}, \omega = 1, ..., S\}, \tag{2.13}$$

where it is assumed, that the supremum on the right-hand-side is bounded, the two-stage linear stochastic EP-model can be defined as:

Model 2.2.2. *Excess Probability*

$$\begin{aligned} &\min\{Q_{\mathbb{E}}(x) + \rho \cdot Q_{EP}(x) : x \in \mathcal{X}\} \\ = &\min\{c^T x + \sum_{\omega \in \Omega} \pi_\omega q^T y_\omega + \rho \cdot \sum_{\omega \in \Omega} \pi_\omega \Theta_\omega \quad : \\ &\qquad Tx + W y_\omega = z_\omega \\ &\qquad c^T x + q^T y_\omega - M \cdot \Theta_\omega \leq \eta \\ &\qquad x \in \mathcal{X}, y_\omega \in \mathcal{Y}, \Theta_\omega \in \{0,1\}, \omega = 1, ..., S\} \end{aligned} \tag{2.14}$$

The above model can be interpreted as: Given a specific scenario, it is counted and penalized by the objective function whenever the corresponding objective value exceeds the predefined threshold $\eta \in \mathbb{R}_+$.

Considering the Mean-Risk model with EE, further continuous variables ν_ω for each scenario $\omega \in \Omega$ are imposed. Using continuous variables also the amount of exceeding $\eta \in \mathbb{R}_+$ can be taken into account.

Model 2.2.3. *Expected Excess*

$$\begin{aligned}
&\min\{Q_{\mathbb{E}}(x) + \rho \cdot Q_{EE}(x) : x \in \mathcal{X}\} \\
= &\min\{c^T x + \sum_{\omega \in \Omega} \pi_\omega q^T y_\omega + \rho \cdot \sum_{\omega \in \Omega} \pi_\omega \nu_\omega \quad : \\
&\qquad Tx + W y_\omega = z_\omega \\
&\qquad c^T x + q^T y_\omega - \nu_\omega \leq \eta \\
&\qquad x \in \mathcal{X}, y_\omega \in \mathcal{Y}, \nu_\omega \in \mathbb{R}_+, \omega = 1, ..., S\}
\end{aligned} \tag{2.15}$$

Due to the fact, that the block structure of the constraint matrix is maintained, one is able to apply all techniques, which are developed for the risk neutral model.

3. Routing under Uncertainty

Routing problems appear in a widespread area of real world applications, like telecommunication, dispatching of goods, public transportation including airplanes, and navigation. In this thesis we are focusing on a stochastic extension of a special class of vehicle routing problem dealing with the determination of an optimal route given a single vehicle with limited capacity to deliver and collect various goods to and from a set of customers. Therefore this section will provide fundamentals regarding VRPs that are important for comprehension of the present work.

3.1. Deterministic Routing

In deterministic settings it is considered that all input data is known a priorily. Nevertheless, routing problems, especially VRPs, are hard to solve and belong to the NP-hard complexity class.[1] Due to the fact that the VRP and its various extensions are of great interest for logistics, distribution, and transportation, it is one of the most studied combinatorial optimization problem.

The classical VRP, known as Capacitated VRP (CVRP), is defined on a complete undirected graph $G = (V, E)$, where $V := \{0, 1, .., n\}$ denotes the customers, including one special node, called the depot, indexed with 0 and $E := \{(i, j)|i, j \in V, i < j\}$ indicating all edges between customers and depot. Each customer is assigned one positive number d_i representing the delivery

[1]VRPs are multiple traveling salesman problems with additional routing constraints and therefore NP-hard (see e.g.[15]).

(or pickup) amount. Further, a fleet of homogeneous vehicles $1, ..., K$ is given, each having a capacity limitation Q. The CVRP consists of determining (a maximum of) K routes starting and ending at the depot, servicing each customer exactly once, complying with the capacity limitation and minimizing the overall traveled distance. To formulate the CVRP within the framework of integer programming, binary variables x_{ij} are introduced to indicate, whether a vehicle travels between customer i and j or not.

For $S \subset V$, let $\delta(S) := \{(i,j)|i \in S, j \notin S \text{ or } i \notin S, j \in S\}$, if $S = \{i\}$, then we simply write $\delta(i)$ instead of $\delta(\{i\})$. Further let $r(S)$ be the minimum number of vehicles needed to serve the customers of a subset S of customers. Then an integer formulation of the CVRP is given by:

Model 3.1.1. *Capacitated Vehicle Routing Problem*

$$\min \sum_{(i,j)\in E} c_{ij}x_{ij} \tag{3.1}$$

$$s.t. \sum_{(i,j)\in\delta(i)} x_{ij} = 2,\ i \in V/\{0\} \tag{3.2}$$

$$\sum_{(i,j)\in\delta(0)} x_{ij} \leq 2K, \tag{3.3}$$

$$\sum_{(i,j)\in\delta(S)} x_{ij} \geq 2r(S),\ S \subseteq V/\{0\}, S \neq \emptyset \tag{3.4}$$

$$x_{ij} \in \{0,1\},\ (i,j) \in E, i,j \neq 0 \tag{3.5}$$

$$x_{ij} \in \{0,1,2\},\ (i,j) \in \delta(0) \tag{3.6}$$

Constraints (3.2) ensure that each customer is visited exactly once. A maximum number of K vehicles leave the depot, because of constraint (3.3). If one imposes, that exactly K vehicles have to serve the costumers, than inequality (3.3) changes to an equality. Constraints (3.4) prohibit the formation of subtours. Rather than writing the subtour elimination constraints explicitly, they are only generated when needed. This is due to the fact that the number of subtour elimination constraints is exponential.

An alternative formulation which is amenable to column generation can be defined as follows: Let $\mathcal{R}$ be the set of all feasible routes that is the set of routes starting from the depot, visiting a subset $S \subseteq V/\{0\}$ of customers, such that $\sum_{i \in S} d_i \leq Q$ and returning to the depot. Further, let c_r be the cost of route $r \in \mathcal{R}$ given by the sum of the costs of the arcs belonging to route r. Using a binary variable z_r for each route $r \in \mathcal{R}$, taking value 1 if and only if route r belongs to the solution, the CVRP can be modeled as:

Model 3.1.2. *Capacitated Vehicle Routing Problem*

$$\min \sum_{r \in \mathcal{R}} c_r z_r \tag{3.7}$$

$$s.t. \sum_{r \in \mathcal{R}} y_{ir} z_r = 1, \; i \in V/\{0\} \tag{3.8}$$

$$\sum_{r \in \mathcal{R}} z_r \leq K, \tag{3.9}$$

$$z_r \in \{0,1\}, \; r \in \mathcal{R} \tag{3.10}$$

The coefficients $y_{ir} \in \{0,1\}$ indicate, whether customer i is visited along route r or not, such that constraints (3.8) ensure that each customer is visited exactly once. Constraint (3.9) states that the number of routes is at most K. Here, again the inequality is changed to an equality, if the number of routes should be exactly K.

In practice it is common to replace the partitioning constraints (3.8) by the following covering constraints:

$$\sum_{r \in \mathcal{R}} y_{ir} z_r \geq 1, \; i \in V/\{0\} \tag{3.11}$$

By virtue of having an extremely large number of feasible routes, the linear relaxation of model (3.1.2), where constraints (3.8) are replaced by constraints (3.11) is solved considering a limited subset of columns $\mathcal{R}' \subseteq \mathcal{R}$. To generate further promising routes that are routes having negative reduced costs and to prove optimality, a pricing problem is repeatedly solved. Optimality is achieved, when no more routes with negative reduced costs are found.

Replacing equality (3.8) by inequality (3.11) has no impact on the optimal solution, since no optimal solution visits any customer more than once. Using set covering constraints has the advantage to find feasible solutions easier. Moreover, they are preferable to set partitioning constraints, because when relaxing the integrality constraints, a set covering model has a smaller dual space, cf. [18].

As mentioned at the beginning the VRP has a tremendous variety of applications. Using additional side constraints, the VRP can be easily dilated. For a detailed overview of the most common VRP extensions and solution approaches based on exact methods, heuristics and metaheuristics see [72], [15]. Other ideas based on approximations and reformulations are for example the K-tree [2] approximation presented by Fisher [25] or the reformulation as multi-commodity flow problem defined in [3].

3.1.1. Vehicle Routing with Simultaneous Delivery and Pickup

In comparison to the CVRP, the VRP with Simultaneous Delivery and Pickup (VRPSDP) involves supplementary routing constraints. Instead of one positive number per customer, two positive numbers d_i and p_i are assigned, representing the delivery and pickup amount at customer i. In contrast to the CVRP, the orientation of the route is now important to know and how the vehicle load changes while proceeding the route. Therefore further continuous nonnegative variables D_{ij} and P_{ij} need to be introduced, indicating the amount of delivery load and collected load carried along arc (i, j). The following mixed-integer programming formulation is taken from [18].

[2] A K-tree is a generalization of a spanningtree. Given a graph with $n+1$ nodes, a K-tree is defined as set of $n+K$ arcs spanning the graph

Model 3.1.3. *Vehicle Routing Problem with Simultaneous Delivery and Pickup*

$$\min \sum_{(i,j)\in E} c_{ij}x_{ij} \tag{3.12}$$

$$s.t. \sum_{j\in V} x_{ij} = 1,\ i \in V/\{0\} \tag{3.13}$$

$$\sum_{j\in V/\{0\}} x_{0j} \leq K, \tag{3.14}$$

$$\sum_{j\in V} x_{ij} = \sum_{j\in V} x_{ji},\ i \in V \tag{3.15}$$

$$\sum_{j\in V} P_{ij} - \sum_{j\in V} P_{ji} = p_i,\ i \in V/\{0\} \tag{3.16}$$

$$\sum_{j\in V} D_{ji} - \sum_{j\in V} D_{ij} = d_i,\ i \in V/\{0\} \tag{3.17}$$

$$P_{ij} + D_{ij} \leq Qx_{ij},\ (i,j) \in E \tag{3.18}$$

$$P_{ij}, D_{ij} \geq 0,\ (i,j) \in E \tag{3.19}$$

$$x_{ij} \in \{0,1\},\ (i,j) \in E \tag{3.20}$$

Constraints (3.13) provide for visiting each customer exactly once and constraint (3.14) enforces that this is done by a maximum number of K vehicles. In this formulation no subtour elimination constraints like constraints (3.4) in model 3.1.1 are required, because of constraints (3.15), (3.16), and (3.17). These are flow conservation constraints on the number of vehicles and on the amounts of pickup and delivery load. Constraints (3.18) ensure that the vehicle capacity is not exceeded.

Regarding exact solution methods for the VRPSDP Dell'Amico, Righini and Salani presented in 2006 the only exact algorithm so far, cf [18]. They proposed a branch and price algorithm based on a set covering formulation of the masterproblem. The largest instance that is solved to optimality, has 40 customers. A survey and detailed description on heuristic approaches is given in [8]. Variants of the VRPSDP like clustered backhauls, mixed linehauls and backhauls, divisible delivery and pickup, as well as solution approaches based on exact methods, heuristics and metaheuristics are presented in [57].

3.1.2. Generalized Traveling Salesman Problem

The Generalized Traveling Salesman Problem (GTSP) is a variant of the well known Traveling Salesman Problem (TSP) in which the nodes are partitioned into clusters and the salesman has to visit at least one node for each cluster. In what follows the notation introduced by Fischetti, González, and Toth in [24] is used.

Given a complete undirected graph $G = (V, E)$ with node set $V := \{0, 1, ..., n\}$, edge set $E := \{(i, j)|i, j \in N, i \neq j\}$ with associated arc costs c_{ij} and, a proper partition $C_1, ..., C_m$ of N. For each $S \subseteq N$, let

$$E(S) := \{(i, j) \in E|i \in S, j \in S\},$$

$$\delta(S) := \{(i, j) \in E|i \in S, j \notin S\},$$

For $v \in N$ they write $\delta(v)$ instead $\delta(\{v\})$, and $C_{h(v)}$ denotes the cluster containing v. With these notations, they formulate an integer linear programming model for the GTSP as follows. Let $x_{ij} = 1$ if edge $(i, j) \in E$ is chosen in the optimal solution, $x_{ij} = 0$ otherwise. Further let $y_v = 1$ if node $v \in N$ is visited, $y_v = 0$ otherwise.

Model 3.1.4. *Generalized Traveling Salesman Problem*

$$\min \sum_{(i,j)\in E} c_{ij} x_{ij} \tag{3.21}$$

$$s.t. \sum_{(i,j)\in\delta(v)} x_{ij} = 2y_v,\ v \in N \tag{3.22}$$

$$\sum_{v\in C_h} y_v \geq 1,\ h = 1, ..., m \tag{3.23}$$

$$\sum_{(i,j)\in\delta(S)} x_{ij} \geq 2(y_i + y_j - 1),\ S \subset N, 2 \leq |S| \leq n - 2, \tag{3.24}$$

$$i \in S, j \in N \backslash S \tag{3.25}$$

$$x_{ij} \in \{0, 1\},\ (i, j) \in E, \tag{3.26}$$

$$y_v \in \{0, 1\},\ v \in N \tag{3.27}$$

Constraints (3.22) are the node degree constraints which impose that the number of edges incident with a node is either 2 if y_v is visited or 0 otherwise. To ensure that at least one node in each cluster is visited, constraints (3.23) are established. Note that when the arc costs satisfy the triangle inequality, constraints (3.23) are equivalent to

$$\sum_{v \in C_h} y_v = 1,\ h = 1, ..., m, \tag{3.28}$$

indicating that exactly one node in each cluster is visited. Constraints (3.24) are required to guarantee connectivity, which means each cut separating two visited nodes i and j must be crossed at least twice.

An exact algorithm based on a branch-and-cut procedure is presented by Fischetti, González, and Toth in [24]. A problem reduction algorithm deleting redundant vertices and edges are given by Gutin and Karapetyan [32]. Using the fact that the GTSP can be converted to an equivalent TSP with the same number of vertices, efficient TSP solvers can be established (cf. [5], [43], [44], [53]). Heuristics for the GTSP can be found for example in [33], [66], [67].

3.2. Capturing Uncertainty

Due to the fact that in almost every process regarding route planning input data is not known prior to the solution process, one has to deal with randomness to capture uncertainty. To this purpose stochastic programming concepts are adapted to formulate reasonable models which reflect real world applications satisfactorily.

Stochastic Vehicle Routing Problems (SVRPs) are extensions of the deterministic VRPs in which some data is assumed to be random and following a given probability distribution. Regarding SVRPs it is no longer required to satisfy the constraints for all realizations of the random variables, and one has to define new feasibility and optimality conceptions. In comparison to deterministic VRPs, they are considerably more difficult to solve, since they combine the characteristics of stochastic and integer programs. For a detailed introduction to the theory of linear and integer programming cf. [63].

Concerning uncertain input data in the field of vehicle routing, the three most common causes are: stochastic demands, stochastic customers and stochastic travel times. The best studied of all SVRP is the VRP with Stochastic Demands (VRPSD), in which delivery (or pickup) demand d_i is replaced by a random variable ξ_i. In the case of stochastic customers it is assumed that the customers are only present with some probability, but have deterministic demands. Stochastic times are assumed, whenever one has to deal with uncertain travel or service times. In addition to the classical capacity constraints, one imposes typically also duration and/or time window constraints.

Modeling SVRP using stochastic programming concepts, one can adapt the two main branches presented in section 2.1, CCP and SPR. Using CCP the distance traveled is minimized while controlling the probability of route failure. A route failure is referred to as for example violation of capacity limitations or time window restrictions. However the cost of such a failure is not taken into account. In [70] Stewart and Golden showed that under appropriate assumptions, solving CCP models involves the same level of difficulty as solving a deterministic VRP with the same parameters.

In the framework of SPR first an a priori solution is computed, then the realization of the random variable becomes known and in a second stage, a recourse or corrective action is applied to the first stage solution to compensate for possible failures. A common corrective action for VRPSD is a return trip to the depot, whenever the vehicle runs out of capacity. After replenishing (or unloading in the pickup case) the planned route is either resumed at the point of failure or a new routing sequence for the remaining customers is planned. An alternative strategy to reduce the expected cost of corrective action is to break the planned vehicle routes at predefined or strategical points, which are based on the anticipated customer demands. In SPR the expected distance traveled is subject to minimization. Since the potential location of a failure has a significant impact on the expected route length, two routes with the same probability of failure and the same distance might have quite different expected distance values.

In [70] the two mentioned frameworks for solving a multiple VRPSD are compared by Stewart and Golden. They concluded that if the route failure penalty cost is known, SPR models produce lower costs than CCP.

3.2.1. VRP with Stochastic Demands

Taking the recourse point of view, the VRPSD can be formulated as a two-stage stochastic program, where in a first-stage having incomplete information an a priori route is constructed and in a second-stage compensation strategies are performed, when route failures occur, which cause additional costs. In the VRPSD each customer $i = 1, ..., n$ is assigned a random variable ξ_i, representing its stochastic demand. To prevent from systematically failing of routes because of unbalanced assigned customers with respect to expected demands, it is common to require that the expected demand of a route $(0, i_1, ..., i_t, 0)$ does not exceed the vehicle capacity:

$$\sum_{j=1}^{t} \mathbb{E}(\xi_{i_j}) \leq Q. \tag{3.29}$$

In contrast to the deterministic version, because of return trips to the depot, the binary variable x_{ij} can now also take the value 2 for arcs connected to the depot that is $x_{0j} \in \{0, 1, 2\}$ for $j = 1, ..., n$. The compensation costs $\mathcal{Q}(x)$ for a first-stage solution x are added to the objective function.

Model 3.2.1. *Vehicle Routing Problem with Stochastic Demands*

$$\min \sum_{i<j} c_{ij} x_{ij} + \mathcal{Q}(x) \tag{3.30}$$

$$s.t. \sum_{i=1}^{n} x_{0i} = 2K,\ i \in V/\{0\} \tag{3.31}$$

$$\sum_{i<k} x_{ij} + \sum_{j>k} x_{ij} = 2, k \in V/\{0\} \tag{3.32}$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - \left\lceil \sum_{i \in S} \mathbb{E}(\xi_i)/D \right\rceil,\ S \subseteq V/\{0\}, 2 \leq |S| \leq n-1 \tag{3.33}$$

$$x_{ij} \in \{0,1\},\ 1 \leq i < j \leq n \tag{3.34}$$

$$x_{0j} \in \{0,1,2\},\ j \in V/\{0\} \tag{3.35}$$

Neglecting the $\mathcal{Q}(x)$ term this model is that of a deterministic CVRP, in which customer demands are $\mathbb{E}(\xi)$. Constraints (3.31) and (3.32) are the node degree constraints. To ensure that the expected demand of any route does not exceed the vehicle capacity and to prevent from subtours constraints (3.33) are imposed.

Formulating the compensation costs $\mathcal{Q}(x)$ in terms of decision variables and linear relationships is quite complex. However, for a given a priori solution x, $\mathcal{Q}(x)$ is separable in the routes and can easily be computed under some assumptions, depending on the recourse strategy and on stated problem features and characteristics. Furthermore, in stochastic settings also route orientation needs to be captured, which is irrelevant for the deterministic case. In [42] it is described how to compute the compensation cost for divisible goods.

In [22] an overview concerning recourse strategies, stochastic mathematical programming formulations, and solution frameworks is given by Dror, Laporte and Trudeau. Depending on the number of vehicles considered, exact solution methods are either based on a branch-and-price or branch-and-cut framework. In [12] a branch-and-price algorithm is proposed by Christiansen and Lysgaard and Laporte, Louveaux, and van Hamme presented a branch-and-cut algorithm based on the Integer L-Shaped method in [42].

3.3. Solution Concepts and Algorithms

Despite the fact that there are only finitely many feasible points,[3] solving VRPs, deterministic or stochastic, remains a formidable task. Exact solution methods very quickly come to their limits, such that for almost all real world instances only heuristics or metaheuristics are practical. In what follows some of the well-established exact algorithms, heuristics and metaheuristics are summarized.

3.3.1. Exact Algorithms

The two most popular exact solution frameworks are based on Column Generation (CG) embedded in a Branch&Bound scheme and on a Branch&Cut (B&C) procedure, respectively. Depending on the problem studied one of these technics outperformes the other. For instance having a vehicle fleet of more than 5 vehicles, it is common to use a CG based algorithm.

Solving VRPs using CG one operates on a Restricted Masterproblem (RMP) that comes up by restricting model (3.1.2) to a limited set $\mathcal{R}' \subseteq \mathcal{R}$ of feasible routes. To guarantee feasibility the RMP must be properly initialized. Therefore either a dummy column is added, representing a fictitious, infeasible route visiting all customers, but having extremely high costs, such that it will never appear in the optimal solution, or feasible routes are generated with the aid of problem specific heuristics. To determine auspicious routes a so called pricing problem is solved, which identifies routes with least negative reduced costs that are added to the RMP. Solving the pricing problem is the challenging task in the CG procedure. Due to this a reformulation of the pricing subproblem as Shortest Path Problem with Resource Constraints (SPPRC) can be carried out. The SPPRC is solved using dynamic programming techniques[4]. The CG

[3] This is due to the fact that only binary variables or variables contained in the set $\{0, 1, 2\}$ are necessary to formulate the LP, thus only finitely many combinations of variables exist.

[4] In general, a labeling algorithm is performed using domination rules, state space relaxations and bidirectional search refinements, for a detailed description see [18] and [19].

algorithm is embedded in a Branch&Bound (B&B) framework, hence this algorithm is also called Branch&Price (B&P). A didactic introduction to the use of the CG technique is given in chapter 1 of [19]. Literature concerning B&P approaches for the VRP can be found for example in [18] for the VRP with Simultaneous Distribution and Collection and for the CVRP with stochastic demands in [12].

In B&C procedures the initial problem is relaxed in the sense that integrality constraints, subtour elimination constraints and possible further troublesome constraints are neglected. During the solution process feasibility cuts are added to cut off infeasible solutions. In [48] a B&C algorithm for the CVRP is presented by Lysgaard, Letchford, and Eglese.
Regarding stochastic settings formulated as two-stage stochastic programs with recourse, also objective cuts are added that bound the compensation costs, which are also relaxed in the initialization and replaced by a general lower bound. The *Integer L-Shaped* method, introduced 1993 by Laporte and Louveaux [40], is one of the most promising B&C procedures, when solving stochastic integer programs with recourse. The efficiency of a B&C method relies mainly on finding a good lower bound for the problem studied. It is common to determine lower bounds only by solving the LP-Relaxation of the masterproblem, which of course could be very weak depending on the cuts added and on the initial integer formulation.

Since the *Integer L-Shaped* method is chosen as solution framework for the problem studied in this thesis, the method will be described in detail. In the following, we use the notation introduced in [40] by Laporte and Louveaux. Let X be a set, where $X = \bar{X} \cap \{0,1\}^n$ and $\bar{X}$ is the polytope defined by the set of constants in X, the following problem is called *Current Problem*:

Model 3.3.1. *Current Problem*

$$\min c^T x + \Theta \tag{3.36}$$

$$s.t.\ Ax = b \tag{3.37}$$

$$D_k x \geq d_k,\ k = 1, ..., s \tag{3.38}$$

$$E_l x + \Theta \geq e_l,\ l = 1, ..., t \tag{3.39}$$

$$0 \leq x \leq 1, \Theta \in \mathbb{R} \tag{3.40}$$

Constraints (3.38) are used to approximate the feasible region $\bar{X}$, they are referred to as *feasibility cuts*. At the initial stage of the problem $\bar{X}$ is relaxed, during the solution procedure feasibility cuts are only added when needed, i.e. to cut off infeasible solutions. A set of feasibility cuts is said to be valid if there exists some finite value s, such that $x \in \bar{X}$ if and only if $\{D_k x \geq d_k,\ k = 1, ..., s\}$ (cf. [40] p. 135).

The recourse costs $\mathcal{Q}(x)$ are approximated using the variable Θ along with constraints (3.39), which are referred to as *optimality cuts*. A set of optimality cuts is said to be valid if for all $x \in X, (x, \Theta) \in \{(x, \Theta) | E_l x + \Theta \geq e_l,\ l = 1, ..., t\}$ implies $\Theta \geq \mathcal{Q}(x)$ (cf. [40] p. 135). Note that, in practice, only for the optimal (x^*, Θ^*) it is required to have $\Theta \geq \mathcal{Q}(x)$ and not for all feasible (x, Θ).

In general terms the Integer L-Shaped method is described as follows:

Algorithm 3.3.2. *Integer L-Shaped*

Step 0 (Initialization)

Set $\nu = 0, t = 0, s = 0,$
$\bar{z} = +\infty$
$\Theta = -\infty$ *or any valid general lower bound* L.
Define the first pendant node as the initial current problem.

Step 1 (Selection)

Using a selection criterion, select a pendant node,
if *there is none* ***STOP***.

Step 2 (Separation)

2.1 $\nu = \nu + 1$

2.2 *Solve the current problem,*
if *the current problem has no feasible solution* ***then*** *fathom the node and go to* ***Step 1****,*
else *let* (x^ν, Θ^ν) *be the optimal solution to the problem.*

2.3 *Search for violated constraints of type (3.38),*
if *one is found* ***then*** *add one feasibility cut (3.38) to the current problem, set* $s = s + 1$ *go to* ***2.2****.*
if $c^T x^\nu + \Theta^\nu > \bar{z}$ ***then*** *fathom the node and go to* ***Step 1****.*

2.4 *Search for violated integrality constraints,*
if *one is found* ***then*** *go to* ***Step 3****,*
else *solution* x^ν *is feasible.*

2.5 *Compute* $\mathcal{Q}(x^\nu), z^\nu = c^T x^\nu + \mathcal{Q}(x^\nu)$*, set* $\bar{z} = \min\{\bar{z}, z^\nu\}$*,*
if $\Theta^\nu \geq \mathcal{Q}(x^\nu)$ ***then*** *fathom the node and go to* ***Step 1****,*
else *add an optimality cut (3.39), set* $t = t + 1$ *go to* ***2.2****.*

Step 3 (Branching)

Using a branching criterion, create two new nodes, append them to the list of active nodes and go to ***Step 1****.*

Solving stochastic vehicle routing problems which are formulated as stochastic programs with recourse by means of the Integer L-Shaped method, the feasibility and optimality cuts take on special forms. In the initial stage only degree constraints of each vertex of the underlying graph are taken into account. Feasibility cuts appear as subtour elimination and capacity constraints, they are added in the separation step 2.3. Since having a finite number of routes, assume that these are indexed by r and define

$$S_r := \{x_{ij} : x_{ij} = 1 \text{ in the } r\text{th route}\},$$

where x_{ij} is a binary variable indicating if vehicle travels between customer i and j. Further, let Θ_r be the cost of recourse for route r that are costs due to compensation strategies. The optimality cuts which are added in Step 2.5 can be defined as:

$$\Theta \geq (\Theta_r - L)\left(\frac{\sum_{x_{ij} \in S_r} x_{ij} - (n-1)}{2}\right) + L, \tag{3.41}$$

where n is the number of customers. Each cut is active at one feasible route only. Hence a result, constraints (3.41) only bound the value of recourse associated with the feasible solutions that were used to create them. However, note that the cut for a feasible route is active in more fractional solutions than the standard optimality cut, thus cuts of the form (3.41) are stronger than the cuts used in the generic L-Shaped method (cf. [34]).

One can also impose an optimality cut which is only used to eliminate from further considerations the r-th feasible solution:

$$\sum_{x_{ij} \in S_r} x_{ij} \leq |S_r| - 1. \tag{3.42}$$

Numerically, this cut is more stable than cut (3.41), since it is composed of coefficients equal to one. However, it provides no information on the value of recourse. Using exclusively cuts of the form (3.42), the number of active subproblems may increase enormously. This is due to the fact that the quality of bound $c^T x^\nu + \Theta^\nu$ in the separation step will be poor whenever x^ν is infeasible. Consequently, with a poor general lower bound, the Integer L-Shaped algorithm will tend to enumerate feasible solutions. Regarding the determination of an appropriate general lower bound, it is crucial to keep balance between computation time and quality. Given a problem that is studied, searching for a good bound may be of no avail or highly time consuming. Another possibility to avoid the enumeration of feasible solutions, is to provide better approximations of the recourse cost using other lower bounding functionals.

In [34] Hjorring and Holt present a new type of optimality cut, referred to as general optimality cut. Since many feasible solutions might have a number of x_{ij} values in common, they define a particular type of subsets of x_{ij}. Let $x_{ij} = 1$, for all $x_{ij} \in S_p$ and $x_{ij} \notin S_p$ are unspecified, except for feasibility considerations, and Θ_p a corresponding lower bound on the expected second-stage value. Define the general optimality cut as

$$\Theta \geq (\Theta_p - L)\Big(\sum_{x_{ij} \in S_p} x_{ij} - (|S_p| - 1)\Big) + L, \tag{3.43}$$

where $|S_p|$ is the cardinality of S_p.
The advantage of general optimality cuts is an improvement of the lower bound on Θ for all solutions S_r where $\sum_{x_{ij} \in S_r} x_{ij} = |S_p|$, if $\Theta_p > L$. However, for solutions in which $\sum_{x_{ij} \in S_r} x_{ij} < |S_p|$, this cut will not be active.

The number of cuts required to prove optimality depends on the gap between the lower bound from the LP relaxation and the best upper bound. Further, for the vehicle routing problem with stochastic demands also the geographical and demand distributions of the customers have a main impact on the number of cuts.

3.3.2. Heuristics and Metaheuristics

Heuristics and metaheuristics are used to determine acceptable solutions in reasonable computing time. Especially for big instances these solution methods are indispensable. Contrary to metaheuristics that try to explore the most promising regions of the solution space, the search space which is investigated by heuristics is quite limited. Most heuristics can be easily extended to account for various constraints occurring when formulating VRPs in different real-life contexts. Metaheuristics require finely tuned parameters, such that it could be difficult to extend them to other situations. Comparing the quality of solutions, metaheuristics outperform the classical heuristics at the price of increased computing time.

Heuristics can be divided into three major groups, route constructive heuristics, route improvement heuristics, and two-phase heuristics.

Constructive heuristics successively assemble routes, while respecting feasibility restrictions and keeping tabs on the solution costs. Strategies used for constructing VRP solutions are either merging existing routes using a savings criterion or gradually assigning vertices to vehicle routes using an insertion technique. The most popular savings algorithm is perhaps the Clarke&Wright algorithm, cf.[14]. Modifications of the Clarke&Wright savings algorithm are presented by Gaskell in [27] and Yellow in [77]. Including circumferential routes and introducing a route shape parameter, they try to generate improved routes also at the end of the solution procedure. Insertion techniques are presented by Mole and Jameson in [51] and by Christofides, Mingozzi, and Toth in [13].

Route improvement heuristics either change the sequencing of customers in vehicle routes taken separately or exchange parts between different routes. Operating on one single route, any improvement heuristic for the TSP can be applied. In [45] Lin defined the λ-opt mechanism, where λ edges are removed and the remaining parts are reconnected. Enhancements are described for example in [46], [54], and [59]. Multiroute edge exchanges are classified by Van Breedam in [73] as string cross, string exchange, string relocation, and string mix. Further exchange schemes are presented by Thompson and Psaraftis [71] and Kindervater and Savelsbergh [37].

Two-phase heuristics can be divided into two groups: cluster-first, route-second and route-first, cluster-second. A cluster-first, route-second method is for example the sweep algorithm (cf. [31], [75], [76]), where initially feasible clusters are formed by rotating a ray centered at the depot. For each of this clusters a TSP is solved to obtain feasible routes. In [26] by Fisher and Jaikumar the clustering is based on solving a generalized assignment problem. The location-based heuristic of Bramel and Simchi-Levi [10] identify the seeds by solving a capacitated location problem. Route-first, cluster-second methods proposed by Beasley [4] start with determining a giant TSP tour not taking into account side constraints. In a second phase the giant tour is decomposed into feasible vehicle routes.

Heuristics summarized above are only used for deterministic settings. However, some of them have been adapted to the stochastic case. In [23] Dror and Trudeau proposed a savings based heuristic for the SVRP. The article [7] by Bertsimas proposes various heuristics for the VRPSD based on lower and upper bounds determined for different strategies and assumptions.

Metaheuristics are strategies that guide the search process to identify promising regions of the solution space and to escape local optima. For this purpose metaheuristics allow deteriorating and even infeasible intermediary solutions while exploring the search space. The complexity of such a procedure ranges from simple local search techniques to involved learning processes. Metaheuristics applied to the VRP are Simulated Annealing (SA), Deterministic Annealing (DA), Tabu Search (TS), Genetic Algorithms (GA), Ant Systems (AS), and Neural Networks (NN). Descriptions of the mentioned metaheuristics are presented in chapter 6 of [72] by Gendreau, Laporte, and Potvin. A structured list of references for various metaheuristics and problem types for the VRP and its extensions are given in [30].

In [29] TS is adapted to the VRP with Stochastic Demands and Customers. A hybrid-metaheuristic for the VRPSD is presented by Bianchi et. al in [9].

4. Single-VRP with Simultaneous Delivery and Stochastic Pickup

In this section we propose a two-stage stochastic programming formulation for the VRP with Simultaneous Delivery and Stochastic Pickup (VRPSDSP). Routing decisions by known delivery quantities make up the first stage, while recourse decisions are made in the second stage when uncertain pickup quantities have been revealed. The overall objective is to optimize the cost of the first stage routing decisions plus the total expected penalty cost incurred in the second stage. Further, risk averse strategies are applied. Here, the Excess Probability and the Expected Excess are chosen as risk measures.

4.1. Literature Review

The recent literature offers just a few contributions to stochastic extensions of the VRPSDP. In [50] it is assumed by Minis and Tatarakis, that the vehicle follows a predefined customer sequence and returns to the depot whenever the vehicle needs to load/unload. A dynamic programming algorithm is proposed to determine the expected routing cost. Hou and Zhou presented in [35] a chance constrained programming model for the Stochastic Vehicle Routing Problem with Uncertain Demand and Travel Times and Simultaneous Pickups and Deliveries, that is solved using a genetic algorithm.

4.2. Modeling: Complete Recourse

The VRPSDSP is defined on a complete undirected graph $G = (N_0, E)$, where $N_0 = \{0, 1, ..., n\}$ is the vertex set and $E = \{(i, j) : i, j \in N_0, i < j\}$ the edge set. The depot is indexed with 0 and the customers are $N = \{1, ..., n\}$. The distance between nodes i and j is denoted by c_{ij}. Each customer demands a quantity d_i of goods to be delivered and an unknown quantity $p_i(\omega)$ of goods to be picked up. We assume that the quantities to be picked up are following a finite discrete probability distribution with realizations $p(\omega) = (p_1(\omega), ..., p_n(\omega)), \omega \in \Omega$ and probabilities $\pi(\omega), \omega \in \Omega$. For the time being, we restrict ourselves to a single vehicle with capacity D. Further we postulate that all delivery quantities can be delivered with a single vehicle, i.e. $\sum_{i=1}^{n} d_i \leq D$.

A feasible solution to the problem consists of a first-stage a priori route where all deliveries and "as much pickups as possible" are made complying with the vehicle capacity limitation visiting each customer exactly once and a second-stage corrective route collecting all pickup quantities which have been left at the customers due to insufficient vehicle capacity. The situation of the vehicle reaching a customer without sufficient capacity to collect the customer's pickup amount is referred to as a route failure.
The objective is to find a pair of an a priori and corrective routes that minimizes the distance traveled in terms of the a priori route plus the expected distance traveled according to the corrective route.

As an application of this special SVRP one can think of forwarding agencies which deliver beverage crates and pick up returned empties. Due to the fact, that the customers behavior of returning their empties is not known with certainty, the quantity of returned empties is assumed to be random.

According to two-stage stochastic linear programming concepts, we formulate our specific SVRP in the sense of a two-stage random integer linear programming model with recourse. Routing decision only on the basis of the known delivery orders make up the first stage x, while second stage decisions y are made for compensation after disclosure of the unknown pickup orders $p(\omega)$. The latter are called recourse decisions. Of course, the second stage decisions y depend on the first stage decision x and on the realization of the random variable $p(\omega)$. The alternating decision and observation scheme is depicted in figure (4.1).

$$\text{decide } x \longmapsto \text{ observe } p(\omega) \longmapsto \text{ decide } y = y(x, p(\omega))$$

Figure 4.1.: Two-stage scheme - Non-anticipativity

To compensate for possible route failures an additional vehicle is sent. More specifically, the additional vehicle also starts and returns to the depot after the complete information on abandoned pickup orders are available and collects the missing units.

Let x_{ij} be a binary variable which is 1 if the vehicle travels between node i and j, 0 otherwise. Let $\mathcal{Q}(x)$ denote the expected recourse cost. Then the model reads:

Model 4.2.1. *Stochastic Vehicle Routing with Simultaneous Delivery and Stochastic Pickup*

$$\min \sum_{i<j} c_{ij} x_{ij} + \mathcal{Q}(x) \tag{4.1}$$

$$\sum_{j=1}^{n} x_{0j} = 2, \tag{4.2}$$

$$\sum_{i<k} x_{ik} + \sum_{j>k} x_{kj} = 2, \ \forall k \in N \tag{4.3}$$

$$\sum_{(i,j)\in S} x_{ij} \leq |S| - 1, \ (S \subset N, 2 \leq |S| \leq n-1) \tag{4.4}$$

$$x = x_{ij} \in \{0, 1\} \tag{4.5}$$

Constraints (4.2) and (4.3) specify the degree of each vertex, whereas constraints (4.4) eliminate subtours. Apart from $\mathcal{Q}(x)$, this model is that of a deterministic traveling salesman problem.

Given a first-stage solution x, the computation of the recourse cost $\mathcal{Q}(x)$ can be carried out individually for the scenarios $\omega = 1, ..., S$. For each scenario one identifies a set of fail-proved nodes, i.e. nodes where there is a route failure for sure. Since the possible route failures also depend on the orientation of the route, the minimum over both orientations is taken:

$$\mathcal{Q}(x) = \min\{\mathcal{Q}^1(x), \mathcal{Q}^2(x)\} \tag{4.6}$$

where $\mathcal{Q}^\delta(x)$ denotes the expected cost of recourse for orientation $\delta = 1, 2$. Let $F^\delta(\omega)$ be the set of fail-proved nodes for scenario ω and orientation δ. Then one has to solve for each scenario and orientation $\delta = 1, 2$ a traveling salesman problem with node set $F^\delta(\omega)$.

Thus, to determine the compensation costs for the risk neutral case one has to solve the following decomposable integer program:

Model 4.2.2. *Recourse Costs*

$$\mathcal{Q}(x) = \min_{y,\delta=1,2} \sum_{\omega\in\Omega} \pi_\omega \sum_{i<j} c_{ij} y_{ij}(\omega) \tag{4.7}$$

$$\sum_{j=1}^{n} y_{0j}(\omega) = 2,\ \omega \in \Omega \tag{4.8}$$

$$\sum_{i<k} y_{ik}(\omega) + \sum_{j>k} y_{kj}(\omega) = 2,\ k \in F^\delta(\omega), \omega \in \Omega \tag{4.9}$$

$$\sum_{(i,j)\in S} y_{ij}(\omega) \le |S| - 1,\ S \subset F^\delta(\omega), |S| \ge 2, \omega \in \Omega \tag{4.10}$$

$$y = y_{ij} \in \{0, 1\} \tag{4.11}$$

Of course, the above problem could be solved from one single minimization, however it is more reasonable to solve each single scenario problem for each orientation individually.

4.2.1. Deterministic Equivalent

Formulating the deterministic equivalent is not straight forward. Including the compensation cost $\mathcal{Q}(x)$ in terms of decision variables in the masterproblem, one has to deal with the determination of an (optimal) route and identification of the failure nodes simultaneously.

Model 4.2.3. *Deterministic Equivalent*

$$\min \sum_{(i,j)\in E} c_{ij}x_{ij} + \sum_{\omega\in\Omega} \pi_\omega \sum_{(i,j)\in E} c_{ij}y_{ij}^\omega \tag{4.12}$$

$$s.t. \sum_{j=1}^{n} x_{0j} = 1, \sum_{j=1}^{n} x_{j0} = 1, \tag{4.13}$$

$$\sum_{j=1}^{n} x_{ij} = 1, \, i \in V/\{0\} \tag{4.14}$$

$$\sum_{j=1}^{n} y_{0j}^\omega \leq 1, \sum_{j=1}^{n} y_{j0}^\omega \leq 1, \tag{4.15}$$

$$F_i^\omega(\sum_{j=1}^{n} y_{ij}^\omega - 1) = 0, \, i \in V/\{0\}, \, \omega \in \Omega \tag{4.16}$$

$$\sum_{j=1}^{n} x_{ij} = \sum_{j=1}^{n} x_{ji}, \, i \in V \tag{4.17}$$

$$\sum_{j=1}^{n} y_{ij}^\omega = \sum_{j=1}^{n} y_{ji}^\omega, \, i \in V \tag{4.18}$$

$$\sum_{j\in V} P_{ij}^\omega - \sum_{j\in V} P_{ji}^\omega = p_i(\omega), \, i \in V/\{0\}, \, \omega \in \Omega \tag{4.19}$$

$$\sum_{j\in V} D_{ji} - \sum_{j\in V} D_{ij} = d_i, \, i \in V/\{0\} \tag{4.20}$$

$$P_{ij}^\omega + D_{ij} \leq Qx_{ij} + (p_i(\omega) - d_i)F_i^\omega, \, (i,j) \in E, \, \omega \in \Omega \tag{4.21}$$

$$(p_i(\omega) - d_i)F_i^\omega \geq 0, \, i \in V/\{0\}, \, \omega \in \Omega \tag{4.22}$$

$$\sum_{(i,j)\in S^\omega} y_{ij}^\omega \leq |S^\omega| - 1, \, S^\omega \subset \{i|F_i^\omega = 1\}, 2 \leq |S^\omega| \leq n-1, \, \omega \in \Omega \tag{4.23}$$

$$P_{ij}^\omega, D_{ij} \geq 0, \, (i,j) \in E, \, \omega \in \Omega \tag{4.24}$$

$$x_{ij}, y_{ij}^\omega, F_i^\omega \in \{0,1\}, \, (i,j) \in E, \, \omega \in \Omega \tag{4.25}$$

4.2.2. Risk Averse Strategies

Concerning risk averse routing strategies, one tries to capture the variability of the random variable and hedge against fluctuations of extreme events that equalize themselves in the mean. The Mean-Risk Models presented in section (2.2) introduced for the EP and EE are taken. The objective function of model (4.2.2) is extended by $\mathcal{Q}_{EP}(x)$ or by $\mathcal{Q}_{EE}(x)$ as defined below. Further, additional constraints (4.27) and (4.30) are imposed, respectively.

Excess Probability:

$$\mathcal{Q}_{EP}(x) = \rho \sum_{\omega \in \Omega} \pi_\omega \Theta_\omega \tag{4.26}$$

$$routeCost + recCost_\omega - M \cdot \Theta_\omega \leq \eta, \ \forall \omega \in \Omega \tag{4.27}$$

$$\Theta_\omega \in \{0, 1\} \tag{4.28}$$

Expected Excess:

$$\mathcal{Q}_{EE}(x) = \rho \sum_{\omega \in \Omega} \pi_\omega \nu_\omega \tag{4.29}$$

$$routeCost + recCost_\omega - \nu_\omega \leq \eta, \ \forall \omega \in \Omega \tag{4.30}$$

$$\nu_\omega \in \mathbb{R} \tag{4.31}$$

To illustrate how routes may change, when concerning different risk measures and routing strategies, lets have a look at the following example. Given a vehicle with capacity 25, 7 customers, and three different scenarios. Delivery and pickup amounts, as well as customer‘s coordinates are depicted in table 4.1.

	π_ω	0	1	2	3	4	5	6	7
x-coord		30	70	80	70	90	40	0	10
y-coord		20	90	70	70	100	20	0	10
d_i		0	5	5	5	5	5	5	5
scenario 1	0.9	0	5	5	5	5	5	5	10
scenario 2	0.05	0	1	1	6	1	10	5	1
scenario 3	0.05	0	10	15	1	6	1	1	1

Table 4.1.: Customer Data

Solutions for the different stochastic models are depicted in figures 4.2, 4.3, 4.4, 4.5, and 4.6. For the Mean-Risk Models three different values for the constant ρ have been considered, η was set to the optimal value of the expected value model, i.e. $\eta = \mathbb{E}[f(x, \omega)]$.

For the EP model the big M was set to 380. One obtains the route 0-6-7-1-4-2-3-5-0 with optimal value 293.

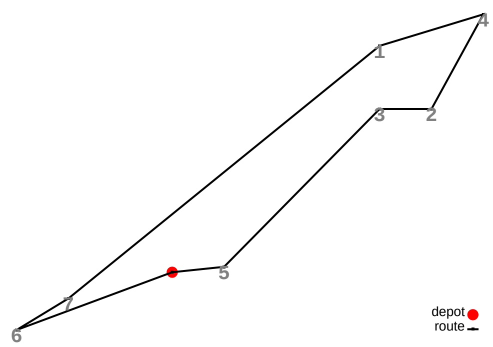

Figure 4.2.: Solution for Expected Value Model

Now, for all Mean-Risk Models η is set to 293. Concerning different values for ρ, optimal routes and its corresponding objective value concerning different values for ρ are depicted in table 4.2.

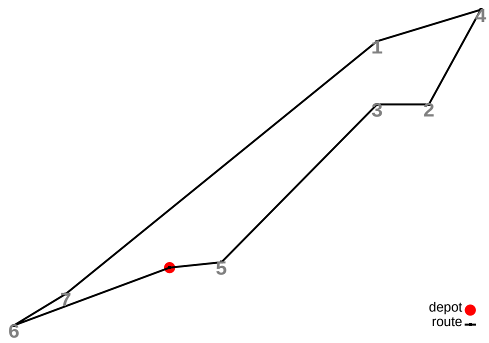

Figure 4.3.: Solution for Risk Model: Excess Probability; $\rho = 1, 10$

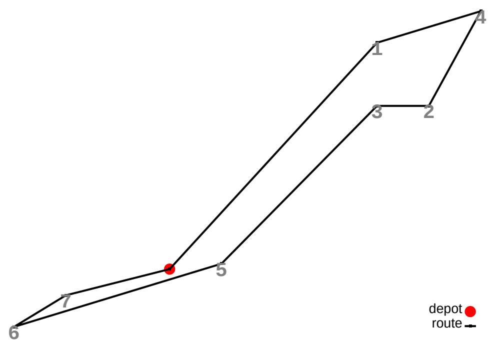

Figure 4.4.: Solution for Risk Model: Excess Probability; $\rho = 100$

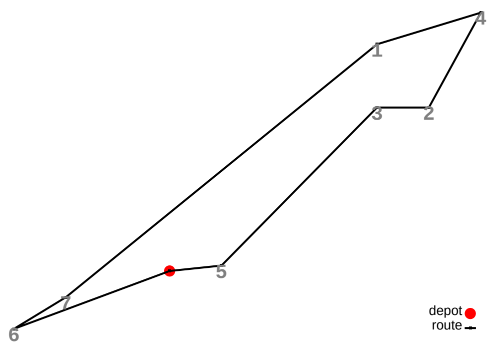

Figure 4.5.: Solution for Risk Model: Expected Excess; $\rho = 1$

	EP		EE	
ρ	route	obj	route	obj
1	0-6-7-1-4-2-3-5-0	293	0-6-7-1-4-2-3-5-0	303
10	0-6-7-1-4-2-3-5-0	294	0-1-4-3-2-5-6-7-0	357
100	0-7-6-5-3-2-4-1-0	299	0-1-4-3-2-5-6-7-0	882

Table 4.2.: Solutions for Mean-Risk Models

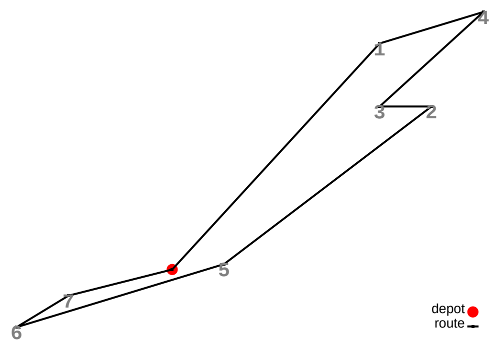

Figure 4.6.: Solution for Risk Model: Expected Excess; $\rho = 10, 100$

4.3. Solution Framework: Integer L-Shaped Algorithm

To solve the problem described in section 4.2 by means of the Integer L-Shaped method the *Current Problem* 4.3.1 is defined by relaxing the integrality and subtour elimination constraints. Further, the $\mathcal{Q}(x)$ term of the objective function is replaced by Θ which is set to be greater or equal L, where L is set to 0, for a start.

Model 4.3.1. *Current Problem*

$$\min \sum_{i<j} c_{ij} x_{ij} + \Theta \tag{4.32}$$

$$s.t. \sum_{j=1}^{n} x_{0j} = 2 \tag{4.33}$$

$$\sum_{i<k} x_{ik} + \sum_{j>k} x_{kj} = 2,\ k = 1, ..., n \tag{4.34}$$

$$\Theta \geq L \tag{4.35}$$

$$0 \leq x_{ij} \leq 1,\ 0 \leq i < j \leq n \tag{4.36}$$

The algorithm starts with solving the above problem 4.3.1. During the solution procedure first feasibility cuts according to subtour elimination constraints are added. If no more violated subtour elimination constraints could be found, for fractional solutions a lower bound on the recourse cost is determined and an additional cut is added using LBFs. To compute a lower bound on the cost of recourse for fractional solutions the concept of partial routes is applied. Next, one checks for violated integrality constraints and branches whenever they are found. If the solution is integer the expected recourse cost is determined and an optimality cut of the form (3.41) is added.

4.3.1. Partial Routes

The concept of partial routes was originally introduced by Hjorring and Holt in [34]. They define a partial route as a sequence of customers interconnected by integer arcs as well as an integer arc connected to the depot, followed by an unsequenced set of customers, and finished with a chain of customers again interconnected with integer arcs and also linked with an integer arc to the depot.

Definition 4.3.2. *Given a set $N = \{1, ..., n\}$ of customers. A Partial Route (PR) is a sequence*

$$(S, U, T),$$

where S, T, and U are disjunctive sets with $S \cup T \cup U = N$ and arcs connecting customers in $S = (0, i_1, ..., i_s)$ and $T = (i_t, ..., i_n, 0)$ are integer, whilst customers in $U = \{i_{s+1}, .., i_{t-1}\}$ are unsequenced, i.e. arcs connecting S and U, and U and T are fractional, respectively.

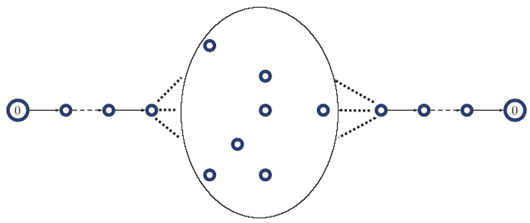

Figure 4.7.: Partial Route

In [36] this concept was generalized by Jabali, Rei, Gendreau, and Laporte. They wanted to take advantage of structural parts in the unsequenced set U. To exploit possible sequenced parts in U, they define a general partial route as follows.

Definition 4.3.3. *Given a set $N = \{1, ..., n\}$ of customers. A General Partial Route (GPR) is a sequence*

$$(0, S_1, U_1, S_2, U_2, ..., S_{k-1}, U_{k-1}, S_k, 0),$$

where all S_i and U_j are disjunctive with $\bigcup_{i=1}^{k} S_i \bigcup_{j=1}^{k-1} U_j = N$ and all customers in $S_i := (i_1, ..., i_s)$ are interconnected by integer arcs, whilst customers in $U_i := \{i_1, .., i_t\}$ are unsequenced and connected to sets S_i by fractional arcs.

Note that S_1 and S_k and also S and T for the partial routes might be empty. i.e. there exist no integer arcs from the depot.

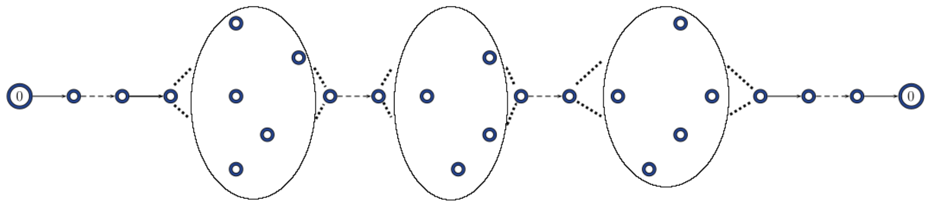

Figure 4.8.: General Partial Route

4.3.2. Lower Bounding Functionals

To improve the efficiency of the Integer L-Shaped method lower bounding functionals are commonly used to tighten the linear relaxation of the current subproblem. Lower bounding functionals are employed to provide information not only of integer solutions, but also of fractional solutions.

The construction of the lower bounding functional $W_h(x)$ for each partial route h is adapted from Jabali, Rei, Gendreau and Laporte [36], where vehicle routing with stochastic demands is considered. They define the functional $W_h(x)$ for the case of multiple vehicles. Further, they tested three different aggregation strategies including the concept of partial routes originally proposed by Hjorring and Holt [34] and their defined general partial routes. In the present thesis their approach is followed for the single vehicle case and for the two mentioned aggregation strategies based on partial routes and general partial routes. According to their formula, $W_h(x)$ then reads:

$$
\begin{aligned}
W_h(x) = & \sum_{(0,j)\in S} x_{0j} + \sum_{(i,j)\in S, i\neq 0} 3x_{ij} + \sum_{j\in U} x_{0j} + \sum_{i\in U} x_{i0} \\
& + \sum_{i,j\in U, i,j\neq 0} 3x_{ij} + \sum_{(i,j)\in T, j\neq 0} 3x_{ij} + \sum_{(i,0)\in T} x_{i0} \\
& -(3|A| + |D|) + 1
\end{aligned} \tag{4.37}
$$

where $|A|$ is the number of arcs not connected to the depot and $|D|$ is the number of arcs connected to the depot. Constructing an LBF out of a partial route h given a solution x, only those variables x_{ij} are taken into account which corresponding value given by the solution x is strictly positive. Since each edge is represented by one variable, the term edge and variable are used synonymously. The coefficients of all edges containing the depot are equal to 1 while all other edges have a coefficient of 3. The number of counted arcs is subtracted by itself $(3|A| + |D|)$ and 1 is added, such that $W_h(x)$ equals 1. For a detailed description of all parts of formula (4.37) see [36].

Given a solution x satisfying (4.33), (4.34), (4.36) and, containing a PR h, a valid inequality for the single vehicle case is given by:

$$\Theta \geq L + (\Theta_h - L) W_h(x), \tag{4.38}$$

where Θ_h is a lower bound on the cost of recourse for PR h. In the next section it is described how to compute Θ_h using different aggregation strategies. Inequality (4.38) reduces to $\Theta \geq \Theta_h$, if $W_h(x') = 1$. This is the case, whenever PR h is compatible with a PR h', where x' satisfies (4.33), (4.34), (4.36), and contains h'. Route compatibility is attained if all customer vertices are ordered in h' as in h. Otherwise $W_h(x) \leq 0$ and the cost of recourse is bounded below by the general lower bound L.

4.3.3. Computation of Lower Bounds on $\mathcal{Q}(x)$

This section provides the computation of a lower bound on the cost of recourse using the concept of PR introduced by Hjorring and Holt [34]. In what follows their definition of PRs is used. Given a PR $h := (0, i_1, ..., i_s, ..., i_e, .., i_n, 0)$, where the customers $S := (i_1, ..., i_s)$ and $T := (i_e, .., i_n)$ are specified, whilst customers $U := \{i_{s+1}, .., i_{e-1}\}$ are unsequenced. A lower bound on the cost of recourse is determined as follows. First, for both orientations $\delta = 1, 2$ and each scenario $\omega = 1, .., S$ the set of fail-proved nodes $F^\delta(\omega)$ is identified. For sequenced customers those nodes can be identified exactly. For the unstructured vertex set U, only a maximum of one fail-proved node is taken into account, which is obviously a lower bound. Let $resCap := D - \sum_{i=1}^{n} d_i$ be the residual capacity, which is left in the vehicle, before starting a determined tour and i_0 referring to the index of the depot. Further, let (S, U, T) be orientation 1, then the fail-proved nodes $F^1(\omega)$ are identified as follows:

Subroutine 4.3.4. *Identifying Fail-Proved Nodes*

$S = (i_1, ..., i_s)$ *:*

$res_\omega(i_0) = resCap$

for $j = 1, ..., s$

$res_\omega(i_j) = res_\omega(i_{j-1}) + d_{i_j} - p_{i_j}(\omega)$

- *if* $res_\omega(i_j) < 0$

set $res_\omega(i_j) = 0$ *FAILURE at customer* i_j*, store customer* i_j *in* $F^1(\omega)$

- *if(*$j = s$*)*

set $res_\omega(S) = res_\omega(i_s)$

END for

$U = \{i_{s+1}, .., i_{e-1}\}$ *:*

if $res_\omega(S) + \sum_{l=1}^{k} d_{i_l} - \sum_{l=1}^{k} p_{i_l}(\omega) < 0$ *then there must be a FAILURE in the unstructured vertex set;*

all customers having $d_{i_l} - p_{i_l}(\omega) < 0, l = 1, ...k$ *form a cluster* $cluster_\omega(U)$*, that is:*

$$cluster_\omega(U) = \{i_j | d_{i_l} - p_{i_l}(\omega) < 0, l = 1, ...k\} \tag{4.39}$$

and set $res_\omega(U) = 0$

else *there exists a sequence of customers such that all pickups can be made without a FAILURE*

set $res_\omega(U) = res_\omega(S_{i-1}) + \sum_{l=1}^{k} d_{i_l} - \sum_{l=1}^{k} p_{i_l}(\omega)$

END if

$T = (i_e, ..., i_n)$ *:*

$res_\omega(i_{e-1}) = res_\omega(U)$

for $j = e, ..., n$

$res_\omega(i_j) = res_\omega(i_{j-1}) + d_{i_j} - p_{i_j}(\omega)$

- *if* $res_\omega(i_j) < 0$

set $res_\omega(i_j) = 0$ *FAILURE at customer* i_j*, store customer* i_j *in* $F^1(\omega)$

END for

Let $T^{-1} := (i_n, ..., i_e)$ and $S^{-1} := (i_s, ..., i_1)$, then the fail-proved nodes $F^2(\omega)$ for orientation 2 are identified with subroutine 4.3.4 according to the sequence (T^{-1}, U, S^{-1}).

Computing the lower bound Θ_h is done in two different ways. Either artificial nodes are imposed for clusters $cluster_\omega(U)$ and a TSP is solved on a modified graph or a GTSP is solved for each scenario. The procedures are described in detail below.

An artificial node, indexed $n+1$, for $cluster_\omega(U)$ is created and added to $F^1(\omega), \omega = 1, ..., S$. The TSP which is solved for each scenario and orientation is defined on a graph $G_\omega = (F^1(\omega), E_\omega)$, with edge set $E_\omega = \{(i,j) : i,j \in F^1(\omega), i < j\}$. Further, the distances between the nodes are defined as:

$$c_{ij}(\omega) = \begin{cases} c_{ij} & \text{if } j \neq n+1, \\ min\_dist(i, cluster_\omega(U)) & \text{if } j = n+1, \end{cases}$$

where $min\_dist(i, cluster_\omega(U))$ is the minimum distance from the node set $cluster_\omega(U)$ to node i, that is for fixed node i:

$$min\_dist(i, cluster_\omega(U)) = \min\{c_{ij} : j \in cluster_\omega(U)\}.$$

A lower bound on the cost of recourse for a given PR h is then given by:

$$\Theta_h = \min\{\sum_{\omega=1}^{S} \pi_\omega T^1(\omega), \sum_{\omega=1}^{S} \pi_\omega T^2(\omega)\}, \tag{4.40}$$

where $T^\delta(\omega)$ denotes the optimal value of the TSP defined on the graph G_ω and orientation $\delta = 1, 2$.

The clusters for solving the GTSP are defined as follows:

- for each fail-proved node in $F^{\delta}(\omega)$ its own cluster $C_s(\omega)$, $s = 1, ..., |F^{\delta}(\omega)|$ is created, i.e. each cluster $C_s(\omega)$ consists only of one node
- $cluster_{\omega}(U)$ is added to the cluster set of the GTSP

We end up with $|F^{\delta}(\omega)|+1$ different clusters. The distances between customers are taken from the initial problem. A lower bound on the cost of recourse for a given PR h is then given by:

$$\Theta_h = \min\{\sum_{\omega=1}^{S} \pi_{\omega} G^1(\omega), \sum_{\omega=1}^{S} \pi_{\omega} G^2(\omega)\}, \tag{4.41}$$

where $G^{\delta}(\omega)$ denotes the optimal value of the GTSP defined for the clusters $C_s(\omega)$ and $cluster_{\omega}(U)$ and orientation $\delta = 1, 2$.

Concerning a GPR, one tries to exploit the set $cluster_{\omega}(U)$ by identifying further sequences connected by integer arcs, in the sense depicted in figure 4.8. Subroutine 4.3.4 is modified, such that it provides for clusters $cluster_{\omega}(U_i)$ for each unstructured set U_i. Artificial nodes for the TSP are created for each cluster $cluster_{\omega}(U_i)$. For the GTSP approach all clusters $cluster_{\omega}(U_i)$ are added to the cluster set.

4.3.4. Global Lower Bounds

The computation of a global lower bound for the cost of recourse, is mainly dependent on the sequencing of the customers. Prior to the solution process, lower bounds on the expected cost of recourse are determined, when fixing one customer i, $i = 1, ..., n$ that is visited first or last. More specifically, one generates n additional cuts constructed from PRs that consist of one chain containing only node i and one unstructured part consisting of nodes $N/\{0, i\}$.
First/Last Fixing Inequalities:

$$\Theta \geq \sum_{\omega \in \Omega} \pi_{\omega} R_{\omega}(i) x_{0i}, \quad \forall i \in N \backslash \{0\} \tag{4.42}$$

where $R_\omega(i) = \min\{R^1_\omega(i), R^2_\omega(i)\}$ and $R^\delta_\omega(i)$ are the compensation costs of scenario ω visiting customer i is visited first ($\delta = 1$) or last ($\delta = 2$). We entitle cuts (4.42) *F-Cuts*.

A global lower bound L is obtained by taking the minimum over $i \in N \backslash \{0\}$ of $R_\omega(i)$:

$$L := \min_{i \in N/\{0\}} \sum_{\omega \in \Omega} \pi_\omega R_\omega(i) \tag{4.43}$$

5. Computational Experiments

5.1. Implementation Details

The Integer L-Shaped algorithm was coded on a C++ environment with CPLEX 12.6. Problems were solved on a computer with 126 GB of RAM using one 2.6-GHz Dual-Core AMD Opteron(tm) processor, and operating under Linux Ubuntu 10.04.4 LTS. The separation procedure of the subtour elimination constraints was performed using the CVRPSEP package of Lysgaard, Letchford and Eglese [48].

5.2. Computational Results

To assess the performance of the algorithm several sets of randomly generated scenarios were formed. The instances are based on the Solomon benchmark problems [68]. For instances with 10, 15, 20, and 50 customers demands as well as the geographical data are taken from the Solomon clustered (c-type), random (r-type), and mixed (rc-type) instances.

For the 3 graph types we created 3 different groups of pickup data, where either all or 50% of the customers have stochastic pickup demands. First, the delivery demands were shuffled to generate the pickup amounts. Second, the pickup demands were generated by multiplying the delivery demands with a uniformly distributed deviation factor of 20% and 70%. Third, the delivery demands have been perturbed by a normal distributed random variable with expected value $\mu = 1$ truncated at 0 with a standard deviation of $\sigma = 0.2$ and $\sigma = 0.7$. Random pickup amounts $p_i(\omega)$ for each customer $i \in N/\{0\}$ and scenario $\omega \in \Omega$ are calculated as follows:

$$p_i(\omega) = \mu(\omega) \cdot d_i, \tag{5.1}$$

where $\mu(\omega)$ is a uniformly or normally distributed random variable and d_i are the delivery demands taken from the Solomon benchmark instances. The case where only 50% of the customers have stochastic pickup demands is considered for the instances with a deviation factor or coefficient of variation of 70%, only.

The probabilities for each scenario were created randomly with the C++ function srand48() using time as seed. For each customer size, instances with 10, 30, 50, and 100 scenarios were built, such that we end up with 336 instances. The computation time limit was set to 5 hours.

5.2.1. Risk Neutral Model

For the risk neutral case, we tested 4 different algorithmic features, named C_PR, C_PR+GTSP, C_GPR, and C_GPR+GTSP. In all implementations F-Cuts presented in section 4.3.4 are used. For C_PR and C_GPR the construction of an artificial node is implemented and PRs or GPRs are used, respectively. In C_PR+GTSP and C_GPR+GTSP the computation of a GTSP is deployed and PRs or GPRs are used, respectively.

The experimental results for the risk neutral model are summarized in tables 5.1 - 5.7. To avoid overloading the text we display the average over the number of customers and scenarios and refer to the appendix A for the detailed and complete tables. Although the averaged results represent the typical situation, there are outliers as well. Column one is referred to the code-type and column two displays the graph-type. In column three the average gap at the root node is depicted. Column four denotes the average CPU time over the solved as well as over the unsolved instances[1]. Columns five to eight display

[1]An instance is considered "unsolved" if no solution has been found after a CPU time of 18000 seconds.

the average number of subtour elimination constraints, the average number of cuts gained from LBFs, the average number of GPRs, and the average number of optimality cuts, respectively. The gaps are reported in the ninth column, which concern the average of unsolved instances and solved instances having a 0.00% gap. The last column shows the number of solved instances.

For the benefit of the reader, the immediate text to come is structured into the paragraphs: Implementations, Generated Cuts, Graph Types, and Distributions.

Implementations Regarding the 4 different implementations the C_PR+GTSP performs the best. Using C_PR+GTSP solves the most instances, produces the smallest gaps, and also has the least CPU time, except for the shuffled data. In the test, we observed that extending PRs into GPRs did not lead to the advantages, one might expect due to enriched structure. Therefore, it was not worth it to spend additional computation time to generate GPRs from PRs. For the 10- and 15- customers instances the number of GPRs can be neglected. Only starting from customer size 20 a few GPRs can be found (see Appendix A tables A.1-A.21). In comparison to the number of LBFs, the number of GPRs are insignificant, which explains the very small influence regarding the gaps. Combining GPRs and the GTSP even produces inferior results in some problem settings.

Generated Cuts Pertaining to the generated cuts, the average number of cuts gained form LBFs is always lower than the average number of optimality cuts. Surprisingly, the C_PR+GTSP implementation has created on average the largest number of cuts gained from LBFs, except for the uniformly distributed instances with 100% stochastic customers and 70% deviation. Due to the fact, that creating an artificial node instead of solving a GTSP is less time consuming, one might have anticipated, that C_PR would create the largest number. Moreover, as expected, implementations using GPRs produce less cuts than the ones with PRs in all settings.

Table 5.1.: Computational results on instances with shuffled delivery demand

code	graph-type	root	CPU (sec)	Sub	LBF	GPR	OptCut	gap	solved
C_PR	clustered	7.02%	5112.74	178	2436	-	2319	3.37%	12/16
	random	5.49%	4532.31	179	1068	-	1212	0.74%	12/16
	mixed	12.50%	9629.86	222	6049	-	6774	6.03%	8/16
	overall-avrg	8.34%	6424.97	193	3184	-	3435	3.38%	32/48
C_PR+GTSP	clustered	7.02%	5129.87	181	2482	-	2335	3.30%	12/16
	random	5.49%	4525.18	190	1128	-	1242	0.71%	12/16
	mixed	12.50%	9569.18	225	6248	-	7097	5.76%	8/16
	overall-avrg	8.34%	6408.08	199	3286	-	3558	3.26%	32/48
C_GPR	clustered	7.02%	4997.41	176	2150	14	2413	3.34%	12/16
	random	5.49%	4537.12	188	1012	4	1261	0.73%	12/16
	mixed	12.50%	9622.09	221	4345	21	5337	5.93%	8/16
	overall-avrg	8.34%	6385.54	575	2502	13	3004	3.33%	32/48
C_GPR+GTSP	clustered	7.02%	5119.95	178	2098	12	2418	3.37%	12/16
	random	5.49%	4541.53	197	1261	7	1561	0.65%	12/16
	mixed	12.50%	9621.39	228	4860	23	6035	6.05%	8/16
	overall-avrg	8.34%	6427.62	201	2740	14	3338	3.36%	32/48

Table 5.2.: Computational results on instances with 100% random customers, normally distributed pickup demand, and $\sigma = 0.2$

code	graph-type	root	CPU (sec)	Sub	LBF	GPR	OptCut	gap	solved
C_PR	clustered	3.87%	9045.28	169	3531	-	3074	2.05%	4/8
	random	2.94%	9000.12	304	1852	-	2247	1.05%	4/8
	mixed	4.07%	13526.70	178	6119	-	6017	2.26%	2/8
	overall-avrg	3.63%	10524.04	217	3834	-	3779	1.79%	10/24
C_PR+GTSP	clustered	3.87%	9039.46	157	3871	-	3210	1.98%	4/8
	random	2.94%	6773.92	256	1561	-	1807	0.82%	5/8
	mixed	4.07%	13530.89	184	9358	-	9848	1.75%	2/8
	overall-avrg	3.63%	9781.42	199	4930	-	4955	1.52%	11/24
C_GPR	clustered	3.87%	9054.55	164	3310	17	3062	2.04%	4/8
	random	3.28%	9000.22	318	2148	10	2698	0.99%	4/8
	mixed	3.98%	13555.15	169	5129	18	5264	2.21%	2/8
	overall-avrg	3.71%	10536.64	217	3529	15	3675	1.75%	10/24
C_GPR+GTSP	clustered	3.87%	9066.02	164	3469	23	3268	2.03%	4/8
	random	3.28%	9000.18	317	2142	8	2663	0.99%	4/8
	mixed	3.98%	13539.84	170	5388	18	5488	2.19%	2/8
	overall-avrg	3.71%	10535.35	217	3666	16	3806	1.74%	10/24

Graph Types Comparing computational results for all graph types, implementations and data situations, see tables 5.1 - 5.7, yields the conclusion that our algorithm, with negligible exceptions, performed best at the random graph instances. That means, reached optimality fastest and if not, terminated with smallest gaps. For the clustered and mixed types the performance fluctuates. The clustered instances show better results for the shuffled and the 100% stochastic customer data, whereas the mixed type performs better for the 50% stochastic customer data.

Distributions When comparing the different distributions, the instances with uniformly distributed pickup demands and 20% deviation are the easiest ones. Further, the results create the impression that instances with normally distributed demands are more difficult to solve than instances with uniformly distributed demands. Concerning the tendency by increasing the deviation factor the complexity and intricacy grow.

Table 5.3.: Computational results on instances with 50% random customers, normally distributed pickup demand, and $\sigma = 0.7$

code	graph-type	root	CPU (sec)	Sub	LBF	GPR	OptCut	gap	solved
C_PR	clustered	11.66%	10676.80	356	5274	-	6819	7.11%	8/16
	random	6.30%	7265.42	360	2886	-	2905	2.47%	10/16
	mixed	10.52%	12429.95	241	6759	-	7635	5.40%	5/16
	overall-avrg	9.49%	10124.06	319	4973	-	5786	4.99%	23/48
C_PR+GTSP	clustered	11.66%	10214.85	322	4973	-	6217	5.68%	8/16
	random	6.30%	7302.67	364	3060	-	2949	2.28%	10/16
	mixed	10.52%	12381.63	224	7139	-	7259	4.25%	5/16
	overall-avrg	9.49%	9966.38	303	5057	-	5475	4.07%	23/48
C_GPR	clustered	11.29%	10727.02	345	4243	2	5600	7.16%	7/16
	random	6.07%	7051.33	355	2346	5	2504	2.45%	10/16
	mixed	9.88%	12392.09	225	5125	3	5757	5.22%	5/16
	overall-avrg	9.08%	10056.81	308	3905	3	4620	4.94%	22/48
C_GPR+GTSP	clustered	11.29%	11344.74	328	4001	2	5074	7.26%	6/16
	random	6.07%	7073.23	353	2206	3	2449	2.47%	10/16
	mixed	9.88%	12391.81	222	4843	5	5284	5.27%	5/16
	overall-avrg	9.08%	10269.93	301	3683	3	4269	5.00%	21/48

Out of the 336 instances 209 could be solved to optimality. Our algorithm is able to solve almost all 10 customer instances, except for the clustered graph type concerning the 100% stochastic customer data and normally distribution with $\sigma = 0.7$.

Table 5.4.: Computational results on instances with 100% random customers, normally distributed pickup demand, and $\sigma = 0.7$

code	graph-type	root	CPU (sec)	Sub	LBF	GPR	OptCut	gap	solved
C_PR	clustered	13.84%	12623.12	396	5536	-	5654	7.84%	5/16
	random	12.38%	9565.61	519	3443	-	4418	5.36%	8/16
	mixed	15.78%	13516.76	310	6095	-	6950	10.39%	4/16
	overall-avrg	14.00%	11901.83	408	5025	-	5674	7.86%	17/48
C_PR+GTSP	clustered	13.84%	11598.31	361	5474	-	7652	6.60%	6/16
	random	12.38%	9399.21	493	3546	-	4464	4.70%	8/16
	mixed	15.78%	12602.65	262	6459	-	7013	7.56%	5/16
	overall-avrg	14.00%	11200.06	372	5160	-	6376	6.29%	19/48
C_GPR	clustered	13.55%	11636.36	373	5084	4	6987	7.35%	6/16
	random	11.92%	9323.25	519	3785	3	5079	5.07%	8/16
	mixed	15.10%	13503.19	309	6266	6	7089	9.32%	4/16
	overall-avrg	13.52%	11487.60	400	5045	4	6385	7.25%	18/48
C_GPR+GTSP	clustered	13.55%	11625.96	346	4524	5	6137	7.46%	6/16
	random	11.92%	9321.15	493	3370	2	4402	5.12%	8/16
	mixed	15.10%	13503.42	298	4914	2	5518	9.48%	4/16
	overall-avrg	13.52%	11483.51	379	4269	3	5352	7.35%	18/48

Table 5.5.: Computational results on instances with 100% random customers, uniformly distributed pickup demand, and 20% deviation

code	graph-type	root	CPU (sec)	Sub	LBF	GPR	OptCut	gap	solved
C_PR	clustered	0.28%	1110.91	34	945	-	458	0.00%	8/8
	random	0.01%	0.42	11	0	-	3	0.00%	8/8
	mixed	1.30%	527.47	44	1561	-	1073	0.00%	8/8
	overall-avrg	0.53%	546.27	30	836	-	511	0.00%	24/24
C_PR+GTSP	clustered	0.28%	539.41	34	943	-	458	0.00%	8/8
	random	0.01%	0.32	11	0	-	1	0.00%	8/8
	mixed	1.30%	566.77	43	1606	-	1072	0.00%	8/8
	overall-avrg	0.53%	368.83	30	850	-	510	0.00%	24/24
C_GPR	clustered	0.28%	886.38	34	807	2	459	0.00%	8/8
	random	0.01%	0.38	11	0	0	1	0.00%	8/8
	mixed	1.29%	1024.66	40	1269	0	1003	0.00%	8/8
	overall-avrg	0.53%	637.14	28	692	1	488	0.00%	24/24
C_GPR+GTSP	clustered	0.28%	864.44	34	798	2	458	0.00%	8/8
	random	0.01%	0.33	11	0	0	1	0.00%	8/8
	mixed	1.29%	954.07	41	1295	0	1003	0.00%	8/8
	overall-avrg	0.53%	606.28	29	697	1	487	0.00%	24/24

Table 5.6.: Computational results on instances with 50% random customers, uniformly distributed pickup demand, and 70% deviation

code	graph-type	root	CPU (sec)	Sub	LBF	GPR	OptCut	gap	solved
C_PR	clustered	7.35%	7459.27	213	3387	-	3222	4.09%	10/16
	random	3.92%	4807.68	271	1275	-	1462	2.23%	12/16
	mixed	6.62%	10448.56	193	6043	-	5693	3.96%	8/16
	overall-avrg	5.96%	7571.84	226	3568	-	3459	3.43%	30/48
C_PR+GTSP	clustered	7.35%	7154.61	220	3931	-	3738	3.59%	10/16
	random	3.92%	4706.09	260	1540	-	1716	1.86%	12/16
	mixed	6.62%	10124.42	175	6433	-	6255	2.91%	8/16
	overall-avrg	5.96%	7328.37	218	3968	-	3903	2.79%	30/48
C_GPR	clustered	7.09%	7883.33	207	2872	2	3035	4.15%	9/16
	random	3.94%	4730.48	283	1289	3	1523	2.19%	12/16
	mixed	6.45%	10446.85	185	4380	5	4266	3.95%	7/16
	overall-avrg	5.82%	7686.88	225	2847	3	2112	3.43%	28/48
C_GPR+GTSP	clustered	7.09%	7884.74	210	2900	2	3086	4.21%	9/16
	random	3.94%	4728.51	271	1272	2	1510	2.19%	12/16
	mixed	6.45%	9878.27	184	4860	5	4682	3.89%	8/16
	overall-avrg	5.68%	7497.17	222	3011	3	3093	3.43%	29/48

Table 5.7.: Computational results on instances with 100% random customers, uniformly distributed pickup demand, and 70% deviation

code	graph-type	root	CPU (sec)	Sub	LBF	GPR	OptCut	gap	solved
C_PR	clustered	11.03%	9805.65	337	5048	-	6233	5.77%	8/16
	random	6.41%	6716.80	326	2439	-	2866	1.97%	11/16
	mixed	11.55%	13500.16	253	7508	-	8105	6.90%	4/16
	overall-avrg	9.66%	10007.54	305	4998	-	5735	4.88%	23/48
C_PR+GTSP	clustered	11.03%	9651.59	306	5095	-	5868	4.68%	8/16
	random	6.41%	6608.45	323	2626	-	2741	1.65%	11/16
	mixed	11.55%	11615.08	214	6849	-	6899	4.52%	6/16
	overall-avrg	9.66%	9291.71	281	4857	-	5169	3.62%	25/48
C_GPR	clustered	10.93%	9697.77	310	4189	5	5179	5.80%	8/16
	random	6.17%	6862.17	330	2031	3	2742	2.01%	10/16
	mixed	10.96%	13500.24	238	5598	4	5805	6.86%	4/16
	overall-avrg	9.35%	10020.29	293	2989	4	4575	4.89%	22/48
C_GPR+GTSP	clustered	10.93%	9743.69	311	3942	5	5148	5.78%	8/16
	random	6.17%	6899.98	305	1692	2	2296	2.01%	10/16
	mixed	10.96%	13500.39	239	6204	5	6754	6.70%	4/16
	overall-avrg	9.35%	10048.02	285	3946	4	4733	4.83%	22/48

Due to the fact, that lower bounds are only determined by solving the linear relaxation of the master problem, these bounds are very weak. Optimal or present best solutions are found at the very beginning of the solution process, consequently most of the solution time is spent to tighten the lower bound. By finding a way to produce better lower bounds, the algorithm could be speed up enormously.

5.2.2. Averaged Value versus Averaged Data (Model)

In this section best available solutions gained from the Expected Value Model (EV-Model) are compared to solutions determined by taking the Expected Value (EV) of the random data. The results are summarized in tables 5.8 and 5.9. Column one is referred to the graph-type and column two displays the number of customers. In column three to five the average relative additional costs when inserting the solution gained from the model with random data replaced by their EV are reported, we refer to this model as EV of Expected Solution (EEV). That is, let $\bar{x} = argmin\{f(x, \mathbb{E}[\omega])\}$, further let $c_{EEV} = \mathbb{E}[f(\bar{x}, \omega)]$ and $c_{EV} = \min\{\mathbb{E}[f(x, \omega)]\}$, then the relative additional costs are determined as follows:

$$(c_{EEV} - c_{EV}) * 100 / c_{EEV}.$$

Only the instances with 100% stochastic pickup data and 100 scenarios are taken into account.

Comparing the 3 different instance classes, one gains the highest savings when using the EV-Model for the shuffled instances (see table 5.8). Although the average of instances with normally distributed pickup demands produce higher savings compared to the uniformly distributed instances, for the clustered and mixed graph types it is vice versa.

Table 5.9 shows the average savings with respect to the number of customers. When increasing the number of customers the average savings over the graph types increase for the shuffled instances, but concerning the uniformly and normally distributed pickup data the average savings decrease for the 50-customer instances.

Table 5.8.: Comparison of savings for instances with up to 50 customers

graph-type	n	shuffled	uniformly 70%	normally 0.7
clustered	10	4.9335%	1.3216%	0.1806%
	15	4.8820%	0.1278%	0.0000%
	20	6.2440%	4.1569%	3.5948%
	50	4.3700%	1.0574%	1.5713%
avrg		3.0607%	1.6659%	1.3367%
random	10	4.1481%	0.0000%	0.0000%
	15	0.4309%	0.0000%	0.0000%
	20	5.0140%	0.0000%	0.7963%
	50	9.5358%	1.0456%	0.0000%
avrg		4.8762%	0.2614%	0.1991%
mixed	10	0.3635%	0.0000%	0.4524%
	15	7.2658%	1.0861%	1.7682%
	20	9.8969%	4.0045%	5.6803%
	50	5.1982%	1.5826%	1.8742%
avrg		5.6811%	1.6683%	2.4438%
overall-avrg		4.5393%	1.1985%	1.3265%

Table 5.9.: Average savings concerning the number of customers

n	shuffled	uniformly 70%	normally 0.7
10	3.1484%	0.4405%	0.2110%
15	3.4272%	0.4046%	0.5894%
20	5.1429%	2.7205%	3.3571%
50	6.4388%	1.2285%	1.1485%

5.2.3. Risk Averse Models

This section deals with the computational results for the risk averse models, more specifically for the Excess Probability (EP) and Expected Excess (EE) risk measures introduced in section 2.2 and adapted to the VRPSDSP in section 4.2.2.

We tested the risk averse models for the instances with 100 scenarios and 100% stochastic customer data, only. Most of the calculated instances could not be solved to optimality and the resulting gaps are very poor, especially for the EE-Model (for details see appendix A tables A.22 - A.25). When reporting computational results, we find it convenient to refer to best available solutions, only. This concerns solutions for which we are able to prove optimality and solutions, where we could not, but at least were able to provide lower bounds for the optimal value.

Table 5.10.: Solutions determined by EP-Model compared to solutions of EV-Model, EEV, and EE-Model with $\rho = 50$.

			route-costs				recourse-costs			
data-type	n	graph-type	EP	EV	EEV	EE	EP	EV	EEV	EE
shuffle	10	clustered	58.49	±0	-3.20	+1.69	2.86	±0	+15.29	+47.14
		random	182.06	±0	-9.02	±0	7.42	±0	+32.65	±0
		mixed	141.37	±0	-3.59	±0	0.00	±0	+6.89	±0
	15	clustered	103.19	±0	-0.57	+0.65	0.94	±0	+7.33	+49.06
		random	237.27	-6.99	-8.11	+3.53	7.15	+9.06	+12.36	+42.85
		mixed	159.08	±0	-4.54	±0	9.20	±0	+28.44	±0
	20	clustered	127.85	-3.84	-3.84	+0.23	8.86	+4.01	+4.01	+44.05
		random	268.05	±0	-5.71	±0	2.77	±0	+37.81	±0
		mixed	225.97	±0	-5.16	±0	34.34	±0	+49.07	±0
	50	clustered	255.37	-13.27	-14.94	-12.81	41.27	+27.76	+45.83	+27.88
		random	472.49	±0	-7.18	-2.19	28.05	±0	+79.87	+9.62
		mixed	411.81	±0	-37.97	+22.32	60.95	±0	+70.97	-15.98
norm0.7	10	clustered	55.42	±0	-0.13	-0.13	50.84	±0	+0.65	+0.65
		random	173.04	±0	±0	±0	58.55	±0	±0	±0
		mixed	138.06	±0	-0.28	±0	35.21	±0	+37.60	±0
	15	clustered	102.62	±0	±0	+4.39	19.12	±0	±0	-2.27
		random	230.28	±0	±0	+7.95	61.21	±0	±0	-4.75
		mixed	157.75	+0.24	-4.12	+0.40	67.92	-0.02	+7.11	+1.41
	20	clustered	127.86	-3.84	-4.21	+2.77	68.48	+5.59	+4.12	+3.60
		random	262.34	+1.67	-1.12	+1.67	91.63	-1.09	+2.63	-1.09
		mixed	221.93	+2.63	+6.27	+2.69	121.63	+8.00	-0.45	+4.34
	50	clustered	244.63	+0.25	-4.60	+0.01	92.43	+15.41	+24.56	+12.50
		random	462.03	±0	±0	+9.09	111.62	±0	±0	+1.18
		mixed	376.73	+2.66	-6.80	+0.01	141.19	+7.00	+10.77	+6.15
uni70%	10	clustered	57.06	±0	-1.77	±0	30.39	±0	+5.95	±0
		random	173.04	±0	±0	+9.02	75.76	±0	±0	-3.53
		mixed	137.78	±0	±0	+4.58	55.71	±0	±0	-0.65
	15	clustered	104.39	±0	-1.77	+4.63	6.71	±0	+1.91	-1.82
		random	229.16	±0	±0	±0	62.26	±0	±0	±0
		mixed	154.54	+2.38	-0.91	+4.19	85.74	+0.04	+2.53	-1.23
	20	clustered	129.60	+0.35	-5.96	±0	26.35	+0.87	+6.84	±0
		random	262.34	±0	±0	+15.72	80.69	±0	±0	-1.78
		mixed	220.05	+6.88	-0.65	±0	75.92	+0.08	+2.21	±0
	50	clustered	241.00	+0.36	-0.98	+1.27	84.54	+4.02	+3.04	+19.21
		random	468.74	±0	-6.71	-2.03	70.36	±0	+16.15	+7.46
		mixed	370.74	+1.11	-0.80	+4.14	119.43	+3.36	+1.89	-2.79

The parameter ρ was set to 50 and 100 and threshold η was chosen to be the solution of the EV-Model. The results for $\rho = 50$ are depicted in table 5.10 and table 5.11 present the results for $\rho = 100$. Solutions determined by the EP-Model (x_{EP}) are compared to solutions gained from the EV-Model (x_{EV}), the EEV (x_{EEV}), and the EE-Model (x_{EE}), respectively. In column four the route costs for x_{EP} ($costs(x_{EP})$) are displayed. Columns five to seven show the difference between ($costs(x_{EP})$) and the route costs of the EV-Model ($costs(x_{EV})$), EEV ($costs((x_{EEV})$), and the EE-Model ($costs((x_{EE})$), respectively[2]. Column eight depicts the recourse costs given by x_{EP} ($rec(x_{EP})$), columns nine to eleven display the difference between $rec(x_{EP})$ and the recourse costs given by x_{EV} ($rec(x_{EV})$), x_{EEV} ($rec(x_{EEV})$), and x_{EE} ($rec(x_{EE})$), respectively.

Table 5.11.: Solutions determined by EP-Model compared to solutions of EV-Model, EEV, and EE-Model with $\rho = 100$.

			route-costs				recourse-costs			
data-type	n	graph-type	EP	EV	EEV	EE	EP	EV	EEV	EE
shuffle	10	clustered	58.49	±0	-3.20	+1.69	4.51	±0	+24.30	+95.49
		random	182.06	±0	-9.02	±0	11.69	±0	+48.27	±0
		mixed	141.37	±0	-3.59	±0	0.00	±0	+9.67	±0
	15	clustered	103.19	±0	-0.57	+0.65	1.48	±0	+11.36	+98.52
		random	237.27	-6.99	-8.11	+3.53	10.84	+13.47	+17.94	+89.16
		mixed	159.08	±0	-4.54	±0	12.86	±0	+39.45	±0
	20	clustered	130.52	-6.5	-6.55	-0.52	6.75	+12.45	+14.56	+5.14
		random	268.05	±0	-5.71	±0	4.09	±0	+54.56	±0
		mixed	229.60	-3.63	-8.79	-3.63	43.60	+4.67	+70.59	+4.67
	50	clustered	257.58	-15.48	-17.15	-4.17	72.05	+29.39	+56.55	+47.01
		random	472.49	±0	-7.18	-2.19	44.88	±0	+97.27	+10.81
		mixed	410.60	-1.65	-36.75	-1.65	85.99	+2.49	+91.31	+2.49
norm0.7	10	clustered	55.42	±0	-0.13	-0.26	78.84	±0	+1.03	+1.03
		random	173.04	±0	±0	±0	83.45	±0	±0	±0
		mixed	138.06	±0	-0.28	±0	48.62	±0	+3.42	±0
	15	clustered	102.62	±0	±0	+4.39	28.60	±0	±0	-2.98
		random	229.16	±0	±0	+6.83	88.25	±0	±0	-4.40
		mixed	157.75	+0.24	-4.12	+0.40	90.50	-0.01	+10.47	+2.48
	20	clustered	130.26	-2.27	-2.27	+0.85	91.72	+6.88	+6.88	+9.94
		random	262.34	+1.67	±0	+1.67	116.23	+0.42	±0	+0.42
		mixed	219.42	+5.14	+8.78	+5.20	154.76	+10.01	-3.39	+7.24
	50	clustered	244.88	+0.01	-4.86	-3.45	116.92	+1.54	+20.45	+16.59
		random	462.03	±0	±0	+5.74	137.08	±0	±0	+2.49
		mixed	375.17	+4.22	-5.24	+5.14	170.41	+6.62	+10.89	+0.81
uni70%	10	clustered	57.06	±0	-1.77	±0	47.08	±0	+9.18	±0
		random	173.04	±0	±0	+9.02	125.76	±0	±0	-3.53
		mixed	137.78	±0	±0	+4.58	105.71	±0	±0	-0.65
	15	clustered	104.39	±0	-1.77	+4.63	6.70	±0	+1.92	-1.82
		random	229.16	±0	±0	+8.60	112.26	±0	±0	-0.41
		mixed	154.54	+2.38	-0.91	±0	135.74	+0.04	+2.53	±0
	20	clustered	129.60	+0.35	-5.96	+4.62	26.35	+0.87	+6.84	+1.04
		random	262.34	±0	±0	+15.72	130.69	±0	±0	-1.78
		mixed	220.05	+6.88	-0.65	±0	75.92	+0.08	+2.21	±0
	50	clustered	241.00	+0.36	-0.98	+0.72	114.74	+5.74	+3.95	+40.07
		random	468.74	±0	-6.71	-2.03	90.65	±0	+21.58	+10.46
		mixed	370.74	+1.11	-0.80	+5.77	169.43	+3.35	+1.88	-3.64

EP versus EV Comparing x_{EP} and x_{EV} 22 of 36 routes are identical. Among the routes that differ, those belonging to shuffled instances, the resulting route costs are lower ($costs(x_{EV}) < costs(x_{EP})$) and the recourse costs are higher ($rec(x_{EV}) > rec(x_{EP})$). This may be due to the fact, that the scenario solutions for the shuffled instances from the EV-Model are subject to fluctuations. The fluctuations could be caused by large deviations of the

[2] The differences are calculated as: $costs(x_{EV}) - costs(x_{EP})$, $costs(x_{EEV}) - costs(x_{EP})$, and $costs(x_{EE}) - costs(x_{EP})$, respectively.

pickup demands concerning different scenarios. For the normally and uniformly distributed instances, route costs as well as recourse costs are higher ($costs(x_{EV}) > costs(x_{EP})$ and $rec(x_{EV}) > rec(x_{EP})$), however there are some exceptions.

EP versus EEV Regarding solutions gained from EEV x_{EEV}, for each data setting, route costs are lower and recourse costs are higher compared to solutions determined by the EP-Model x_{EP}, i.e. $costs(x_{EEV}) < costs(x_{EP})$ and $rec(x_{EEV}) > rec(x_{EP})$, except for one instance. For the mixed 20-customer instance with normally distributed data an exception occurred. Here, just the opposite could be observed: $costs(x_{EEV}) > costs(x_{EP})$ and $rec(x_{EEV}) < rec(x_{EP})$. However, this can be explained by the remaining gap. The instance was not solved to optimality and the best solution found upon termination was taken. Inserting x_{EP} in the EEV, one gains lower costs than for x_{EEV}. Due to the fact, that for the EEV only one scenario with the EV of the random data is considered, it was to be expected that $costs(x_{EEV}) < costs(x_{EP})$ and $rec(x_{EEV}) > rec(x_{EP})$.

EP versus EE Solutions gained from the EE-Model seem to be more pessimistic. That means most route costs and recourse costs are higher compared to solutions calculated by the EP-Model. However, one has to take into account, that most instances still have a quite large gap (see appendix A table A.24).

$\rho = 50$ versus $\rho = 100$ Changing parameter ρ from 50 to 100, does not have a major influence on the solutions, i.e. only a few routes change. However, of course, the recourse costs increase. Comparing solutions determined by the different models, one observes the same correlation for $\rho = 50$ and $\rho = 100$.

Cross-Check Exemplary, we would like to examine the clustered 50-customer instance with shuffled demand in detail. Figure 5.1 shows the quantitative difference in the cost distribution over the ensemble of pickup scenarios for the different risk averse measures.

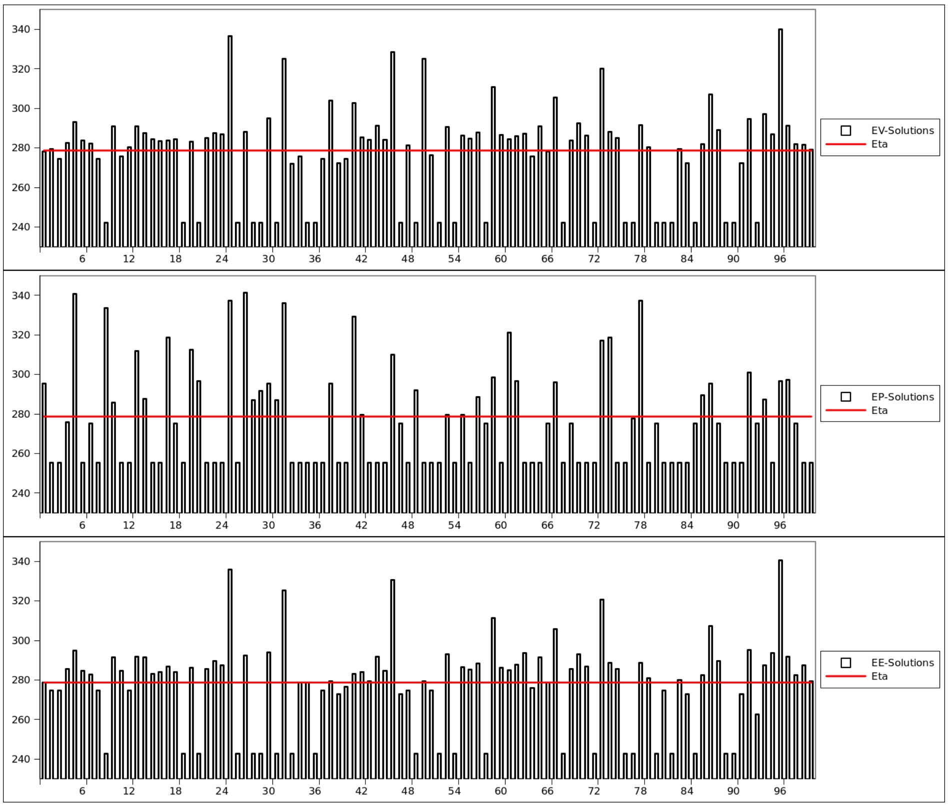

Figure 5.1.: 50-customer instance: Objective values for each of the hundred scenarios are rendered with bar charts for the solutions given by the EV-Model, EP-Model, and EE-Model (from top to bottom) for $\eta = 278.7135$ and $\rho = 50$.

For the EV-Model a weighted sum over all single-scenario objectives is minimized. That means scenarios with high objective values can be compensated by scenarios with low objective values. For the EP-Model it only matters whether the objective value exceeds η or not. Therefore, the arising flexibility given by scenarios that do not exceed η can be used to decrease objective values of scenarios above η, such that the number of scenarios exceeding η can be reduced from 61 to 37. However, scenarios exceeding η are expected to be "lost" anyway and a growth of objective values can be observed. Regarding the EE-Model, also the amount of excess over η is taken into account. Hence, decreasing objective values of scenarios above η gains more attention.

The value of η has a main impact on finding optimal and even feasible solutions. In our calculations η is chosen, such that feasible and optimal solutions can be found easily. Taking evenly small probabilities, like in our present example (see figure 5.2), the selection of the treated scenario exerts little influence. However, choosing strongly asymmetric probabilities, i.e. there are outliers with high probabilities, would result in more fluctuating solutions.

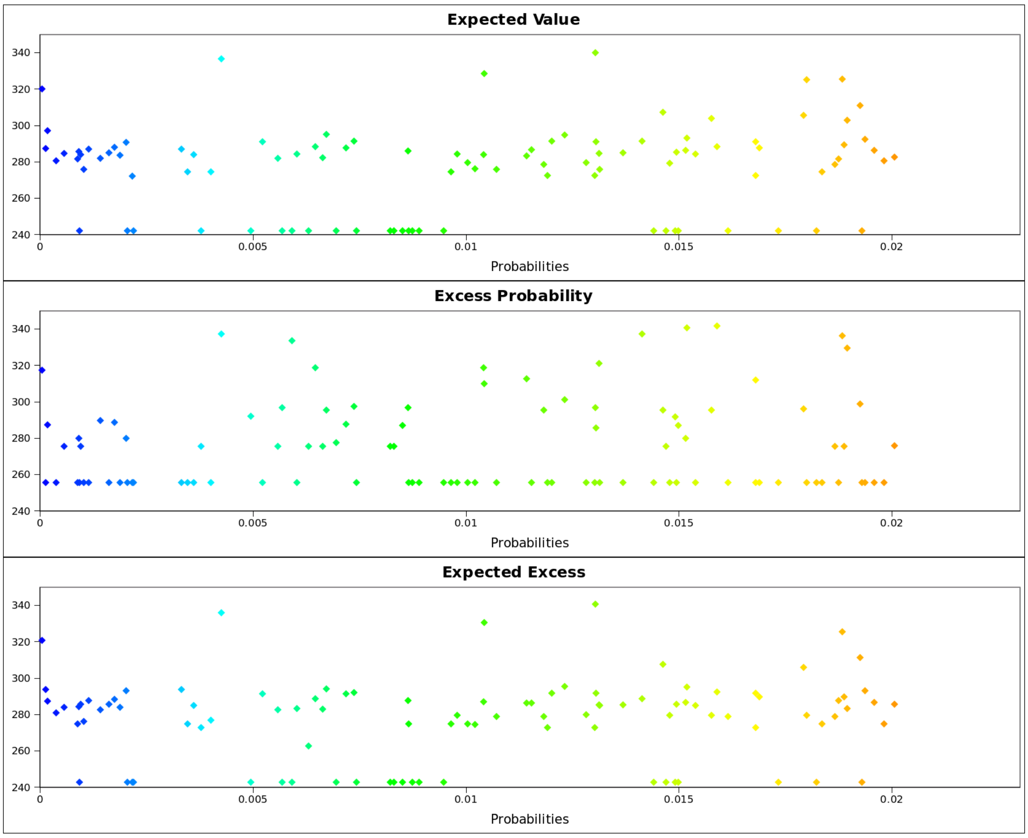

Figure 5.2.: 50-customer instance: Probabilities together with objective values are plotted for each of the hundred scenarios and different risk measures; EV-Model, EP-Model, and EE-Model (from top to bottom) for $\eta = 278.7135$ and $\rho = 50$.

Figure 5.3 shows another example, here the random 15-customer instance with shuffled demand is taken. In figure 5.4 probabilities and objective values for each of the hundred scenarios are displayed. Opposing EP-Model to EV-

Model the number of scenarios exceeding η can be reduced from 15 to 6. For the EE-Model another observation can be made. We get a solution with no variability, that means for each scenario we gain the same objective value.

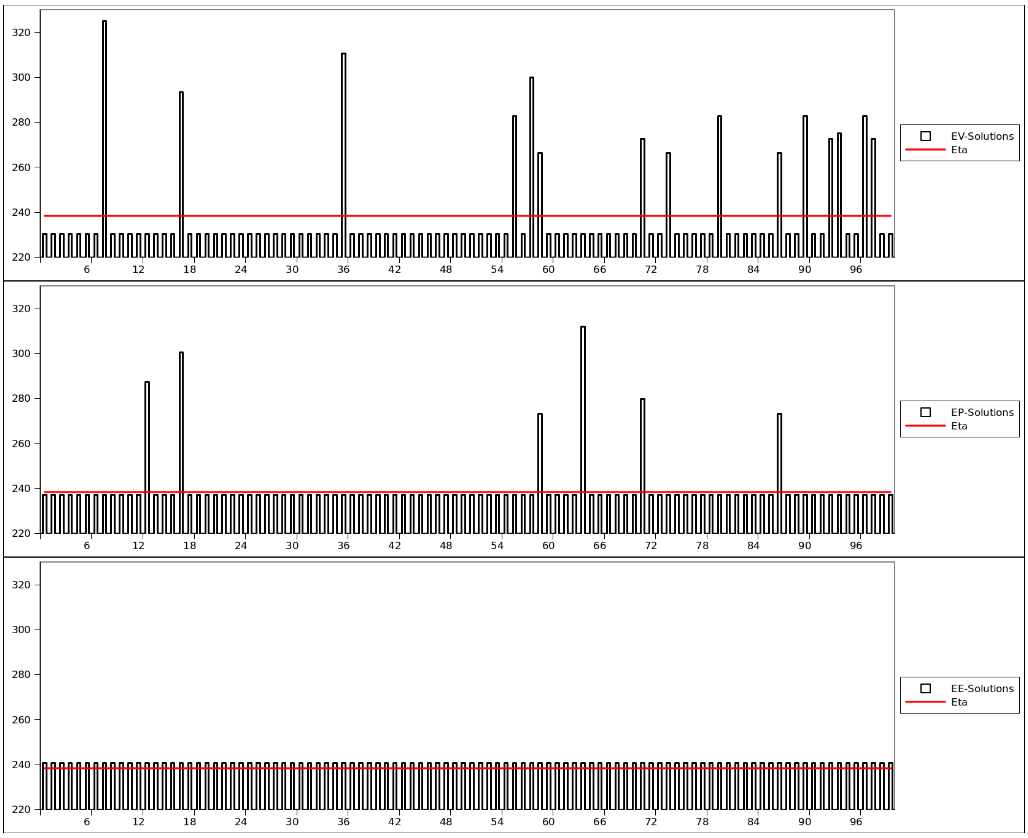

Figure 5.3.: 15-customer instance: Objective values for each of the hundred scenarios are rendered with bar charts for the solutions given by the EV-Model, EP-Model, and EE-Model (from top to bottom) for $\eta = 238.3877$ and $\rho = 50$.

The calculated route for the EE-Model serves all customers without a failure. The recourse costs only arise, because the route costs exceed η. Hence, the recourse costs are independent of the random variables:

$$\mathbb{E}_\omega[f(x,\omega)] + \rho \cdot \mathbb{E}_\omega[\max\{f(x,\omega) - \eta, 0\}] \tag{5.2}$$

$$= c^T x + \underbrace{\sum_{\omega\in\Omega} \pi_\omega q^T y_\omega}_{=0} + \rho \cdot \max\{c^T x + \underbrace{\sum_{\omega\in\Omega} \pi_\omega q^T y_\omega}_{=0} - \eta, 0\} \tag{5.3}$$

$$= c^T x + \rho \cdot \max\{\underbrace{c^T x - \eta}_{>0}, 0\} \tag{5.4}$$

$$= c^T x + \rho \cdot (c^T x - \eta) \tag{5.5}$$

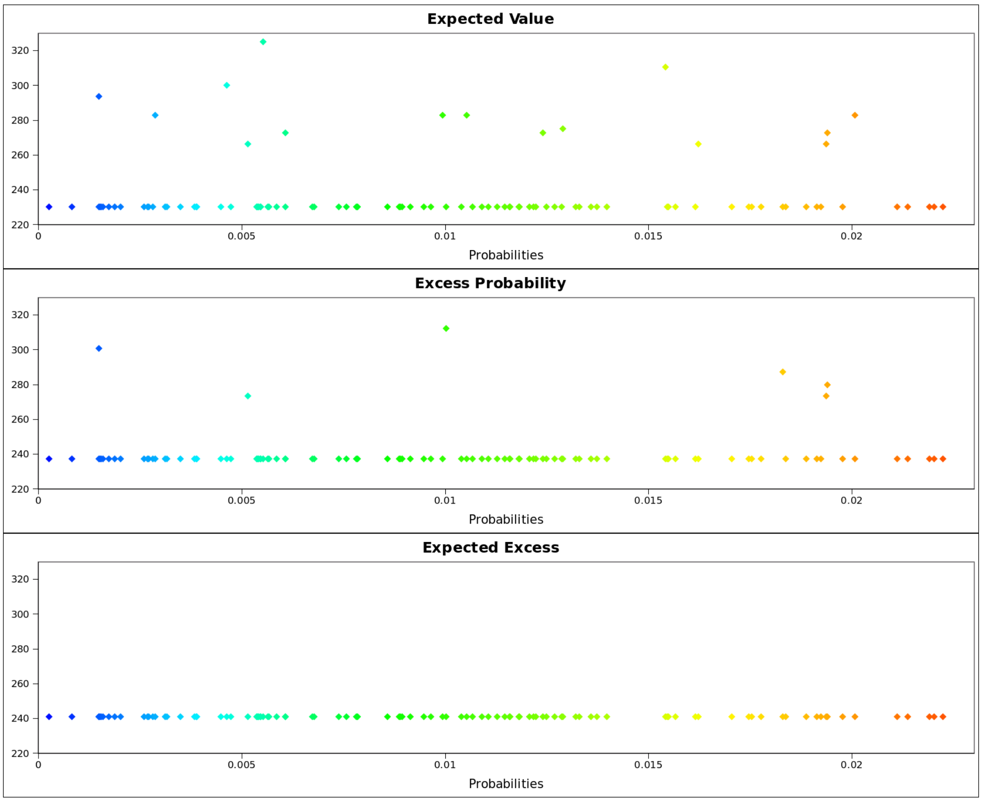

Figure 5.4.: 15-customer instance: Probabilities together with objective values are plotted for each of the hundred scenarios and different risk measures; EV-Model, EP-Model, and EE-Model (from top to bottom) for $\eta = 238.3877$ and $\rho = 50$.

For the random 15-customer instance with shuffled demand and 100 scenarios, we finally report the objective values and recourse costs of the different routes calculated by the stochastic models EV-Model, EP-Model, and EE-Model, respectively, in table 5.12. Although, there exists a route serving all customers without a failure for all realizations of the pickup data, for the EV-Model and the EP-Model this route is not optimal. Due to the fact, that recourse caused by scenarios with small probabilities increase the objective value only little for the EV-Model and also for EP-Model, if η is not exceeded, it could be beneficial to accept routes that have failures for some realizations.

Table 5.12.: Cross-Check for random 15-customer instance with shuffled demand and 100 scenarios; $\rho = 50$

	EP-Model		EV-Model		EE-Model	
	objective	recourse-costs	objective	recourse-costs	objective	recourse-costs
EP-solution	**244.429399**	7.154578	240.741112	3.466291	411.380994	174.106174
EV-solution	246.489517	16.212642	**238.387741**	8.110866	578.21894	347.942065
EE-solution	290.796112	50.00	240.796105	0.00	**361.216358**	120.420254

6. Conclusion

We have developed an exact algorithm for the vehicle routing problem with simultaneous delivery and stochastic pickup. The problem was formulated within the framework of stochastic integer programming and solved by means of the Integer L-Shaped algorithm. To our knowledge, this is the first two-stage stochastic two index formulation for this type of problem. Furthermore, the recourse policy which is performed for the problem examined, has never been considered.

Our results show that the proposed algorithm determines promising routes at the beginning of the solution procedure. However, the challenge in solving certain instances to optimality resides in tightening the lower bound that is produced in the search process. Especially for the risk averse implementation the resulting gaps are very high and could only be seen as a first step towards combining mean risk models and SVRPs. However, the effect of the different risk measures becomes apparent (see figures 5.1 and 5.3).

The generalized definition of partial routes, which is provided in [36] by Jabali, Rei, Gendreau and Laporte does not give a competitive edge, unfortunately. However, the use of a generalized traveling salesman problem instead of creating an artificial node for computing a lower bound on the cost of recourse presented section 4.3.2 has a beneficial effect. For future research the construction of disaggregated cuts, which are used for example by Côté, Potvin and, Gendreau in [16] will be of great interest.

A. Appendix

Tables A.1 - A.21 display the detailed computational results for the 4 different implementations. In column three the gap at the root node is depicted. Column four denotes the CPU time. Columns five to eight display the number of subtour elimination constraints, the number of cuts gained from lower bounding functionals, the number of GPRs, and the number of optimality cuts, respectively. The gaps are reported in the last column.

For instances with uniformly and normally distributed pickup demand and a deviation factor of 20% and $\sigma = 0.2$ the 10 and 15 customer instances are solved under a tenth of a second, therefore only results for the 20 and 50 customer instances are reported.

Tables A.22 - A.25 show the results for the EP-Model and the EE-Model with $\rho = 50$ and $\rho = 100$, respectively. The risk averse models were tested for the instances with 100 scenarios and 100% stochastic customer data, only. Column three denotes the number of customers, column four the gap at the root node, and column five the CPU time. Columns six to nine display the number of subtour elimination constraints, the number of cuts gained from lower bounding functionals, the number of GPRs, and the number of optimality cuts, respectively. The gaps are reported in the last column.

Table A.1.: Computational results on instances with shuffled delivery demand using F-Cuts, PR, and artificial nodes.

graph-type	n	scen	root	CPU (sec)	Sub	LBF	OptCut	gap
clustered	10	10	0.00%	0.10	5	1	8	0.00%
		30	8.00%	23.29	38	155	211	0.00%
		50	8.11%	73.79	54	314	413	0.00%
		100	7.10%	76.30	59	278	392	0.00%
		avrg	5.80%	43.37	39	187	256	0.00%
	15	10	0.13%	0.18	15	5	12	0.00%
		30	0.13%	0.27	15	5	8	0.00%
		50	0.30%	0.46	14	6	10	0.00%
		100	3.08%	1.17	19	19	25	0.00%
		avrg	0.91%	0.52	16	9	14	0.00%
	20	10	0.00%	0.07	13	0	1	0.00%
		30	3.90%	1007.00	121	2785	2604	0.00%
		50	2.08%	60.86	35	242	197	0.00%
		100	4.93%	8560.36	283	9146	10524	0.00%
		avrg	2.73%	2407.07	113	3043	3332	0.00%
	50	10	19.21%	18000	532	6516	5635	13.64%
		30	19.10%	18000	538	6505	5669	13.35%
		50	18.54%	18000	559	6731	5961	13.72%
		100	17.73%	18000	548	6273	5437	13.27%
		avrg	18.64%	18000	544	6506	5676	13.50%
		clustered-avrg	7.02%	5112.74	178	2436	2319	3.37%
random	10	10	1.93%	0.28	8	4	9	0.00%
		30	1.11%	0.33	6	2	7	0.00%
		50	0.00%	0.35	5	1	5	0.00%
		100	1.21%	0.56	4	2	7	0.00%
		avrg	1.06%	0.38	6	2	7	0.00%
	15	10	1.08%	0.28	6	5	5	0.00%
		30	0.00%	0.27	3	0	2	0.00%
		50	0.00%	0.60	5	1	3	0.00%
		100	1.93%	2.03	13	11	17	0.00%
		avrg	0.75%	0.79	7	4	7	0.00%
	20	10	5.79%	467.13	115	888	824	0.00%
		30	3.26%	13.63	34	76	93	0.00%
		50	6.47%	17.66	35	96	88	0.00%
		100	4.98%	13.87	27	56	62	0.00%
		avrg	5.12%	128.07	53	279	267	0.00%
	50	10	13.07%	18000	562	4054	5102	1.64%
		30	15.05%	18000	686	3734	4623	3.51%
		50	16.49%	18000	654	4227	4457	3.43%
		100	15.43%	18000	695	3937	4091	3.21%
		avrg	15.01%	18000	649	3988	4568	2.95%
		random-avrg	5.49%	4532.31	179	1068	1212	0.74%
mixed	10	10	0.00%	0.02	5	0	1	0.00%
		30	0.48%	0.25	6	6	8	0.00%
		50	0.00%	0.18	6	0	5	0.00%
		100	1.76%	0.85	8	20	32	0.00%
		avrg	0.56%	0.33	6	6	12	0.00%
	15	10	3.67%	3.56	20	64	63	0.00%
		30	14.52%	3309.13	102	5213	5672	0.00%
		50	17.07%	2477.7	92	4592	4675	0.00%
		100	2.88%	4286.01	95	6265	6842	0.00%
		avrg	12.04%	2519.10	77	4034	4313	0.00%
	20	10	12.00%	18000	227	13782	16809	4.78%
		30	18.34%	18000	289	14170	17524	7.99%
		50	16.56%	18000	268	14295	17593	7.18%
		100	17.85%	18000	246	13241	17579	7.57%
		avrg	16.19%	18000	258	13872	17376	6.88%
	50	10	17.83%	18000	524	6185	5439	15.55%
		30	21.92%	18000	580	6351	5460	17.42%
		50	23.09%	18000	558	6209	5485	18.44%
		100	21.96%	18000	528	6385	5189	17.54%
		avrg	21.20%	18000	548	6282	5393	17.24%
		mixed-avrg	12.50%	9629.86	222	6049	6774	6.03%
overall-avrg solved: 32/48			8.34%	6424.97	193	3184	3435	3.38%

Table A.2.: Computational results on instances with shuffled delivery demand using F-Cuts, PR, and GTSP.

graph-type	n	scen	root	CPU (sec)	Sub	LBF	OptCut	gap
clustered	10	10	0.00%	0.10	5	1	7	0.00%
		30	8.00%	20.94	38	155	210	0.00%
		50	8.11%	67.96	54	314	413	0.00%
		100	7.10%	57.21	59	280	392	0.00%
	avrg		5.80%	36.55	39	188	256	0.00%
	15	10	0.13%	0.18	15	5	10	0.00%
		30	0.13%	0.24	15	5	7	0.00%
		50	0.30%	0.40	14	6	8	0.00%
		100	3.08%	1.17	19	19	25	0.00%
	avrg		0.91%	0.50	16	9	12	0.00%
	20	10	0.00%	0.05	13	0	0	0.00%
		30	3.90%	1067.94	122	2745	2604	0.00%
		50	2.08%	46.36	35	242	197	0.00%
		100	4.93%	8815.33	278	9242	10522	0.00%
	avrg		2.73%	2482.42	112	3057	3331	0.00%
	50	10	19.21%	18000	580	6696	5871	13.60%
		30	19.10%	18000	535	6697	5614	12.47%
		50	18.54%	18000	572	6588	6021	13.45%
		100	17.73%	18000	538	6719	5459	13.24%
	avrg		18.64%	18000	556	6675	5741	13.19%
	clustered-avrg		7.02%	5129.87	181	2482	2335	3.30%
random	10	10	1.93%	0.22	8	4	9	0.00%
		30	1.11%	0.21	6	2	7	0.00%
		50	0.00%	0.23	5	1	5	0.00%
		100	1.21%	0.37	4	2	7	0.00%
	avrg		1.06%	0.26	6	2	7	0.00%
	15	10	1.08%	0.19	6	5	5	0.00%
		30	0.00%	0.18	3	0	2	0.00%
		50	0.00%	0.28	5	1	3	0.00%
		100	1.93%	1.50	13	11	17	0.00%
	avrg		0.75%	0.54	7	4	7	0.00%
	20	10	5.79%	366.58	123	895	809	0.00%
		30	3.26%	12.05	36	77	93	0.00%
		50	6.47%	12.41	34	93	89	0.00%
		100	4.98%	8.58	29	54	56	0.00%
	avrg		5.12%	99.91	56	280	262	0.00%
	50	10	13.07%	18000	637	4231	5034	2.56%
		30	15.05%	18000	672	3869	4838	3.17%
		50	16.49%	18000	699	4422	4472	2.50%
		100	15.43%	18000	762	4379	4423	3.18%
	avrg		15.01%	18000	692	4225	4692	2.85%
	random-avrg		5.49%	4525.18	190	1128	1242	0.71%
mixed	10	10	0.00%	0.01	5	0	0	0.00%
		30	0.48%	0.21	6	6	8	0.00%
		50	0.00%	0.12	6	0	5	0.00%
		100	1.76%	0.86	8	23	31	0.00%
	avrg		0.56%	0.30	6	7	11	0.00%
	15	10	3.67%	2.79	20	64	58	0.00%
		30	14.52%	3041.54	102	5251	5667	0.00%
		50	17.07%	2342.89	92	4509	4676	0.00%
		100	12.88%	3718.42	94	6220	6839	0.00%
	avrg		12.04%	2276.41	77	4011	4310	0.00%
	20	10	12.00%	18000	182	14529	18089	2.00%
		30	18.34%	18000	290	14652	18901	6.92%
		50	16.56%	18000	263	14349	18363	7.16%
		100	17.85%	18000	277	14685	18856	7.46%
	avrg		16.19%	18000	253	14554	18552	5.88%
	50	10	17.83%	18000	495	6185	5643	15.18%
		30	21.92%	18000	607	6438	5554	18.15%
		50	23.09%	18000	579	6391	5487	18.44%
		100	21.96%	18000	576	6671	5374	16.91%
	avrg		21.20%	18000	564	6421	5514	17.17%
	mixed-avrg		12.50%	9569.18	225	6248	7097	5.76%
overall-avrg solved: 32/48			8.34%	6408.08	199	3286	3558	3.26%

Table A.3.: Computational results on instances with shuffled delivery demand using F-Cuts, GPRs, and artificial nodes.

graph-type	n	scen	root	CPU (sec)	Sub	LBF	GPR	OptCut	gap
clustered	10	10	0.00%	0.15	5	1	0	7	0.00%
		30	8.00%	17.05	33	110	0	179	0.00%
		50	8.11%	84.47	58	170	1	413	0.00%
		100	7.10%	69.76	58	186	0	391	0.00%
		avrg	5.80%	42.86	38	117	0	248	0.00%
	15	10	0.13%	0.25	15	4	0	10	0.00%
		30	0.13%	0.35	15	5	0	7	0.00%
		50	0.30%	0.52	14	3	0	8	0.00%
		100	3.08%	1.20	19	11	0	25	0.00%
		avrg	0.91%	0.58	16	6	0	12	0.00%
	20	10	0.00%	0.10	13	0	0	0	0.00%
		30	3.90%	1143.81	123	2479	0	2605	0.00%
		50	2.08%	69.44	39	222	4	197	0.00%
		100	4.93%	6571.39	286	7731	5	10466	0.00%
		avrg	2.73%	1946.19	115	2608	2	3317	0.00%
	50	10	19.21%	18000	550	5661	15	5951	13.63%
		30	19.10%	18000	556	5380	87	5913	13.34%
		50	18.54%	18000	535	5913	28	5653	13.34%
		100	17.73%	18000	501	6527	82	6782	13.18%
		avrg	18.64%	18000	536	5870	53	6075	13.37%
	clustered-avrg		7.02%	4997.41	176	2150	14	2413	3.34%
random	10	10	1.93%	0.35	8	2	0	9	0.00%
		30	1.11%	0.33	6	1	0	7	0.00%
		50	0.00%	0.36	5	1	0	5	0.00%
		100	1.21%	0.56	4	1	0	7	0.00%
		avrg	1.06%	0.40	6	1	0	7	0.00%
	15	10	1.08%	0.32	6	2	0	5	0.00%
		30	0.00%	0.32	3	0	0	2	0.00%
		50	0.00%	0.54	5	1	0	3	0.00%
		100	1.93%	2.48	13	4	0	17	0.00%
		avrg	0.75%	0.92	7	2	0	7	0.00%
	20	10	5.79%	472.93	120	709	3	824	0.00%
		30	3.26%	15.46	37	36	2	93	0.00%
		50	6.47%	18.48	35	52	2	88	0.00%
		100	4.98%	11.77	25	22	0	54	0.00%
		avrg	5.12%	129.66	54	205	2	265	0.00%
	50	10	13.00%	18000	576	3895	16	5187	1.64%
		30	15.05%	18000	686	3777	10	4841	3.45%
		50	16.49%	18000	685	3936	19	4657	2.57%
		100	15.43%	18000	798	3756	6	4379	4.00%
		avrg	15.01%	18000	686	3841	13	4766	2.92%
	random-avrg		5.49%	4537.12	188	1012	4	1261	0.73%
mixed	10	10	0.00%	0.01	5	0	0	0	0.00%
		30	0.48%	0.18	6	6	1	8	0.00%
		50	0.00%	0.18	6	0	0	5	0.00%
		100	1.76%	0.85	8	18	0	31	0.00%
		avrg	0.56%	0.30	6	6	0	11	0.00%
	25	10	3.67%	2.86	19	34	0	58	0.00%
		30	14.52%	3032.2	99	4064	6	5675	0.00%
		50	17.07%	2390.14	87	3502	6	4671	0.00%
		100	12.88%	3809.85	94	5137	6	6836	0.00%
		avrg	12.04%	2308.76	75	3184	4	4310	0.00%
	20	10	12.00%	18000	230	14424	27	22705	4.55%
		30	18.34%	18000	292	13872	20	23459	7.81%
		50	16.56%	18000	270	14508	26	22481	7.00%
		100	17.85%	18000	284	14153	20	22746	7.32%
		avrg	16.19%	18000	269	14239	23	22848	6.67%
	50	10	17.83%	18000	542	6625	41	6137	16.17%
		30	21.92%	18000	590	5806	46	5465	18.18%
		50	23.09%	18000	544	6092	56	5457	15.62%
		100	21.96%	18000	528	6338	88	5197	16.15%
		avrg	21.20%	18000	551	6215	58	5564	16.53%
	mixed-avrg		12.50%	9622.09	221	4345	21	5337	5.93%
overall-avrg solved: 32/48			8.34%	6385.54	575	2502	13	3004	3.33%

Table A.4.: Computational results on instances with shuffled delivery demand using F-Cuts, GPRs, and GTSP.

graph-type	n	scen	root	CPU (sec)	Sub	LBF	GPR	OptCut	gap
clustered	10	10	0.00%	0.09	5	1	0	7	0.00%
		30	8.00%	13.88	33	110	0	179	0.00%
		50	8.11%	62.87	58	170	1	413	0.00%
		100	7.10%	53.59	58	186	0	391	0.00%
		avrg	5.80%	32.61	38	117	0	248	0.00%
	15	10	0.13%	0.15	15	4	0	10	0.00%
		30	0.13%	0.20	15	5	0	7	0.00%
		50	0.30%	0.33	14	3	0	8	0.00%
		100	3.08%	0.70	19	11	0	25	0.00%
		avrg	0.91%	0.34	16	6	0	12	0.00%
	20	10	0.00%	0.06	13	0	0	0	0.00%
		30	3.90%	971.49	124	2475	0	2605	0.00%
		50	2.08%	45.1	39	222	4	197	0.00%
		100	4.93%	8770.79	286	7739	5	10530	0.00%
		avrg	2.73%	2446.86	116	2609	2	3333	0.00%
	50	10	19.21%	18000	560	5682	50	6314	13.62%
		30	19.10%	18000	545	5854	67	6162	13.32%
		50	18.54%	18000	542	5868	25	5885	13.75%
		100	17.73%	18000	527	5231	35	5949	13.24%
		avrg	18.64%	18000	544	5659	44	6078	13.48%
	clustered-avrg		7.02%	5119.95	178	2098	12	2418	3.37%
random	10	10	1.93%	0.48	8	2	0	9	0.00%
		30	1.11%	0.44	6	1	0	7	0.00%
		50	0.00%	0.55	5	1	0	5	0.00%
		100	1.21%	0.78	4	1	0	7	0.00%
		avrg	1.06%	0.56	6	1	0	7	0.00%
	15	10	1.08%	0.46	6	2	0	5	0.00%
		30	0.00%	0.46	3	0	0	2	0.00%
		50	0.00%	0.67	5	1	0	3	0.00%
		100	1.93%	2.67	13	4	0	17	0.00%
		avrg	0.75%	1.06	7	2	0	7	0.00%
	20	10	5.79%	606.05	120	709	3	824	0.00%
		30	3.26%	17.55	37	36	2	93	0.00%
		50	6.47%	20.67	35	52	2	88	0.00%
		100	4.98%	13.76	25	22	0	54	0.00%
		avrg	5.12%	164.51	54	205	2	265	0.00%
	50	10	13.07%	18000	623	4747	24	6292	1.55%
		30	15.05%	18000	772	4860	22	5902	3.36%
		50	16.49%	18000	706	5004	51	5952	2.47%
		100	15.43%	18000	784	4738	1	5712	3.09%
		avrg	15.01%	18000	721	4837	24	5964	2.62%
	random-avrg		5.49%	4541.53	197	1261	7	1561	0.65%
mixed	10	10	0.00%	0.03	5	0	0	0	0.00%
		30	0.48%	0.38	6	6	1	8	0.00%
		50	0.00%	0.28	6	0	0	5	0.00%
		100	1.76%	1.49	8	18	0	31	0.00%
		avrg	0.56%	0.55	6	6	0	11	0.00%
	25	10	3.67%	4.23	19	34	0	58	0.00%
		30	14.52%	3129.05	100	4013	6	5672	0.00%
		50	17.07%	2567.63	83	3454	6	4671	0.00%
		100	12.88%	4239.2	100	5140	6	6836	0.00%
		avrg	12.04%	2485.03	76	3160	4	4309	0.00%
	20	10	12.00%	18000	205	8758	27	12300	5.05%
		30	18.34%	18000	259	9142	22	13644	8.18%
		50	16.56%	18000	268	8580	27	12675	7.47%
		100	17.85%	18000	265	8177	20	13132	7.77%
		avrg	16.19%	18000	249	8664	24	12938	7.12%
	50	10	17.83%	18000	556	8018	43	7539	15.93%
		30	21.92%	18000	568	7748	56	6845	16.74%
		50	23.09%	18000	588	7241	60	6689	17.83%
		100	21.96%	18000	609	7429	90	6462	17.78%
		avrg	21.20%	18000	580	7609	62	6884	17.07%
	mixed-avrg		12.50%	9621.39	228	4860	23	6035	6.05%
overall-avrg solved: 32/48			8.34%	6427.62	201	2740	14	3338	3.36%

Table A.5.: Computational results on instances with 100% random customers, normally distributed pickup demand, and $\sigma = 0.2$ using F-Cuts, PRs, and GTSP.

graph-type	n	scen	root	CPU (sec)	Sub	LBF	OptCut	gap
clustered	20	10	0.00%	0.01	13	0	0	0.00%
		30	3.07%	3.63	27	69	71	0.00%
		50	3.04%	192.89	59	719	513	0.00%
		100	2.41%	119.17	51	520	340	0.00%
	avrg		2.13%	78.92	38	327	231	0.00%
	50	10	4.60%	18000	201	7069	5646	2.12%
		30	3.49%	18000	212	7268	6182	2.41%
		50	6.10%	18000	322	6872	5801	4.80%
		100	8.23%	18000	368	8447	7127	6.51%
	avrg		5.61%	18000	276	7414	6189	3.96%
	clustered-avrg		3.87%	9039.46	157	3871	3210	1.98%
random	20	10	0.00%	0.06	6	0	0	0.00%
		30	0.25%	0.21	6	4	6	0.00%
		50	0.54%	0.27	8	6	10	0.00%
		100	0.69%	0.52	7	6	9	0.00%
	avrg		0.37%	0.27	7	4	6	0.00%
	50	10	5.63%	190.29	108	151	144	0.00%
		30	5.51%	18000	637	4042	4985	2.27%
		50	6.67%	18000	638	4414	4679	2.03%
		100	4.26%	18000	634	3866	4623	2.25%
	avrg		5.52%	13547.57	504	3118	3608	1.64%
	random-avrg		2.94%	6773.92	256	1561	1807	0.82%
mixed	20	10	1.97%	215.84	37	741	437	0.00%
		30	5.44%	18000	235	18700	19731	1.51%
		50	7.05%	18000	239	17466	22523	2.46%
		100	8.29%	18000	278	16114	22839	4.96%
	avrg		5.69%	13553.96	197	13255	16382	2.23%
	50	10	0.50%	31.29	43	162	46	0.00%
		30	1.86%	18000	142	7899	3553	0.27%
		50	2.97%	18000	220	6882	4830	1.71%
		100	4.51%	18000	280	6897	4827	3.11%
	avrg		2.46%	13507.82	171	5460	3314	1.27%
	mixed-avrg		4.07%	13530.89	184	9358	9848	1.75%
overall-avrg solved: 10/24			3.63%	9781.42	199	4930	4955	1.52%

Table A.6.: Computational results on instances with 100% random customers, normally distributed pickup demand, and $\sigma = 0.2$ using F-Cuts, GPRs, and GTSP.

graph-type	n	scen	root	CPU (sec)	Sub	LBF	GPR	OptCut	gap
clustered	20	10	0.00%	0.02	13	0	0	0	0.00%
		30	3.07%	4.86	27	72	11	71	0.00%
		50	3.04%	358.42	61	633	3	513	0.00%
		100	2.46%	164.89	51	502	7	340	0.00%
	avrg		2.14%	132.05	38	302	5	231	0.00%
	50	10	4.60%	18000	219	6379	54	6035	2.10%
		30	3.49%	18000	217	6459	74	6215	2.41%
		50	6.10%	18000	337	6677	21	6302	4.76%
		100	8.23%	18000	386	7033	12	6670	6.97%
	avrg		5.61%	18000	290	6637	40	6306	4.06%
	clustered-avrg		3.87%	9066.02	164	3469	23	3268	2.03%
random	20	10	0.00%	0.09	6	0	0	0	0.00%
		30	0.25%	0.33	6	4	0	6	0.00%
		50	0.54%	0.44	8	6	0	10	0.00%
		100	0.73%	0.62	8	7	1	10	0.00%
	avrg		0.38%	0.37	7	4	0	6	0.00%
	50	10	6.38%	18000	529	4350	15	5251	0.89%
		30	5.51%	18000	709	4509	38	5506	2.69%
		50	6.67%	18000	636	4261	9	5109	1.98%
		100	6.20%	18000	635	4001	3	5410	2.36%
	avrg		6.19%	18000	627	4280	16	5319	1.98%
	random-avrg		3.28%	9000.18	317	2142	8	2663	0.99%
mixed	20	10	1.99%	298.3	39	657	0	437	0.00%
		30	5.37%	18000	189	6570	3	9461	2.59%
		50	7.03%	18000	210	6876	1	9242	4.30%
		100	7.74%	18000	230	7103	10	9173	5.09%
	avrg		5.53%	13574.58	167	5302	4	7078	3.00%
	50	10	0.44%	20.44	45	148	23	45	0.00%
		30	1.79%	18000	174	7617	48	4927	0.78%
		50	2.99%	18000	205	7239	37	5148	1.69%
		100	4.48%	18000	271	6890	19	5472	3.08%
	avrg		2.43%	13505.11	174	5474	32	3898	1.39%
	mixed-avrg		3.98%	13539.84	170	5388	18	5488	2.19%
overall-avrg solved: 10/24			3.71%	10535.35	217	3666	16	3806	1.74%

Table A.7.: Computational results on instances with 50% random customers, normally distributed pickup demand, and $\sigma = 0.7$ using F-Cuts, PRs, and artificial nodes.

graph-type	n	scen	root	CPU (sec)	Sub	LBF	OptCut	gap
clustered	10	10	8.25%	138	57	403	324	0.00%
		30	7.07%	76.65	44	325	231	0.00%
		50	7.78%	100.17	55	342	299	0.00%
		100	7.95%	341.69	88	582	659	0.00%
		avrg	7.76%	164.13	61	413	378	0.00%
	15	10	4.33%	492.83	73	741	811	0.00%
		30	5.57%	9305.47	324	8730	9555	0.00%
		50	4.20%	1440.23	134	2229	2311	0.00%
		100	5.93%	14933.8	394	9985	13016	0.00%
		avrg	5.01%	6543.08	231	5421	6423	0.00%
	20	10	12.12%	18000	537	8746	15725	9.22%
		30	12.90%	18000	547	10016	16046	9.15%
		50	13.41%	18000	502	9976	15285	10.13%
		100	12.64%	18000	496	9483	12883	9.30%
		avrg	12.77%	18000	520	9555	14985	9.45%
	50	10	17.62%	18000	590	6612	5867	16.61%
		30	23.50%	18000	697	5532	5402	21.24%
		50	21.68%	18000	596	5221	5363	19.44%
		100	21.58%	18000	560	5467	5330	18.60%
		avrg	21.09%	18000	611	5708	5490	18.97%
	clustered-avrg		11.66%	10676.80	356	5274	6819	7.11%
random	10	10	2.53%	0.23	2	3	3	0.00%
		30	0.00%	0.05	0	0	1	0.00%
		50	0.00%	0.03	0	0	1	0.00%
		100	0.00%	0.09	0	0	1	0.00%
		avrg	0.63%	0.10	1	1	2	0.00%
	15	10	4.75%	8.27	26	51	55	0.00%
		30	5.54%	21.52	43	99	85	0.00%
		50	6.23%	53.04	52	153	125	0.00%
		100	6.06%	52.47	48	124	115	0.00%
		avrg	5.64%	33.83	42	107	95	0.00%
	20	10	3.80%	50.56	45	128	111	0.00%
		30	8.12%	8060.48	478	8018	6149	0.00%
		50	9.51%	18000	634	10976	10263	1.91%
		100	9.29%	18000	673	11006	10419	1.88%
		avrg	7.68%	11027.76	458	7532	6736	0.95%
	50	10	10.12%	18000	905	4175	4887	7.64%
		30	11.16%	18000	943	4204	4713	7.87%
		50	12.24%	18000	1045	3825	4904	9.96%
		100	11.43%	18000	864	3415	4649	10.25%
		avrg	11.24%	18000	939	3905	4788	8.93%
	random-avrg		6.30%	7265.42	360	2886	2905	2.47%
mixed	10	10	0.00%	0.04	5	0	1	0.00%
		30	1.62%	1.66	10	23	25	0.00%
		50	1.77%	1.4	8	19	24	0.00%
		100	1.74%	1.35	8	19	25	0.00%
		avrg	1.28%	1.11	8	15	19	0.00%
	15	10	10.69%	874.68	58	1832	1047	0.00%
		30	16.47%	18000	226	12994	15802	4.51%
		50	16.28%	18000	236	12804	15905	5.26%
		100	14.12%	18000	201	12637	14473	4.70%
		avrg	14.39%	13718.67	180	10067	11807	4.82%
	20	10	15.12%	18000	270	10994	14141	3.62%
		30	13.45%	18000	319	10892	13926	7.88%
		50	14.13%	18000	348	10589	14019	9.07%
		100	10.69%	18000	324	9808	11951	8.15%
		avrg	13.35%	18000	315	10571	13509	7.45%
	50	10	7.59%	18000	267	6484	5403	3.28%
		30	14.11%	18000	569	5997	5183	12.53%
		50	14.54%	18000	498	6519	4913	12.26%
		100	16.07%	18000	516	6537	5318	14.05%
		avrg	13.08%	18000	462	6384	5204	10.53%
	mixed-avrg		10.52%	12429.95	241	6759	7635	5.40%
overall-avrg solved: 23/48			9.49%	10124.06	319	4973	5786	4.99%

Table A.8.: Computational results on instances with 50% random customers, normally distributed pickup demand, and $\sigma = 0.7$ using F-Cuts, PRs, and GTSP.

graph-type	n	scen	root	CPU (sec)	Sub	LBF	OptCut	gap
clustered	10	10	8.25%	3.04	14	36	37	0.00%
		30	7.07%	65.08	44	338	231	0.00%
		50	7.78%	84.19	56	343	299	0.00%
		100	7.95%	303.39	88	581	659	0.00%
		avrg	7.76%	113.92	50	324	306	0.00%
	15	10	4.33%	536.89	73	741	811	0.00%
		30	5.57%	1138.29	109	1641	1391	0.00%
		50	4.20%	1403.53	137	2221	2310	0.00%
		100	5.93%	15903.16	386	10123	13018	0.00%
		avrg	5.01%	4745.47	176	3682	4382	0.00%
	20	10	12.12%	18000	430	9096	13908	3.18%
		30	12.90%	18000	531	9763	15900	7.85%
		50	13.41%	18000	472	8982	13988	8.14%
		100	12.64%	18000	501	9602	12845	7.41%
		avrg	12.77%	18000	484	9361	14160	6.65%
	50	10	17.62%	18000	490	7060	6098	12.18%
		30	23.50%	18000	619	6277	6054	17.78%
		50	21.68%	18000	623	6306	5885	16.87%
		100	21.58%	18000	577	6458	6045	17.49%
		avrg	21.09%	18000	577	6525	6020	16.08%
	clustered-avrg		11.66%	10214.85	322	4973	6217	5.68%
random	10	10	2.53%	0.32	2	3	3	0.00%
		30	0.00%	0.08	0	0	1	0.00%
		50	0.00%	0.08	0	0	1	0.00%
		100	0.00%	0.08	0	0	1	0.00%
		avrg	0.63%	0.14	0	1	2	0.00%
	15	10	4.75%	9.4	26	51	55	0.00%
		30	5.54%	23.67	45	97	85	0.00%
		50	6.23%	52.35	51	160	125	0.00%
		100	6.06%	44.94	49	123	115	0.00%
		avrg	5.64%	32.59	43	108	95	0.00%
	20	10	3.80%	65.33	50	132	111	0.00%
		30	8.12%	8646.42	479	7947	6149	0.00%
		50	9.51%	18000	598	11393	10256	1.93%
		100	9.29%	18000	639	11019	10222	1.92%
		avrg	7.68%	11177.94	442	7623	6684	0.96%
	50	10	10.12%	18000	900	4755	5067	6.10%
		30	11.16%	18000	875	4630	5123	7.83%
		50	12.24%	18000	1068	4305	4894	8.95%
		100	11.43%	18000	1036	4337	4975	9.71%
		avrg	11.24%	18000	970	4507	5015	8.15%
	random-avrg		6.30%	7302.67	364	3060	2949	2.28%
mixed	10	10	0.00%	0.04	5	0	1	0.00%
		30	1.62%	1.46	10	22	24	0.00%
		50	1.77%	1.32	8	19	24	0.00%
		100	1.74%	1.81	8	19	25	0.00%
		avrg	1.28%	1.16	8	15	18	0.00%
	15	10	10.69%	101.5	29	288	213	0.00%
		30	16.47%	18000	203	15155	10015	0.51%
		50	16.28%	18000	198	13064	14995	3.41%
		100	14.12%	18000	199	14519	16675	4.52%
		avrg	14.39%	13525.38	157	10756	10474	2.11%
	20	10	15.12%	18000	209	11361	12382	1.78%
		30	13.45%	18000	259	10931	13584	5.38%
		50	14.13%	18000	318	10585	12963	6.01%
		100	10.69%	18000	301	10435	13063	7.29%
		avrg	13.35%	18000	272	10828	12998	5.12%
	50	10	7.59%	18000	253	7671	5709	2.52%
		30	14.11%	18000	468	6455	5600	11.70%
		50	14.54%	18000	540	7034	5455	11.58%
		100	16.07%	18000	577	6673	5420	13.28%
		avrg	13.08%	18000	460	6958	5546	9.77%
	mixed-avrg		10.52%	12381.63	224	7139	7259	4.25%
overall-avrg solved: 23/48			9.49%	9966.38	303	5057	5475	4.07%

Table A.9.: Computational results on instances with 50% random customers, normally distributed pickup demand, and $\sigma = 0.7$ using F-Cuts, GPRs, and artificial nodes.

graph-type	n	scen	root	CPU (sec)	Sub	LBF	GPR	OptCut	gap
clustered	10	10	6.31%	25.45	39	160	0	148	0.00%
		30	8.19%	47.37	50	262	0	248	0.00%
		50	6.48%	38.00	50	258	0	218	0.00%
		100	7.52%	127.1	67	449	0	431	0.00%
		avrg	7.12%	59.48	52	282	0	261	0.00%
	15	10	0.97%	0.25	12	1	0	3	0.00%
		30	5.57%	8304.82	322	7897	0	9532	0.00%
		50	4.20%	1089.40	133	1939	1	2310	0.00%
		100	5.94%	18000	342	7984	0	10463	0.50%
		avrg	4.17%	6848.62	202	4455	0	5577	0.12%
	20	10	12.12%	18000	475	6903	3	11052	9.43%
		30	12.88%	18000	511	6332	12	10554	9.44%
		50	13.35%	18000	436	7003	2	9934	10.36%
		100	12.57%	18000	475	5194	12	10407	9.66%
		avrg	12.73%	18000	474	6358	7	10487	9.72%
	50	10	17.58%	18000	615	6381	0	6143	16.17%
		30	23.36%	18000	705	5551	2	6057	21.17%
		50	21.89%	18000	602	5437	2	6015	19.32%
		100	21.63%	18000	683	6141	0	6086	18.47%
		avrg	21.11%	18000	651	5878	1	6075	18.78%
	clustered-avrg		11.29%	10727.02	345	4243	2	5600	7.16%
random	10	10	0.00%	0.05	0	0	0	1	0.00%
		30	0.00%	0.03	0	0	0	1	0.00%
		50	0.00%	0.03	0	0	0	1	0.00%
		100	0.00%	0.03	0	0	0	1	0.00%
		avrg	0.00%	0.04	0	0	0	1	0.00%
	15	10	4.69%	3.72	35	21	0	47	0.00%
		30	5.51%	10.02	38	30	0	80	0.00%
		50	6.23%	21.49	43	49	0	109	0.00%
		100	6.06%	17.28	39	70	0	98	0.00%
		avrg	5.62%	13.13	39	42	0	84	0.00%
	20	10	3.74%	38.45	44	102	0	106	0.00%
		30	7.94%	4730.12	435	5941	2	4691	0.00%
		50	9.25%	18000	555	8083	1	7723	2.14%
		100	9.06%	18000	539	7654	0	7430	1.96%
		avrg	7.50%	10192.14	393	5445	1	4988	1.02%
	50	10	10.13%	18000	880	3997	70	5440	7.59%
		30	11.30%	18000	1017	4089	4	4976	7.78%
		50	12.20%	18000	1023	4153	6	4646	9.85%
		100	11.04%	18000	1027	3347	3	4713	9.94%
		avrg	11.17%	18000	987	3896	21	4944	8.79%
	random-avrg		6.07%	7051.33	355	2346	5	2504	2.45%
mixed	10	10	0.00%	0.06	5	0	0	3	0.00%
		30	1.03%	0.43	6	8	0	11	0.00%
		50	1.25%	0.58	8	13	0	12	0.00%
		100	0.49%	0.40	7	7	0	9	0.00%
		avrg	0.69%	0.37	6	7	0	9	0.00%
	15	10	9.67%	272.03	42	697	0	378	0.00%
		30	15.60%	18000	204	9485	0	10393	3.23%
		50	15.83%	18000	192	8755	0	10824	4.32%
		100	14.02%	18000	174	8951	0	9190	3.25%
		avrg	13.78%	13568.01	153	6972	0	7696	2.70%
	20	10	12.07%	18000	249	6761	4	10334	4.92%
		30	12.97%	18000	297	7050	0	9684	8.16%
		50	13.12%	18000	290	6708	0	9931	9.29%
		100	11.69%	18000	293	6843	0	8668	8.36%
		avrg	12.46%	18000	282	6840	1	9654	7.68%
	50	10	5.81%	18000	274	6857	21	6048	3.22%
		30	14.08%	18000	539	6866	2	5542	12.50%
		50	14.48%	18000	506	5986	10	5522	12.24%
		100	16.03%	18000	514	7020	10	5566	14.03%
		avrg	12.60%	18000	458	6682	11	5670	10.50%
	mixed-avrg		9.88%	12392.09	225	5125	3	5757	5.22%
overall-avrg solved: 22/48			9.08%	10056.81	308	3905	3	4620	4.94%

Table A.10.: Computational results on instances with 50% random customers, normally distributed pickup demand, and $\sigma = 0.7$ using F-Cuts, GPRs, and GTSP.

graph-type	n	scen	root	CPU (sec)	Sub	LBF	GPR	OptCut	gap
clustered	10	10	8.19%	60.89	50	262	0	248	0.00%
		30	6.31%	33.51	39	160	0	148	0.00%
		50	6.48%	51.5	50	258	0	218	0.00%
		100	7.52%	154.1	69	455	0	431	0.00%
		avrg	7.12%	75.00	52	284	0	261	0.00%
	15	10	0.97%	0.4	12	1	0	3	0.00%
		30	5.57%	18000	301	7242	0	8184	0.61%
		50	4.2%	1215.52	135	1942	1	2310	0.00%
		100	5.94%	18000	308	6552	0	8815	1.05%
		avrg	4.17%	9303.98	189	3934	0	4828	0.42%
	20	10	12.12%	18000	439	5811	3	9042	9.58%
		30	12.88%	18000	477	5408	12	8918	9.59%
		50	13.35%	18000	481	5971	2	8577	10.52%
		100	12.57%	18000	446	4812	11	9029	9.73%
		avrg	12.73%	18000	461	5500	7	8892	9.86%
	50	10	17.58%	18000	565	6652	0	6123	16.17%
		30	23.36%	18000	660	5607	2	6355	21.13%
		50	21.89%	18000	599	6291	0	6575	19.28%
		100	21.63%	18000	624	6584	0	6200	18.45%
		avrg	21.11%	18000	612	6284	0	6313	18.76%
	clustered-avrg		11.29%	11344.74	328	4001	2	5074	7.26%
random	10	10	0.00%	0.11	0	0	0	1	0.00%
		30	0.00%	0.07	0	0	0	1	0.00%
		50	0.00%	0.07	0	0	0	1	0.00%
		100	0.00%	0.08	0	0	0	1	0.00%
		avrg	0.00%	0.08	0	0	0	1	0.00%
	15	10	4.69%	5.71	35	21	0	47	0.00%
		30	5.51%	14.68	38	30	0	80	0.00%
		50	6.23%	30.25	43	49	0	109	0.00%
		100	6.06%	22.4	39	70	0	98	0.00%
		avrg	5.62%	18.26	39	42	0	84	0.00%
	20	10	3.74%	51.39	44	102	0	106	0.00%
		30	7.94%	5046.96	431	5927	3	4696	0.00%
		50	9.25%	18000	503	6177	2	6622	2.63%
		100	9.06%	18000	498	6448	0	6496	2.36%
		avrg	7.50%	10274.59	369	4664	1	4480	1.25%
	50	10	10.13%	18000	1010	4111	32	5621	6.97%
		30	11.3%	18000	963	4202	7	5288	7.77%
		50	12.2%	18000	1031	4679	2	5177	9.85%
		100	11.04%	18000	1020	3479	4	4840	9.94%
		avrg	11.17%	18000	1006	4118	11	5232	8.63%
	random-avrg		6.07%	7073.23	353	2206	3	2449	2.47%
mixed	10	10	0.00%	0.07	5	0	0	3	0.00%
		30	1.03%	0.58	6	8	0	11	0.00%
		50	1.25%	0.73	8	13	0	12	0.00%
		100	0.49%	0.52	7	7	0	9	0.00%
		avrg	0.69%	0.47	6	7	0	9	0.00%
	15	10	9.67%	267.11	42	696	0	378	0.00%
		30	15.6%	18000	181	8086	0	8776	3.38%
		50	15.83%	18000	187	7708	0	9144	4.57%
		100	14.02%	18000	172	8161	0	8199	3.43%
		avrg	13.78%	13566.78	146	6163	0	6624	2.85%
	20	10	12.07%	18000	231	6129	7	8957	5.03%
		30	12.97%	18000	298	6487	0	8845	8.26%
		50	13.12%	18000	300	6324	1	8722	9.37%
		100	11.69%	18000	295	6138	0	8297	8.41%
		avrg	12.46%	18000	281	6270	2	8705	7.77%
	50	10	5.81%	18000	267	6264	24	5400	3.26%
		30	14.08%	18000	572	7302	1	5944	12.3%
		50	14.48%	18000	476	6961	28	5733	12.21%
		100	16.03%	18000	512	7202	26	6106	14.01%
		avrg	12.60%	18000	457	6932	20	5796	10.45%
	mixed-avrg		9.88%	12391.81	222	4843	5	5284	5.27%
overall-avrg solved: 21/48			9.08%	10269.93	301	3683	3	4269	5.00%

Table A.11.: Computational results on instances with 100% random customers, normally distributed pickup demand, and $\sigma = 0.7$ using F-Cuts, PRs, and artificial nodes.

graph-type	n	scen	root	CPU (sec)	Sub	LBF	OptCut	gap
clustered	10	10	15.23%	862.35	141	1614	1972	0.00%
		30	17.40%	18000	574	8946	18377	2.69%
		50	16.03%	18000	512	9030	15096	1.41%
		100	18.05%	18000	498	7963	14503	4.12%
		avrg	16.68%	13715.59	431	6888	12487	2.06%
	15	10	0.00%	0.03	11	0	1	0.00%
		30	2.42%	32.65	30	174	160	0.00%
		50	2.91%	236.45	56	478	419	0.00%
		100	5.44%	2838.41	165	3541	3355	0.00%
		avrg	2.69%	776.88	66	1048	984	0.00%
	20	10	7.65%	18000	364	10394	10728	1.09%
		30	18.24%	18000	530	8562	1225	13.68%
		50	17.37%	18000	542	8072	1140	14.51%
		100	19.16%	18000	521	7165	1081	16.42%
		avrg	15.61%	18000	489	8548	3544	11.43%
	50	10	20.14%	18000	591	6227	5923	16.46%
		30	19.52%	18000	646	6153	5865	17.42%
		50	19.80%	18000	638	5535	5744	18.15%
		100	22.04%	18000	522	4720	4879	19.53%
		avrg	20.38%	18000	599	5659	5603	17.89%
	clustered-avrg		13.84%	12623.12	396	5536	5654	7.84%
random	10	10	4.58 %	0.55	9	11	9	0.00%
		30	10.59%	10.71	36	70	78	0.00%
		50	10.40%	16.21	40	101	83	0.00%
		100	9.57%	12.49	30	75	67	0.00%
		avrg	8.79%	9.99	29	64	59	0.00%
	15	10	5.34%	1.3	17	21	27	0.00%
		30	12.51%	3642.45	377	4029	4823	0.00%
		50	11.07%	1381.85	247	2158	2151	0.00%
		100	12.12%	3984.25	355	4030	4305	0.00%
		avrg	10.26%	2252.46	249	2560	2826	0.00%
	20	10	16.74%	18000	692	9146	10788	7.89%
		30	18.16%	18000	785	7292	10520	11.76%
		50	15.61%	18000	775	6608	9349	9.49%
		100	16.68%	18000	720	6320	8680	10.83%
		avrg	16.80%	18000	743	7342	9834	9.99%
	50	10	13.40%	18000	1055	4151	5579	8.78%
		30	13.92%	18000	1099	3883	5150	11.91%
		50	12.95%	18000	989	3453	4851	11.82%
		100	14.36%	18000	1082	3738	4234	13.25%
		avrg	13.66%	18000	1056	3806	4954	11.44%
	random-avrg		12.38%	9565.61	519	3443	4418	5.36%
mixed	10	10	2.90%	6.15	15	87	62	0.00%
		30	3.86%	28.93	22	153	140	0.00%
		50	3.08%	17.51	19	84	97	0.00%
		100	5.67%	215.57	45	445	503	0.00%
		avrg	3.88%	67.04	25	192	200	0.00%
	15	10	12.63%	18000	179	11218	12538	4.05%
		30	16.76%	18000	250	9703	12805	8.78%
		50	15.18%	18000	212	10337	11606	7.76%
		100	17.11%	18000	196	8849	11144	9.14%
		avrg	15.42%	18000	209	10027	12023	7.43%
	20	10	22.10%	18000	460	8852	11615	17.88%
		30	18.99%	18000	409	8234	10263	14.18%
		50	19.67%	18000	393	8637	9872	15.67%
		100	20.83%	18000	419	8173	9163	17.30%
		avrg	20.40%	18000	420	8474	10228	16.26%
	50	10	23.94%	18000	560	5591	5718	16.55%
		30	25.55%	18000	596	5767	5438	19.17%
		50	22.95%	18000	605	5745	5252	18.16%
		100	21.25%	18000	572	5637	4979	17.65%
		avrg	23.42%	18000	583	5685	5347	17.88%
	mixed-avrg		15.78%	13516.76	310	6095	6950	10.39%
overall-avrg solved: 17/48			14.00%	11901.83	408	5025	5674	7.86%

Table A.12.: Computational results on instances with 100% random customers, normally distributed pickup demand, and $\sigma = 0.7$ using F-Cuts, PRs, and GTSP.

graph-type	n	scen	root	CPU (sec)	Sub	LBF	OptCut	gap
clustered	10	10	15.23%	212.69	69	600	601	0.00%
		30	17.40%	18000	545	8642	19580	0.15%
		50	16.03%	18000	531	10101	16451	0.95%
		100	18.05%	18000	539	9981	18081	3.41%
		avrg	16.68%	13553.17	421	7331	13678	1.13%
	15	10	0.00%	0.03	11	0	1	0.00%
		30	2.42%	38.17	33	173	160	0.00%
		50	2.91%	206.64	55	476	419	0.00%
		100	5.44%	2386.12	173	3579	3352	0.00%
		avrg	2.69%	657.74	68	1057	983	0.00%
	20	10	7.65%	2729.31	173	3984	3238	0.00%
		30	18.24%	18000	509	9083	12758	9.93%
		50	17.37%	18000	471	8368	12368	10.31%
		100	19.16%	18000	521	8510	12188	13.61%
		avrg	15.61%	14182.33	418	7486	10138	8.46%
	50	10	20.14%	18000	543.00	6634.00	5834.00	14.00%
		30	19.52%	18000	594	6167	5889	15.87%
		50	19.80%	18000	554	5602	5957	18.11%
		100	22.04%	18000	460	5678	5548	19.24%
		avrg	20.38%	18000	538	6020	5807	16.80%
	clustered-avrg		13.84%	11598.31	361	5474	7652	6.60%
random	10	10	4.58%	0.55	9	11	9	0.00%
		30	10.59%	9.03	34	72	78	0.00%
		50	10.40%	11.00	38	106	83	0.00%
		100	9.57%	7.15	30	71	67	0.00%
		avrg	8.79%	6.93	28	65	59	0.00%
	15	10	5.34%	1.29	17	21	27	0.00%
		30	12.51%	2490.29	319	2977	3464	0.00%
		50	11.07%	1199.03	239	2179	2152	0.00%
		100	12.12%	2668.96	345	3976	4301	0.00%
		avrg	10.26%	1589.89	230	2288	2486	0.00%
	20	10	16.74%	18000	611	8456	10022	4.59%
		30	18.16%	18000	787	7659	10903	10.71%
		50	15.61%	18000	810	6937	10617	9.35%
		100	16.68%	18000	777	8134	10346	10.20%
		avrg	16.80%	18000	746	7796	10472	8.71%
	50	10	13.40%	18000	976	4346	5256	4.79%
		30	13.92%	18000	921	4102	4904	11.04%
		50	12.95%	18000	990	3794	4702	11.25%
		100	14.36%	18000	980	3893	4490	13.22%
		avrg	13.66%	18000	967	4034	4838	10.07%
	random-avrg		12.38%	9399.21	493	3546	4464	4.70%
mixed	10	10	2.90%	0.84	8	11	13	0.00%
		30	3.86%	17.69	21	128	116	0.00%
		50	3.08%	13.72	19	89	97	0.00%
		100	5.67%	145.14	43	442	503	0.00%
		avrg	3.88%	44.35	23	168	182	0.00%
	15	10	12.63%	3464.93	95	5880	3944	0.00%
		30	16.76%	18000	222	10338	13874	6.06%
		50	15.18%	18000	191	11830	12779	4.97%
		100	17.11%	18000	199	10959	13354	7.33%
		avrg	15.42%	14366.23	177	9752	10988	4.59%
	20	10	22.10%	18000	276	8991	11433	7.84%
		30	18.99%	18000	381	9507	11463	10.65%
		50	19.67%	18000	354	10116	11136	10.47%
		100	20.83%	18000	375	9579	11198	14.14%
		avrg	20.40%	18000	346	9548	11308	10.78%
	50	10	23.94%	18000	464	5797	5724	14.05%
		30	25.55%	18000	552	6241	5412	13.77%
		50	22.95%	18000	486	6889	5662	15.44%
		100	21.25%	18000	511	6552	5501	16.28%
		avrg	23.42%	18000	503	6370	5575	14.88%
	mixed-avrg		15.78%	12602.65	262	6459	7013	7.56%
overall-avrg solved: 19/48			14.00%	11200.06	372	5160	6376	6.29%

Table A.13.: Computational results on instances with 100% random customers, normally distributed pickup demand, and $\sigma = 0.7$ using F-Cuts, GPRs, and artificial nodes.

graph-type	n	scen	root	CPU (sec)	Sub	LBF	GPR	OptCut	gap
clustered	10	10	18.16%	668.12	137	1234	0	1630	0.00%
		30	15.96%	18000	490	6724	0	12998	1.27%
		50	14.06%	3863.9	314	4772	0	6703	0.00%
		100	16.17%	18000	427	6695	0	10628	1.82%
	avrg		16.09%	10133.00	342	4856	0	7990	0.77%
	15	10	0.00%	0.03	11	0	0	1	0.00%
		30	1.58%	13.05	31	128	8	105	0.00%
		50	2.59%	128.65	48	414	0	280	0.00%
		100	4.21%	1508.07	138	1881	0	1837	0.00%
	avrg		2.09%	412.45	57	606	2	556	0.00%
	20	10	7.65%	18000	299	7100	24	8410	1.53%
		30	18.24%	18000	675	13753	13	22998	13.17%
		50	17.15%	18000	480	5924	13	9107	14.48%
		100	18.65%	18000	552	5596	5	8689	16.23%
	avrg		15.42%	18000	502	8093	14	12301	11.35%
	50	10	19.90%	18000	588	7163	0	7852	15.69%
		30	20.21%	18000	597	7286	4	7017	16.79%
		50	19.65%	18000	587	6369	1	7272	17.62%
		100	22.59%	18000	595	6308	0	6264	19.01%
	avrg		20.59%	18000	592	6782	1	7101	17.28%
	clustered-avrg		13.55%	11636.36	373	5084	4	6987	7.35%
random	10	10	4.87%	0.32	7	5	0	5	0.00%
		30	9.48%	6.02	24	15	0	47	0.00%
		50	10.37%	7.55	29	49	0	51	0.00%
		100	9.52%	6.10	23	15	0	45	0.00%
	avrg		8.56%	5.00	21	21	0	37	0.00%
	15	10	5.30%	1.28	16	7	0	28	0.00%
		30	11.87%	1887.21	320	3305	1	3356	0.00%
		50	10.72%	887.67	202	1452	0	1494	0.00%
		100	11.82%	2375.86	283	2758	0	3390	0.00%
	avrg		9.93%	1288.01	205	1880	0	2067	0.00%
	20	10	15.99%	18000	612	6149	0	8757	7.73%
		30	17.59%	18000	697	6156	4	8144	11.31%
		50	14.66%	18000	640	5958	12	7773	8.85%
		100	16.09%	18000	1027	14305	4	21350	8.62%
	avrg		16.08%	18000	744	8142	5	11506	9.13%
	50	10	11.89%	18000	1147	5601	4	7275	8.59%
		30	13.66%	18000	1129	5148	8	6947	11.53%
		50	12.76%	18000	1087	5305	13	6553	11.48%
		100	14.17%	18000	1054	4338	5	6048	12.95%
	avrg		13.12%	18000	1104	5098	8	6706	11.14%
	random-avrg		11.92%	9323.25	519	3785	3	5079	5.07%
mixed	10	10	2.90%	4.68	15	62	1	62	0.00%
		30	3.46%	10.42	17	100	0	92	0.00%
		50	2.07%	4.35	13	40	0	42	0.00%
		100	3.92%	31.62	27	92	0	160	0.00%
	avrg		3.09%	12.77	18	74	0	89	0.00%
	15	10	12.15%	18000	171	8598	0	10033	4.03%
		30	17.57%	18000	227	8160	0	9502	8.37%
		50	14.58%	18000	239	17924	1	24097	5.95%
		100	16.92%	18000	196	6900	0	8806	8.49%
	avrg		15.30%	18000	208	10396	0	13110	6.71%
	20	10	18.54%	18000	366	6322	5	9420	13.06%
		30	17.33%	18000	365	6563	0	8671	12.41%
		50	19.16%	18000	425	6753	0	8039	13.92%
		100	20.32%	18000	399	6804	0	7487	15.53%
	avrg		18.84%	18000	389	6610	1	8404	13.73%
	50	10	23.71%	18000	587	8143	2	7105	14.37%
		30	25.43%	18000	668	8174	18	6861	18.67%
		50	22.74%	18000	640	7970	36	6861	17.63%
		100	20.83%	18000	592	7655	27	6186	16.77%
	avrg		23.18%	18000	622	7986	21	6753	16.86%
	mixed-avrg		15.10%	13503.19	309	6266	6	7089	9.32%
overall-avrg solved: 18/48			13.52%	11487.60	400	5045	4	6385	7.25%

Table A.14.: Computational results on instances with 100% random customers, normally distributed pickup demand, and $\sigma = 0.7$ using F-Cuts, GPRs, and GTSP.

graph-type	n	scen	root	CPU (sec)	Sub	LBF	GPR	OptCut	gap
clustered	10	10	18.16%	700.32	140	1259	0	1630	0.00%
		30	15.96%	18000	450	6079	0	11258	2.09%
		50	14.06%	3650.64	324	4835	0	6704	0.00%
		100	16.17%	18000	417	6245	0	9991	2.11%
		avrg	16.09%	10087.74	333	4604	0	7396	1.05%
	15	10	0.00%	0.04	11	0	0	1	0.00%
		30	1.58%	14.64	31	128	8	105	0.00%
		50	2.59%	137.74	48	391	0	280	0.00%
		100	4.21%	1511.91	138	1881	0	1837	0.00%
		avrg	2.09%	416.08	57	600	2	556	0.00%
	20	10	7.65%	18000	275	6020	24	7504	1.69%
		30	18.24%	18000	579	12155	14	18578	13.30%
		50	17.15%	18000	502	5106	12	8347	14.55%
		100	18.65%	18000	503	5149	10	8154	16.30%
		avrg	15.42%	18000	465	7108	15	10646	11.46%
	50	10	19.90%	18000	570	6780	0	7062	15.79%
		30	20.21%	18000	512	5619	7	5656	16.84%
		50	19.65%	18000	536	5776	1	5744	17.65%
		100	22.59%	18000	503	4957	0	5344	19.04%
		avrg	20.59%	18000	530	5783	2	5952	17.33%
	clustered-avrg		13.55%	11625.96	346	4524	5	6137	7.46%
random	10	10	4.87%	0.39	7	5	0	5	0.00%
		30	9.48%	6.87	24	15	0	47	0.00%
		50	10.37%	7.97	29	49	0	51	0.00%
		100	9.52%	7.09	23	15	0	45	0.00%
		avrg	8.56%	5.58	21	21	0	37	0.00%
	15	10	5.30%	1.47	16	7	0	28	0.00%
		30	11.87%	1889.27	315	3277	1	3356	0.00%
		50	10.72%	900.83	194	1495	0	1494	0.00%
		100	11.82%	2324.53	289	2724	0	3389	0.00%
		avrg	9.93%	1279.03	204	1876	0	2067	0.00%
	20	10	15.99%	18000	623	5362	1	7632	7.93%
		30	17.59%	18000	667	5570	6	7758	11.43%
		50	14.66%	18000	623	5519	2	7066	8.97%
		100	16.09%	18000	966	15244	5	20704	8.63%
		avrg	16.08%	18000	720	7924	4	10790	9.24%
	50	10	11.89%	18000	1049	4217	6	4972	8.72%
		30	13.66%	18000	1024	3734	13	4780	11.63%
		50	12.76%	18000	1002	3499	5	4586	11.59%
		100	14.17%	18000	1029	3195	0	4523	13.05%
		avrg	13.12%	18000	1026	3661	6	4715	11.25%
	random-avrg		11.92%	9321.15	493	3370	2	4402	5.12%
mixed	10	10	2.90%	5.08	15	62	1	62	0.00%
		30	3.46%	12.01	17	100	0	92	0.00%
		50	2.07%	4.28	13	40	0	42	0.00%
		100	3.92%	33.38	27	92	0	160	0.00%
		avrg	3.09%	13.69	18	74	0	89	0.00%
	15	10	12.15%	18000	171	7495	0	8853	4.18%
		30	17.57%	18000	207	7383	0	8783	8.47%
		50	14.58%	18000	185	7082	0	8640	7.38%
		100	16.92%	18000	202	6570	0	8368	8.55%
		avrg	15.30%	18000	191	7132	0	8661	7.15%
	20	10	18.54%	18000	372	5962	2	8588	13.13%
		30	17.33%	18000	386	6118	0	8038	12.46%
		50	19.16%	18000	421	6076	0	7892	13.99%
		100	20.32%	18000	412	7146	0	7510	15.56%
		avrg	18.84%	18000	398	6326	0	8007	13.79%
	50	10	23.71%	18000	553	6644	3	5740	14.79%
		30	25.43%	18000	613	6318	16	5331	18.74%
		50	22.74%	18000	629	5818	8	5067	17.68%
		100	20.83%	18000	547	5713	6	5116	16.8%
		avrg	23.18%	18000	586	6123	8	5314	17.00%
	mixed-avrg		15.10%	13503.42	298	4914	2	5518	9.48%
overall-avrg solved: 18/48			13.52%	11483.51	379	4269	3	5352	7.35%

Table A.15.: Computational results on instances with 50% random customers, uniformly distributed pickup demand, and 70% deviation using F-Cuts, PRs, and artificial nodes.

graph-type	n	scen	root	CPU (sec)	Sub	LBF	OptCut	gap
clustered	10	10	9.21%	9.49	27	109	56	0.00%
		30	4.19%	5.38	20	65	38	0.00%
		50	5.01%	26.75	39	151	105	0.00%
		100	4.39%	9.93	26	103	68	0.00%
		avrg	5.70%	12.89	28	107	67	0.00%
	15	10	0.00%	0.02	11	0	1	0.00%
		30	0.00%	0.02	11	0	1	0.00%
		50	0.00%	0.02	11	0	1	0.00%
		100	0.00%	0.02	11	0	1	0.00%
		avrg	0.00%	0.02	11	0	1	0.00%
	20	10	2.39%	150.43	54	587	304	0.00%
		30	5.24%	18000	342	11366	11053	1.21%
		50	4.58%	11146.26	280	9845	7521	0.00%
		100	6.05%	18000	376	10599	11643	2.28%
		avrg	4.57%	11824.17	263	8099	7630	0.87%
	50	10	19.56%	18000	562	5939	5213	14.57%
		30	20.12%	18000	524	5108	5147	16.67%
		50	18.57%	18000	564	4900	5302	15.09%
		100	18.21%	18000	551	5416	5094	15.67%
		avrg	19.12%	18000	550	5341	5189	15.50%
		clustered-avrg	7.35%	7459.27	213	3387	3222	4.09%
random	10	10	0.00%	0.01	0	0	1	0.00%
		30	0.00%	0.01	0	0	1	0.00%
		50	0.00%	0.01	0	0	1	0.00%
		100	0.00%	0.03	0	0	1	0.00%
		avrg	0.00%	0.01	0	0	1	0.00%
	15	10	0.00%	0.01	1	0	1	0.00%
		30	0.00%	0.05	1	0	1	0.00%
		50	0.00%	0.04	1	0	1	0.00%
		100	0.00%	0.15	1	0	2	0.00%
		avrg	0.00%	0.06	1	0	1	0.00%
	20	10	1.79%	1.14	19	22	17	0.00%
		30	4.92%	419.12	96	465	475	0.00%
		50	6.51%	1717.01	220	2105	1718	0.00%
		100	7.15%	2785.32	294	3030	2688	0.00%
		avrg	5.09%	1230.36	157	1406	1224	0.00%
	50	10	9.35%	18000	813	3880	4598	6.69%
		30	11.75%	18000	953	3664	4781	10.10%
		50	10.86%	18000	964	3931	4789	9.66%
		100	10.39%	18000	979	3297	4310	9.25%
		avrg	10.59%	18000	927	3693	4620	8.93%
		random-avrg	3.92%	4807.68	271	1275	1462	2.23%
mixed	10	10	0.00%	0.01	5	0	1	0.00%
		30	0.00%	0.01	5	0	1	0.00%
		50	0.00%	0.01	5	0	1	0.00%
		100	0.00%	0.02	5	0	1	0.00%
		avrg	0.00%	0.01	5	0	1	0.00%
	15	10	6.42%	974.58	53	1708	1002	0.00%
		30	2.48%	16.59	23	128	71	0.00%
		50	6.24%	7373.33	144	8586	5992	0.00%
		100	6.84%	14812.36	153	13450	10092	0.00%
		avrg	5.50%	5794.22	93	5968	4289	0.00%
	20	10	12.69%	18000	296	11015	14116	7.17%
		30	14.77%	18000	343	12383	13305	10.68%
		50	13.48%	18000	312	11849	12902	10.01%
		100	13.55%	18000	352	10689	12454	10.13%
		avrg	13.62%	18000	326	11484	13194	9.50%
	50	10	4.97%	18000	298	6809	5614	3.93%
		30	4.83%	18000	294	6917	5261	3.88%
		50	9.57%	18000	397	6439	5391	8.39%
		100	10.14%	18000	409	6714	4878	9.11%
		avrg	7.38%	18000	350	6720	5286	6.33%
		mixed-avrg	6.62%	10448.56	193	6043	5693	3.96%
overall-avrg solved: 30/48			5.96%	7571.84	226	3568	3459	3.43%

Table A.16.: Computational results on instances with 50% random customers, uniformly distributed pickup demand, and 70% deviation using F-Cuts, PRs, and GTSP.

graph-type	n	scen	root	CPU (sec)	Sub	LBF	OptCut	gap
clustered	10	10	9.21%	6.54	27	109	56	0.00%
		30	4.19%	3.06	18	63	38	0.00%
		50	5.01%	18.24	39	151	105	0.00%
		100	4.39%	5.34	26	103	68	0.00%
		avrg	5.70%	8.29	28	106	67	0.00%
	15	10	0.00%	0.01	11	0	0	0.00%
		30	0.00%	0.01	11	0	0	0.00%
		50	0.00%	0.01	11	0	0	0.00%
		100	0.00%	0.01	11	0	0	0.00%
		avrg	0.00%	0.01	11	0	0	0.00%
	20	10	2.39%	99.33	56	536	304	0.00%
		30	5.24%	18000	387	13692	12747	0.96%
		50	4.58%	6341.26	269	8020	6314	0.00%
		100	6.05%	18000	431	12059	14378	2.04%
		avrg	4.57%	10610.15	286	8577	8436	0.75%
	50	10	19.56%	18000	543	7630	6380	12.61%
		30	20.12%	18000	613	6164	6451	14.49%
		50	18.57%	18000	570	6969	6645	14.15%
		100	18.21%	18000	502	7400	6319	13.12%
		avrg	19.12%	18000	557	7041	6449	13.59%
	clustered-avrg		7.35%	7154.61	220	3931	3738	3.59%
random	10	10	0.00%	0.01	0	0	0	0.00%
		30	0.00%	0.01	0	0	0	0.00%
		50	0.00%	0.01	0	0	0	0.00%
		100	0.00%	0.01	0	0	1	0.00%
		avrg	0.00%	0.01	0	0	1	0.00%
	15	10	0.00%	0.01	1	0	0	0.00%
		30	0.00%	0.02	1	0	1	0.00%
		50	0.00%	0.02	1	0	1	0.00%
		100	0.00%	0.09	1	0	2	0.00%
		avrg	0.00%	0.03	1	0	1	0.00%
	20	10	1.79%	0.78	19	22	17	0.00%
		30	4.92%	255.59	94	450	475	0.00%
		50	6.51%	1398.52	228	2048	1718	0.00%
		100	7.15%	1642.34	280	3051	2688	0.00%
		avrg	5.09%	824.31	155	1393	1224	0.00%
	50	10	9.35%	18000	717	4736	5405	3.78%
		30	11.75%	18000	916	4408	5470	8.49%
		50	10.86%	18000	991	5094	5836	9.13%
		100	10.39%	18000	914	4828	5834	8.38%
		avrg	10.59%	18000	884	4766	5636	7.45%
	random-avrg		3.92%	4706.09	260	1540	1716	1.86%
mixed	10	10	0.00%	0.01	5	0	0	0.00%
		30	0.00%	0.01	5	0	0	0.00%
		50	0.00%	0.01	5	0	0	0.00%
		100	0.00%	0.01	5	0	0	0.00%
		avrg	0.00%	0.01	5	0	0	0.00%
	15	10	6.42%	133.3	37	456	299	0.00%
		30	2.48%	13.48	23	128	71	0.00%
		50	6.24%	6194.63	142	9026	5996	0.00%
		100	6.84%	11649.23	168	13431	10095	0.00%
		avrg	5.50%	4497.66	92	5760	4115	0.00%
	20	10	12.69%	18000	263	11710	16228	2.80%
		30	14.77%	18000	296	13545	15237	7.20%
		50	13.48%	18000	269	13479	14735	6.66%
		100	13.55%	18000	335	12510	15624	9.35%
		avrg	13.62%	18000	291	12811	15456	6.50%
	20	10	4.97%	18000	244	7221	5307	1.82%
		30	4.83%	18000	286	6792	5161	3.88%
		50	9.57%	18000	357	7225	5406	6.48%
		100	10.14%	18000	362	7410	5923	8.34%
		avrg	7.38%	18000	312	7162	5449	5.13%
	mixed-avrg		6.62%	10124.42	175	6433	6255	2.91%
overall-avrg solved: 30/48			5.96%	7328.37	218	3968	3903	2.79%

Table A.17.: Computational results on instances with 50% random customers, uniformly distributed pickup demand, and 70% deviation using F-Cuts, GPRs, and artificial nodes.

graph-type	n	scen	root	CPU (sec)	Sub	LBF	GPR	OptCut	gap
clustered	10	10	6.60%	5.89	22	84	0	49	0.00%
		30	3.94%	3.17	15	54	0	32	0.00%
		50	5.05%	10.21	28	90	0	74	0.00%
		100	3.82%	6.74	26	67	0	67	0.00%
	avrg		4.85%	6.50	23	74	0	56	0.00%
	15	10	0.00%	0.01	11	0	0	0	0.00%
		30	0.00%	0.01	11	0	0	0	0.00%
		50	0.00%	0.01	11	0	0	0	0.00%
		100	0.00%	0.01	11	0	0	0	0.00%
	avrg		0.00%	0.01	11	0	0	0	0.00%
	20	10	2.39%	107.19	58	515	2	304	0.00%
		30	5.24%	18000	288	6986	10	8086	1.71%
		50	4.58%	18000	273	7434	10	6453	0.58%
		100	5.95%	18000	324	6267	5	8184	2.70%
	avrg		4.54%	13526.80	236	5300	7	5757	1.25%
	50	10	19.37%	18000	541	6310	7	6593	14.15%
		30	20.09%	18000	529	5719	2	6482	16.59%
		50	18.65%	18000	601	6291	1	6142	15.00%
		100	17.77%	18000	559	6129	0	6095	15.60%
	avrg		18.97%	18000	558	6112	2	6328	15.34%
	clustered-avrg		7.09%	7883.33	207	2872	2	3035	4.15%
random	10	10	0.00%	0.01	0	0	0	0	0.00%
		30	0.00%	0.01	0	0	0	0	0.00%
		50	0.00%	0.01	0	0	0	0	0.00%
		100	0.00%	0.01	0	0	0	1	0.00%
	avrg		0.00%	0.01	0	0	0	0	0.00%
	15	10	0.00%	0.01	1	0	0	0	0.00%
		30	0.00%	0.03	1	0	0	1	0.00%
		50	0.00%	0.04	1	0	0	1	0.00%
		100	0.00%	0.11	1	0	0	2	0.00%
	avrg		0.00%	0.05	1	0	0	1	0.00%
	20	10	1.79%	0.96	24	9	0	18	0.00%
		30	4.92%	310.47	95	372	0	474	0.00%
		50	6.51%	1260.15	216	1635	0	1712	0.00%
		100	7.05%	2115.91	280	2510	0	2655	0.00%
	avrg		5.07%	921.87	154	1132	0	1215	0.00%
	50	10	9.39%	18000	920	4507	23	4964	6.24%
		30	11.64%	18000	1102	4367	14	4700	10.00%
		50	10.81%	18000	952	3468	3	4900	9.61%
		100	10.89%	18000	931	3759	15	4932	9.16%
	avrg		10.68%	18000	976	4025	14	4874	8.75%
	random-avrg		3.94%	4730.48	283	1289	3	1523	2.19%
mixed	10	10	0.00%	0.01	5	0	0	0	0.00%
		30	0.00%	0.01	5	0	0	0	0.00%
		50	0.00%	0.01	5	0	0	0	0.00%
		100	0.00%	0.01	5	0	0	0	0.00%
	avrg		0.00%	0.01	5	0	0	0	0.00%
	15	10	5.86%	670.90	54	1387	0	945	0.00%
		30	1.97%	3.18	18	28	0	34	0.00%
		50	6.34%	4475.55	129	6096	0	5453	0.00%
		100	6.98%	18000	147	8501	0	7121	0.52%
	avrg		5.29%	5787.41	87	4003	0	3388	0.13%
	20	10	11.83%	18000	277	7066	0	9119	7.28%
		30	14.74%	18000	338	7316	0	8800	10.59%
		50	12.51%	18000	328	7446	0	8236	9.93%
		100	13.39%	18000	275	6944	0	8210	9.59%
	avrg		13.12%	18000	304	7193	0	8591	9.35%
	50	10	5.10%	18000	306	6959	32	5030	3.98%
		30	4.83%	18000	285	6465	32	5099	3.89%
		50	9.57%	18000	408	6113	7	4929	8.38%
		100	10.13%	18000	381	5758	7	5286	9.09%
	avrg		7.41%	18000	345	6324	20	5086	6.33%
	mixed-avrg		6.45%	10446.85	185	4380	5	4266	3.95%
overall-avrg solved: 28/48			5.82%	7686.88	225	2847	3	2112	3.43%

Table A.18.: Computational results on instances with 50% random customers, uniformly distributed pickup demand, and 70% deviation using F-Cuts, GPRs, and GTSP.

graph-type	n	scen	root	CPU (sec)	Sub	LBF	GPR	OptCut	gap
clustered	10	10	6.60%	6.49	22	84	0	49	0.00%
		30	3.94%	3.50	15	54	0	32	0.00%
		50	5.05%	11.64	28	90	0	74	0.00%
		100	3.82%	7.57	26	67	0	67	0.00%
		avrg	4.85%	7.30	23	74	0	56	0.00%
	15	10	0.00%	0.01	11	0	0	0	0.00%
		30	0.00%	0.01	11	0	0	0	0.00%
		50	0.00%	0.01	11	0	0	0	0.00%
		100	0.00%	0.01	11	0	0	0	0.00%
		avrg	0.00%	0.01	11	0	0	0	0.00%
	20	10	2.39%	126.58	58	515	2	304	0.00%
		30	5.24%	18000	316	8058	8	9041	1.51%
		50	4.58%	18000	281	8558	10	7445	0.04%
		100	5.95%	18000	343	6991	4	9465	2.48%
		avrg	4.54%	13531.65	250	6030	6	6564	1.01%
	50	10	19.37%	18000	566	5651	1	5658	16.13%
		30	20.09%	18000	534	5026	4	5656	16.63%
		50	18.65%	18000	567	5587	0	6032	15.02%
		100	17.77%	18000	552	5711	0	5555	15.62%
		avrg	18.97%	18000	555	5494	1	5725	15.85%
	clustered-avrg		7.09%	7884.74	210	2900	2	3086	4.21%
random	10	10	0.00%	0.01	0	0	0	0	0.00%
		30	0.00%	0.01	0	0	0	0	0.00%
		50	0.00%	0.01	0	0	0	0	0.00%
		100	0.00%	0.02	0	0	0	1	0.00%
		avrg	0.00%	0.01	0	0	0	0	0.00%
	15	10	0.00%	0.01	1	0	0	0	0.00%
		30	0.00%	0.04	1	0	0	1	0.00%
		50	0.00%	0.03	1	0	0	1	0.00%
		100	0.00%	0.13	1	0	0	2	0.00%
		avrg	0.00%	0.05	1	0	0	1	0.00%
	20	10	1.79%	1.02	24	9	0	18	0.00%
		30	4.92%	360.93	95	372	0	474	0.00%
		50	6.51%	1263.61	226	1632	0	1712	0.00%
		100	7.05%	2030.35	268	2392	0	2655	0.00%
		avrg	5.07%	913.98	153	1101	0	1215	0.00%
	50	10	9.39%	18000	776	4364	14	4898	6.26%
		30	11.64%	18000	1047	3846	16	4803	10.02%
		50	10.81%	18000	934	3904	1	4867	9.63%
		100	10.89%	18000	969	3830	8	4722	9.18%
		avrg	10.68%	18000	932	3986	10	4822	8.77%
	random-avrg		3.94%	4728.51	271	1272	2	1510	2.19%
mixed	10	10	0.00%	0.01	5	0	0	0	0.00%
		30	0.00%	0.01	5	0	0	0	0.00%
		50	0.00%	0.01	5	0	0	0	0.00%
		100	0.00%	0.01	5	0	0	0	0.00%
		avrg	0.00%	0.01	5	0	0	0	0.00%
	25	10	5.86%	697.86	54	1387	0	945	0.00%
		30	1.97%	3.53	18	28	0	34	0.00%
		50	6.34%	4487.69	129	6068	0	5452	0.00%
		100	6.98%	8863.26	145	9044	0	7945	0.00%
		avrg	5.29%	3513.09	86	4132	0	3594	0.00%
	20	10	11.83%	18000	291	8386	2	10401	7.15%
		30	14.74%	18000	342	8721	0	10018	10.47%
		50	12.51%	18000	323	8530	0	9170	9.85%
		100	13.39%	18000	279	8319	0	9945	9.47%
		avrg	13.12%	18000	309	8489	0	9884	9.23%
	50	10	5.10%	18000	283	6914	30	5308	3.97%
		30	4.83%	18000	296	7074	27	5360	3.87%
		50	9.57%	18000	372	6515	9	5294	8.34%
		100	10.13%	18000	384	6780	7	5038	9.08%
		avrg	7.41%	18000	334	6821	18	5250	6.32%
	mixed-avrg		6.45%	9878.27	184	4860	5	4682	3.89%
overall-avrg solved: 29/48			5.68%	7497.17	222	3011	3	3093	3.43%

Table A.19.: Computational results on instances with 100% random customers, uniformly distributed pickup demand, and 70% deviation using F-Cuts, PRs, and artificial nodes.

graph-type	n	scen	root	CPU (sec)	Sub	LBF	OptCut	gap
clustered	10	10	5.49%	9.28	27	112	64	0.00%
		30	13.32%	1942.08	271	3305	3906	0.00%
		50	12.21%	2003.02	253	2762	3861	0.00%
		100	13.14%	3026.98	293	3822	5521	0.00%
		avrg	11.04%	1745.34	211	2500	3338	0.00%
	15	10	1.04%	1.25	19	33	21	0.00%
		30	2.58%	158.91	57	463	356	0.00%
		50	2.30%	81.8	45	393	273	0.00%
		100	5.44%	5667.1	270	6420	7402	0.00%
		avrg	2.84%	1477.27	98	1827	2013	0.00%
	20	10	10.74%	18000	433	9821	13295	5.11%
		30	14.51%	18000	479	9760	13278	9.51%
		50	16.50%	18000	548	9402	15725	9.80%
		100	16.15%	18000	556	9789	14457	12.46%
		avrg	14.47%	18000	504	9693	14189	9.22%
	50	10	13.43%	18000	436	7048	5728	10.77%
		30	15.53%	18000	526	6212	5499	13.62%
		50	16.61%	18000	600	6105	5298	14.83%
		100	17.55%	18000	582	5322	5049	16.26%
		avrg	15.78%	18000	536	6172	5394	13.87%
		clustered-avrg	11.03%	9805.65	337	5048	6233	5.77%
random	10	10	3.49%	0.57	9	8	9	0.00%
		30	0.43%	0.35	2	1	5	0.00%
		50	5.17%	4.47	19	27	19	0.00%
		100	6.34%	6.54	19	34	29	0.00%
		avrg	3.86%	2.98	12	18	16	0.00%
	15	10	0.00%	0.05	2	0	1	0.00%
		30	1.50%	0.51	7	8	4	0.00%
		50	3.42%	2.48	17	21	27	0.00%
		100	3.19%	2.7	14	17	19	0.00%
		avrg	2.03%	1.42	10	12	13	0.00%
	20	10	12.72%	18000	719	10522	13258	5.55%
		30	6.29%	1072.97	167	1743	1343	0.00%
		50	6.77%	1391.51	191	1774	1374	0.00%
		100	8.54%	14986.71	579	9015	9571	0.00%
		avrg	8.58%	8862.80	414	5764	6386	1.39%
	50	10	7.57%	18000	646	4917	5398	1.86%
		30	12.44%	18000	930	3964	4804	7.73%
		50	12.37%	18000	894	3737	5098	7.39%
		100	12.28%	18000	994	3229	4900	9.04%
		avrg	11.16%	18000	866	3962	5050	6.50%
		random-avrg	6.41%	6716.80	326	2439	2866	1.97%
mixed	10	10	0.00%	0.16	6	1	7	0.00%
		30	0.00%	0.03	5	0	2	0.00%
		50	2.01%	1.66	12	33	29	0.00%
		100	1.53%	0.75	6	20	13	0.00%
		avrg	0.89%	0.65	7	14	13	0.00%
	15	10	18.44%	18000	244	13632	14987	5.02%
		30	18.92%	18000	258	12980	15021	8.77%
		50	16.27%	18000	236	12413	14705	6.58%
		100	13.24%	18000	205	11952	13826	5.43%
		avrg	16.72%	18000	236	12744	14635	6.45%
	20	10	13.58%	18000	259	10813	14374	5.91%
		30	19.54%	18000	405	11180	12549	12.92%
		50	18.14%	18000	378	10518	11955	12.82%
		100	18.15%	18000	402	10394	11419	15.23%
		avrg	17.35%	18000	361	10726	12574	11.72%
	50	10	7.48%	18000	289	6706	5651	5.08%
		30	11.77%	18000	466	6725	5341	10.46%
		50	12.60%	18000	414	6631	5079	10.66%
		100	13.19%	18000	466	6131	4715	11.55%
		avrg	11.26%	18000	409	6548	5196	9.44%
		mixed-avrg	11.55%	13500.16	253	7508	8105	6.90%
overall-avrg solved: 23/48			9.66%	10007.54	305	4998	5735	4.88%

Table A.20.: Computational results on instances with 100% random customers, uniformly distributed pickup demand, and 70% deviation using F-Cuts, PRs, and GTSP.

graph-type	n	scen	root	CPU (sec)	Sub	LBF	OptCut	gap
clustered	10	10	5.49%	1.74	10	24	16	0.00%
		30	13.32%	917.61	201	2088	2123	0.00%
		50	12.21%	925.37	176	1635	2097	0.00%
		100	13.14%	2486.88	299	4091	5520	0.00%
		avrg	11.04%	1082.90	172	1960	2439	0.00%
	15	10	1.04%	1.79	20	31	21	0.00%
		30	2.58%	19.08	33	155	112	0.00%
		50	2.30%	19.08	31	171	124	0.00%
		100	5.44%	6053.87	280	6400	7403	0.00%
		avrg	2.84%	1523.45	91	1689	1915	0.00%
	20	10	10.74%	18000	345	12145	9475	0.78%
		30	14.51%	18000	504	10440	15871	8.36%
		50	16.50%	18000	443	9492	13865	8.12%
		100	16.15%	18000	497	10052	15687	9.10%
		avrg	14.47%	18000	447	10532	13724	6.59%
	50	10	13.43%	18000	355	6619	5779	5.44%
		30	15.53%	18000	537	6359	5406	12.81%
		50	16.61%	18000	580	6322	5240	14.50%
		100	17.55%	18000	587	5489	5145	15.80%
		avrg	15.78%	18000	515	6197	5392	12.14%
	clustered-avrg		11.03%	9651.59	306	5095	5868	4.68%
random	10	10	3.49%	0.79	9	8	9	0.00%
		30	0.43%	0.41	2	1	5	0.00%
		50	5.17%	4.26	19	27	19	0.00%
		100	6.34%	5.12	19	34	29	0.00%
		avrg	3.86%	2.65	12	18	16	0.00%
	15	10	0.00%	0.04	2	0	1	0.00%
		30	1.50%	0.62	7	8	4	0.00%
		50	3.42%	2.98	17	21	27	0.00%
		100	3.19%	2.87	14	17	19	0.00%
		avrg	2.03%	1.63	10	12	13	0.00%
	20	10	12.72%	18000	673	13485	12102	1.27%
		30	6.29%	1144.01	176	1714	1344	0.00%
		50	6.77%	1454.23	203	1719	1375	0.00%
		100	8.54%	13119.89	575	9096	9556	0.00%
		avrg	8.58%	8429.53	407	6504	6094	0.32%
	50	10	7.57%	18000	611	4956	5311	1.85%
		30	12.44%	18000	937	3815	4438	7.07%
		50	12.37%	18000	979	3725	5002	7.41%
		100	12.28%	18000	928	3394	4608	8.84%
		avrg	11.16%	18000	864	3972	4840	6.29%
	random-avrg		6.41%	6608.45	323	2626	2741	1.65%
mixed	10	10	0.00%	0.23	6	1	7	0.00%
		30	0.00%	0.04	5	0	2	0.00%
		50	2.01%	2.04	12	34	29	0.00%
		100	1.53%	1.08	6	20	13	0.00%
		avrg	0.89%	0.85	7	14	13	0.00%
	15	10	18.44%	5105.63	135	7275	5156	0.00%
		30	18.92%	18000	206	12950	14486	6.18%
		50	16.27%	18000	230	13666	16316	4.88%
		100	13.24%	18000	214	13806	16133	5.03%
		avrg	16.72%	14776.41	196	11924	13023	4.02%
	20	10	13.58%	732.2	42	1143	792	0.00%
		30	19.54%	18000	262	10691	11506	2.37%
		50	18.14%	18000	295	11153	13535	8.21%
		100	18.15%	18000	347	11990	11943	12.15%
		avrg	17.35%	13683.05	236	8744	9444	5.68%
	50	10	7.48%	18000	300	7038	5698	4.63%
		30	11.77%	18000	490	6458	4991	8.86%
		50	12.60%	18000	432	6891	4971	9.69%
		100	13.19%	18000	444	6466	4810	10.32%
		avrg	11.26%	18000	416	6713	5118	8.38%
	mixed-avrg		11.55%	11615.08	214	6849	6899	4.52%
overall-avrg solved: 25/48			9.66%	9291.71	281	4857	5169	3.62%

Table A.21.: Computational results on instances with 100% random customers, uniformly distributed pickup demand, and 70% deviation using F-Cuts, GPRs, and GTSP.

graph-type	n	scen	root	CPU (sec)	Sub	LBF	GPR	OptCut	gap
clustered	10	10	5.15%	10.02	19	91	0	65	0.00%
		30	12.71%	1730	215	2210	0	2682	0.00%
		50	11.14%	1868.67	240	2047	0	3003	0.00%
		100	12.00%	2354.78	285	2824	0	4431	0.00%
		avrg	10.25%	1490.87	190	1793	0	2545	0.00%
	15	10	1.16%	2.27	16	32	2	21	0.00%
		30	2.58%	217.51	58	351	4	356	0.00%
		50	2.30%	123.03	50	305	0	273	0.00%
		100	5.36%	5592.77	263	4846	5	7270	0.00%
		avrg	2.85%	1483.90	97	1384	3	1980	0.00%
	20	10	10.78%	18000	377	6239	16	9771	5.40%
		30	14.51%	18000	422	5683	4	9227	9.88%
		50	16.50%	18000	451	6239	6	8940	10.18%
		100	16.15%	18000	498	5240	5	9519	12.81%
		avrg	14.48%	18000	437	5850	8	9364	9.57%
	50	10	15.55%	18000	458	6686	4	6762	10.65%
		30	15.40%	18000	538	7125	19	7092	12.98%
		50	16.57%	18000	533	6461	4	6305	14.65%
		100	17.03%	18000	546	6692	5	6650	15.97%
		avrg	16.14%	18000	519	6741	8	6702	13.56%
	clustered-avrg		10.93%	9743.69	311	3942	5	5148	5.78%
random	10	10	3.49%	0.66	9	4	0	9	0.00%
		30	0.54%	0.51	0	2	0	2	0.00%
		50	4.97%	3.10	15	7	0	17	0.00%
		100	5.82%	4.67	18	7	0	26	0.00%
		avrg	3.71%	2.23	10	5	0	14	0.00%
	15	10	0.00%	0.07	1	0	0	1	0.00%
		30	1.52%	0.48	10	2	0	4	0.00%
		50	3.42%	3.09	17	8	0	27	0.00%
		100	3.41%	2.90	13	6	0	19	0.00%
		avrg	2.09%	1.64	10	4	0	13	0.00%
	20	10	11.88%	18000	586	5144	0	8051	5.61%
		30	5.76%	1066.63	153	828	0	919	0.00%
		50	6.25%	1317.62	161	952	0	1080	0.00%
		100	8.15%	18000	453	4764	2	6427	0.71%
		avrg	8.01%	9596.06	338	2922	0	4119	1.58%
	50	10	7.57%	18000	642	3731	5	5178	1.91%
		30	11.30%	18000	907	3759	16	5120	7.66%
		50	12.37%	18000	975	3964	2	4907	7.32%
		100	12.28%	18000	924	3890	11	4942	8.96%
		avrg	10.88%	18000	862	3836	8	5037	6.46%
	random-avrg		6.17%	6899.98	305	1692	2	2296	2.01%
mixed	10	10	0.00%	0.49	6	1	0	7	0.00%
		30	0.00%	0.08	5	0	0	2	0.00%
		50	1.99%	3.85	13	28	0	26	0.00%
		100	1.46%	1.78	6	13	0	12	0.00%
		avrg	0.86%	1.55	8	10	0	12	0.00%
	15	10	17.77%	18000	173	8332	0	10083	5.34%
		30	18.54%	18000	214	8061	1	9471	9.16%
		50	16.06%	18000	251	20104	0	26526	5.52%
		100	12.68%	18000	173	7616	0	8790	5.59%
		avrg	16.26%	18000	203	11028	0	13718	6.40%
	20	10	9.92%	18000	242	7855	0	8500	5.87%
		30	17.43%	18000	345	6855	0	8541	12.27%
		50	16.71%	18000	376	6715	0	8203	12.09%
		100	18.51%	18000	399	6857	0	7830	14.42%
		avrg	15.64%	18000	340	7070	0	8268	11.16%
	50	10	7.42%	18000	306	6546	32	5081	4.52%
		30	12.00%	18000	467	6695	33	4901	10.49%
		50	11.95%	18000	399	7046	12	5062	10.62%
		100	12.93%	18000	446	6534	8	5035	11.35%
		avrg	11.07%	18000	404	6705	21	5020	9.24%
	mixed-avrg		10.96%	13500.39	239	6204	5	6754	6.70%
overall-avrg solved: 22/48			9.35%	10048.02	285	3946	4	4733	4.83%

Table A.22.: Computational results for the EP-Model with $\rho = 50$.

data-type	graph-type	n	root	CPU (sec)	Sub	LBF	OptCut	gap
shuffle	clustered	10	23.99%	402.40	127	936	1444	0.00%
		15	6.61%	1.98	21	42	57	0.00%
		20	9.32%	18000	435	10162	16514	4.26%
		50	25.68%	18000	736	8126	7634	20.53%
	avrg		16.40%	9101.09	330	4817	6412	6.20%
	random	10	3.43%	0.81	7	8	15	0.00%
		15	5.21%	8.79	30	82	69	0.00%
		20	9.32%	15.02	38	80	92	0.00%
		50	21.63%	18000	900	5312	6027	5.93%
	avrg		9.90%	4506.15	244	1371	1551	1.48%
	mixed	10	3.37%	1.26	10	29	40	0.00%
		15	17.38%	18000	169	13253	18930	1.82%
		20	25.28%	18000	346	12291	15009	12.64%
		50	29.24%	18000	697	8778	7683	20.36%
	avrg		18.82%	13500.32	306	8588	10416	8.71%
norm0.7	clustered	10	40.58%	18000	657	9226	20624	31.43%
		15	12.24%	18000	506	8218	15463	7.57%
		20	32.86%	18000	808	8922	13078	30.22%
		50	29.34%	18000	552	7003	6947	26.43%
	avrg		28.75%	18000.00	631	8342	14028	23.91%
	random	10	14.34%	4755.27	259	5497	11675	0.00%
		15	28.37%	18000	380	10967	14893	20.59%
		20	30.06%	18000	560	10440	11763	27.64%
		50	26.37%	18000	508	7682	6838	24.42%
	avrg		24.79%	14688.82	427	8647	11292	18.16%
	mixed	10	18.55%	617.11	174	1306	1626	0.00%
		15	18.92%	18000	657	9151	13454	7.63%
		20	23.71%	18000	908	8169	10991	18.14%
		50	19.13%	18000	1021	4255	5103	17.52%
	avrg		20.08%	13654.28	690	5720	7794	10.82%
uni70	clustered	10	32.40%	18000	654	7601	16634	20.73%
		15	5.53%	7114.59	270	6408	8162	0.00%
		20	17.19%	18000	514	7806	10176	13.77%
		50	25.39%	18000	610	8066	7678	24.23%
	avrg		20.13%	15278.65	512	7470	10663	14.68%
	random	10	27.57%	18000	372	6742	18925	15.46%
		15	32.50%	18000	463	9287	11734	27.28%
		20	22.27%	18000	469	9236	10119	19.39%
		50	22.09%	18000	636	8543	7263	20.96%
	avrg		26.11%	18000	485	8452	12010	20.77%
	mixed	10	24.84%	9662.31	570	7089	15743	0.00%
		15	19.93%	18000	642	7267	11594	9.23%
		20	22.17%	18000	867	7107	10176	16.56%
		50	16.65%	18000	1104	4787	5912	12.57%
	avrg		20.90%	15915.58	796	6563	10856	9.59%
	overall-avrg solved: 10/36		20.65%	13627.21	491	6663	9447	12.70%

Table A.23.: Computational results for the EP-Model with $\rho = 100$.

data-type	graph-type	n	root	CPU (sec)	Sub	LBF	OptCut	gap
shuffle	clustered	10	19.69%	991.84	247	2486	4651	0.00%
		15	9.87%	8.36	26	90	113	0.00%
		20	13.30%	18000	461	10389	16707	5.50%
		50	32.23%	18000	853	7788	7196	21.69%
	avrg		18.77%	9250.05	397	5188	7167	6.80%
	random	10	1.82%	1.51	9	35	31	0.00%
		15	22.48%	18000	200	13211	19681	4.04%
		20	31.74%	18000	394	12743	15245	16.72%
		50	34.00%	18000	642	8458	7633	23.66%
	avrg		22.51%	13500.38	311	8612	10648	11.11%
	mixed	10	5.66%	2.07	16	22	28	0.00%
		15	8.23%	52.13	49	153	179	0.00%
		20	13.28%	46.31	44	120	142	0.00%
		50	26.98%	18000	1154	5236	6072	8.60%
	avrg		13.54%	4525.13	316	1383	1605	2.15%
norm0.7	clustered	10	52.97%	18000	655	9689	21550	45.56%
		15	18.58%	18000	574	8288	16050	14.19%
		20	41.22%	18000	828	8677	12034	38.41%
		50	35.00%	18000	559	7009	7035	31.4%
	avrg		36.94%	18000.00	654	8416	14167	32.39%
	random	10	20.35%	18000	335	9256	23387	5.63%
		15	35.62%	18000	475	10722	14517	27.84%
		20	35.66%	18000	650	10488	11808	33.43%
		50	29.83%	18000	559	7052	7000	28.1%
	avrg		30.36%	18000	505	9380	14178	23.75%
	mixed	10	25.31%	10563.49	573	7353	16017	0.00%
		15	24.28%	18000	751	8774	14566	13.73%
		20	28.67%	18000	967	7736	11282	23.38%
		50	22.49%	18000	1079	4275	5113	21.02%
	avrg		25.19%	16140.87	843	7035	11745	14.54%
uni70	clustered	10	43.89%	18000	670	7605	17192	33.24%
		15	5.53%	6780.04	265	6424	8163	0.00%
		20	17.19%	18000	541	7887	10292	13.73%
		50	31.86%	18000	610	8110	7654	30.65%
	avrg		24.62%	15195.01	522	7507	10825	19.41%
	random	10	37.43%	18000	645	7033	16347	18.84%
		15	31.79%	18000	786	7564	12692	22.56%
		20	32.07%	18000	1011	6321	10397	27.22%
		50	20.47%	18000	1139	4633	5940	15.75%
	avrg		30.44%	18000	895	6388	11344	21.09%
	mixed	10	42.43%	18000	572	7212	17666	33.33%
		15	44.06%	18000	553	9441	12365	39.73%
		20	22.27%	18000	455	9200	10100	19.40%
		50	29.31%	18000	694	8634	7211	28.26%
	avrg		34.52%	18000	569	8622	11836	30.18%
	overall-avrg solved: 8/36		26.32%	14512.38	557	6948	10390	17.93%

Table A.24.: Computational results for the EE-Model with $\rho = 50$.

data-type	graph-type	n	root	CPU (sec)	Sub	LBF	OptCut	gap
shuffle	clustered	10	75.26%	18000	703	9460	25295	23.63%
		15	61.80%	18000	508	9345	18569	6.95%
		20	70.20%	18000	1109	10674	15149	53.70%
		50	80.48%	18000	996	7589	7876	62.93%
	avrg		71.94%	18000	829	9267	16722	36.80%
	random	10	44.51%	18000	733	9608	23518	26.61%
		15	59.57%	18000	972	9785	17432	26.66%
		20	71.33%	18000	926	8754	14724	16.63%
		50	88.17%	18000	1390	4751	6497	62.11%
	avrg		65.89%	18000	1005	8225	15543	33.00%
	mixed	10	31.93%	2.13	12	19	58	0.00%
		15	86.80%	18000	827	10965	17438	60.80%
		20	88.10%	18000	1078	11992	14832	76.61%
		50	86.96%	18000	841	8243	7870	70.47%
	avrg		73.45%	13500.30	690	7805	10050	51.97%
norm0.7	clustered	10	90.05%	18000	605	7341	15110	88.34%
		15	78.89%	18000	1047	6216	11878	75.55%
		20	88.50%	18000	1097	6356	9790	87.65%
		50	84.02%	18000	631	6070	6940	82.26%
	avrg		85.36%	18000	845	6496	10930	83.45%
	random	10	85.27%	18000	654	6098	13366	81.08%
		15	80.86%	18000	1113	5143	10483	78.44%
		20	83.56%	18000	1208	5900	9061	81.89%
		50	77.07%	18000	1227	3529	4700	76.46%
	avrg		81.69%	18000	1051	5168	9403	79.47%
	mixed	10	84.91%	18000	593	6629	14527	82.93%
		15	89.50%	18000	792	8057	11153	87.87%
		20	86.52%	18000	913	7493	10348	84.70%
		50	85.52%	18000	635	6659	6769	84.21%
	avrg		86.61%	18000	733	7210	10699	84.93%
uni70	clustered	10	89.74%	18000	610	8031	18356	86.85%
		15	76.61%	18000	1063	7043	14401	66.26%
		20	88.99%	18000	1144	7621	11609	87.43%
		50	84.68%	18000	810	7339	7443	81.61%
	avrg		85.00%	18000	907	7509	12952	80.54%
	random	10	81.24%	18000	719	7007	17943	75.63%
		15	69.22%	18000	1116	7589	13975	65.16%
		20	79.93%	18000	1307	7532	11246	78.27%
		50	80.34%	18000	1449	4578	6195	75.26%
	avrg		77.68%	18000	1148	6677	12340	73.58%
	mixed	10	65.24%	18000	642	8599	20003	58.17%
		15	88.51%	18000	816	10087	12045	86.42%
		20	87.83%	18000	939	8497	11307	87.06%
		50	82.45%	18000	823	7694	7132	81.95%
	avrg		81.01%	18000	805	8719	12622	78.40%
	overall-avrg solved: 1/36		78.74%	17500.06	890	7453	12362	66.90%

Table A.25.: Computational results for the EE-Model with $\rho = 100$.

data-type	graph-type	n	root	CPU (sec)	Sub	LBF	OptCut	gap
shuffle	clustered	10	85.73%	18000	679	7455	19342	41.46%
		15	76.21%	18000	625	7349	13927	16.58%
		20	82.33%	18000	1269	8265	12151	70.18%
		50	88.91%	18000	947	7017	7644	76.40%
	avrg		83.30%	18000	880	7522	13266	51.16%
	random	10	61.39%	18000	686	7744	17304	50.02%
		15	74.54%	18000	1191	7790	14420	45.44%
		20	83.12%	18000	1018	6580	11411	31.42%
		50	93.64%	18000	1389	4640	6114	75.01%
	avrg		78.17%	18000	1071	6689	12312	50.47%
	mixed	10	1.78%	1.20	8	25	32	0.00%
		15	92.86%	18000	908	9212	14895	75.68%
		20	93.59%	18000	997	9264	11694	86.64%
		50	92.77%	18000	958	7613	7347	84.25%
	avrg		70.25%	13500.30	718	6529	8492	61.65%
norm0.7	clustered	10	94.70%	18000	602	6863	13784	93.81%
		15	88.12%	18000	1134	5932	11114	86.01%
		20	93.81%	18000	1100	6908	10238	93.25%
		50	91.13%	18000	599	6227	6718	90.17%
	avrg		91.94%	18000	859	6483	10464	90.81%
	random	10	91.99%	18000	677	7498	14492	89.66%
		15	89.53%	18000	1086	5726	10320	87.76%
		20	90.87%	18000	1205	5533	8831	89.92%
		50	86.75%	18000	1244	3966	4951	86.23%
	avrg		89.78%	18000	1053	5681	9649	88.39%
	mixed	10	91.80%	18000	598	7268	15057	90.70%
		15	94.39%	18000	809	8292	11219	93.50%
		20	92.69%	18000	965	8774	12236	91.51%
		50	92.02%	18000	604	6429	6292	91.26%
	avrg		92.72%	18000	744	7691	11201	91.74%
uni70	clustered	10	94.32%	18000	627	8574	19322	92.96%
		15	86.63%	18000	1101	7640	14545	79.55%
		20	94.11%	18000	1168	7452	11227	93.16%
		50	91.41%	18000	801	7095	7376	89.69%
	avrg		91.62%	18000	924	7690	13118	88.84%
	random	10	89.55%	18000	701	8045	18784	86.40%
		15	81.69%	18000	1155	6796	14445	79.28%
		20	88.73%	18000	1340	7214	11101	87.72%
		50	88.50%	18000	1523	4349	6176	85.66%
	avrg		87.12%	18000	1180	6601	12627	84.77%
	mixed	10	78.90%	18000	593	8530	21026	74.23%
		15	93.88%	18000	824	9451	12203	92.70%
		20	93.42%	18000	1027	9057	11714	92.98%
		50	90.24%	18000	837	7584	7134	89.77%
	avrg		89.11%	18000	820	8656	13019	87.42%
	overall-avrg solved: 1/36		86.00%	17500.03	917	7060	11572	77.25%

Bibliography

[1] R. K. Ahuja, T. L. Magnanti, J. B. Orlin: *Network Flows*. Prentice-Hall, New Jersey, 1993.

[2] E. Angelelli, R. Mansini: *The Vehicle Routing Problem with Time Windows and Simultaneous Pickup and Delivery*. In: A. Klose, M. G. Speranza, L. N. Van Wassenhove (eds) Quantitative Approaches to Distribution Logistics and Supply Chain Management. Springer, Berlin-Heidelberg, pages 249-267, 2002.

[3] M. O. Ball, T. L. Magnanti, C. L. Monma, G. L. Nemhauser: *Network Routing*. Handbooks in Operations Research and Management Science, Vol. 8, 1995.

[4] J.E. Beasley: *Route-first cluster-second methods for vehicle routing*. Omega, Vol. 11, pages 403-408, 1983.

[5] D. Ben-Arieh, G. Gutin, M. Penn, A. Yeo, A. Zverovitch: *Transformation of generalized ATSP into ATSP*. Operations Research Letters, Vol. 31, pages 357-365, 2003.

[6] A. Ben-Tal, L. El-Ghaoui, A. Nemirovski: *Robust Optimization*. Princeton University Press, Princeton and Oxford, 2009.

[7] D.J. Bertsimas: *A Vehicle Routing Problem with Stochastic Demands*. Operations Research, Vol. 40, pages 574-585, 1992.

[8] N. Bianchessi, G. Righini: *Heuristic algorithms for the vehicle routing problem with simultaneous pick-up and delivery*. Computers and Operations Research, Vol. 34, pages 578-594, 2007.

[9] L. Bianchi, M. Birattari, M. Chiarandini, M. Manfrin, M. Mastrolilli, L. Paquete, O. Rossi-Doria, T. Schiavinotto: *Hybrid Metaheuristics for the Vehicle Routing Problem with Stochastic Demands*. Journal of Mathematical Modelling and Algorithms, Vol. 5, pages 91-110, 2006.

[10] J. Bramel, D. Simchi-Levi: *A location based heuristic for general routing problems*. Operations Research, Vol. 43, pages 649-660, 1995.

[11] C.C. Carøe, R. Schultz: *Dual decomposition in stochastic integer programming*. Operations Research Letters, Vol. 24, pages 37-45, 1999.

[12] C.H. Christiansen, J. Lysgaard: *A branch-and-price algorithm for the capacitated vehicle routing with stochastic demands*. Operations Research Letters, Vol. 35, pages 773-781, 2007.

[13] N. Christofides, A. Mingozzi, P.Toth: *The vehicle routing problem*. In N. Christofides, A. Mingozzi, P.Toth, and C. Sandi, editors, Combinatorial Optimization, Wiley, Chichester, pages 315-338, 1979.

[14] G. Clarke, J.V. Wright: *Scheduling of vehicles from a central depot to a number of delivery points*. Operations Research, Vol. 12, pages 568-581, 1964.

[15] J.-F. Cordeau, G. Laporte, M.W.P. Savelsbergh, D. Vigo: *Vehicle Routing*. In C. Barnhart and G. Laporte, editors, Handbook in OR & MS, Vol. 14, pages 367-417, Elsevier B.V., 2007.

[16] J.-F. Côté, M. Gendreau, J.-Y. Potvin: *An Exact Algorithm for the Two-Dimensional Orthogonal Packing Problem with Unloading Constraints*. Operations Research, Vol. 62, pages 1126-1141, 2014.

[17] G.B. Dantzig and J. H. Ramser: *The Truck Dispatching Problem*. Management Science, Vol. 6, pages 80-91, 1959.

[18] M. Dell'Amico, G. Righini, M. Salani: *A Branch-and-Price Approach to the Vehicle Routing Problem with Simultaneous Distribution and Collection*; Transportation Science Vol. 40, pages 235-247, 2006.

[19] G. Desaulniers, J. Desrosiers, M.M. Solomon: *Column Generation.* Springer Science+Business Media Inc., New York, 2005.

[20] J. Desrosiers, F. Soumis, M. Desrochers: *Routing with time windows by column generation.* Networks, Vol. 14, pages 545-565, 1984.

[21] D. Drapkin: *Models and Algorithms for Dominance-Constrained Stochastic Programs with Recourse.* Ph.D. Thesis, Faculty of Mathematics, University of Duisburg-Essen, 2014.

[22] M. Dror, G. Laporte, P. Trudeau: *Vehicle Routing with Stochastic Demands: Properties and Solution Frameworks.* Transportation Science, Vol. 23, pages 166-176, 1989.

[23] M. Dror, P. Trudeau: *Stochastic vehicle routing with modified savings algorithm.* European Journal of Operational Research, Vol. 23, pages 228-235, 1986.

[24] M. Fischetti, J.J. Salazar González, P. Toth: *A Branch-And-Cut Algorithm for the Symmetric Generalized Traveling Salesman Problem.* Operations Research, Vol. 45, pages 378-394, 1997.

[25] M. L. Fisher: *Optimal Solution of Vehicle Routing Problems Using Minimum K-Trees.* Operations Research, Vol. 42, pages 626-642, 1994.

[26] M.L. Fisher, R. Jaikumar: *A generalized assignment heuristic for the vehicle routing problem.* Networks, Vol. 11, pages 109-124, 1981.

[27] T.J. Gaskell: *Bases for vehicle fleet scheduling.* Operational Research Quaterly, Vol. 18, pages 281-295, 1967.

[28] M. Gendreau, G. Laporte, R. Séguin: *Stochastic Vehicle Routing.* European Journal of Operations Research, Vol. 88, pages 3-12, 1996.

[29] M. Gendreau, G. Laporte, R. Séguin: *A Tabu Search Heuristic for the Vehicle Routing Problem with Stochastic Demands and Customers.* Operations Research, Vol. 44, pages 469-477, 1996.

[30] M. Gendreau, J.-Y. Potvin, O. Bräysy, G. Hasle, A. Løkketangen: *Metaheuristics for the Vehicle Routing Problem and its Extensions: A Categorized Bibliography.* CIRRELT-2007-27, August 2007.

[31] B.E. Gillet, L.R. Miller: *A heuristic algorithm for the vehicle dispatch problem.* Operations Research, Vol. 22, pages 340-349, 1974.

[32] G. Gutin, D. Karapetyan: *Generalized Traveling Salesman Problem Reduction Algorithms.* Algorithmic Operations Research, Vol. 4, pages 144-154, 2009.

[33] G. Gutin, D. Karapetyan: *Efficient local search algorithms for known and new neighborhoods for the generalized traveling salesman problem.* European Journal of Operational Research, Vol. 219(2), pages 234-251, 2012.

[34] C. Hjorring, J. Holt: *New optimality cuts for a single-vehicle stochastic routing problem.* Annals of Operations Research, Vol. 86, pages 569-584, 1999.

[35] L. Hou, H. Zhou: *Stochastic Vehicle Routing Problem with Uncertain Demand and Travel Time and Simultaneous Pickups and Deliveries.* Third International Joint Confenrece on Computational Science and Optimization, 2010.

[36] O. Jabali, W. Rei, M. Gendreau, G. Laporte: *Partial-Route Inequalities for the Multi-Vehicle Routing Problem with Stochastic Demands.* Discrete Applied Mathematics, Vol. 177, pages 121-136, October 2014.

[37] G.A.P. Kindervater, M.W.P. Savelsbergh: *Vehicle Routing: Handling edge exchanges.* In E.H.L. Aarts and J.K. Lenstra, editors, Local Search in Combinatorial Optimization, Wiley, Chichester, pages 337-360, 1997.

[38] G. Laporte: *The vehicle routing problem: an overview of exact and approximate algorithms.* European Journal of Operational Research, Vol. 59, pages 345-358, 1992.

[39] G. Laporte: *Vehicle routing.* In: Dell'Amico, Maffioli, Martello (Eds.) Annotated Bibliographies in Combinatorial Optimization. New York, Wiley, 1997.

[40] G. Laporte, F.V. Louveaux: *The integer L-shaped method for stochastic integer programs with complete recourse.* Operations Research Letters, Vol. 13, pages 133-142, 1993.

[41] G. Laporte, F.V. Louveaux, H. Mercure: *Models and exact solutions for a class of stochastic location-routing problems.* European Journal of Operational Research, Vol. 39, pages 71-78, 1989.

[42] G. Laporte, F.V. Louveaux, L. van Hamme: *An Integer L-Shaped Algorithm for the Capacitated Vehicle Routing Problem with Stochastic Demands.* Operations Research, Vol. 50, pages 415-423, 2002.

[43] G. Laporte, F. Semet: *Computational evaluation of transformation procedure for the symmetric generalized traveling salesman problem.* INFOR, Vol. 37, pages 114-120, 1999.

[44] Y.N. Lien, E. Ma, B.W.-S. Wah: *Transformation of the Generalized Traveling-Salesman Problem into Standard Traveling-Salesman Problem.* Information Science, Vol. 74, pages 177-189, 1993.

[45] S. Lin: *Computer solutions of the traveling salesman problem.* Bell System Technical Journal, Vol. 44, pages 2245-2269, 1965.

[46] S. Lin, B. W. Kernighan: *An effective heuristic algorithm for the traveling salesman problem.* Operations Research, Vol. 21, pages 498-516, 1973.

[47] F.V. Louveaux: *An Introduction to Stochastic Transportation Models.* In M. Labbé, G. Lapporte, K. Tanczos, and P. Toint, editors, *Operations Research and Decision Aid Methodologies in Traffic and Transportation Management*, pages 244-263. NATO ASI Series F: Computer and Systems Sciences, Springer-Verlag, Berlin and Heidelberg, 1998.

[48] J. Lysgaard, A.N. Letchford, R.W. Eglese: *A new branch-and-cut algorithm for the capacitated vehicle routing problem.* Mathematical Programming, Vol. 100(2), pages 423-445, 2004.

[49] A. Märkert, R. Gollmer: *DDSIP - Duale Dekomposition in der zweistufigen stochastischen gemischt-ganzzahligen Optimierung.* Homepage: `https://www.uni-due.de/~hn215go/ddsip.shtml`, April 14, 2014.

[50] I. Minis, A. Tatarakis: *Stochastic single vehicle routing problem with delivery and pick up and a predefined customer sequence.* European Journal of Operational Research, Vol. 213, pages 37-51, 2011.

[51] R.H. Mole, S.R. Jameson: *A sequential route-building algorithm for the traveling salesman problem.* Operational Research Quaterly, Vol. 27, pages 503-511, 1976.

[52] A. Mueller, D. Stoyan: *Comparison Methods for Stochastic Models and Risks.* Wiley, Chichester, 2002.

[53] C.E. Noon, J.C. Bean: *An Efficient Transformation of the Generalized Traveling Salesman Problem.* INFOR, Vol. 31, pages 39-44, 1993.

[54] I. Or: *Traveling salesman-type combinatorial optimization problems and their relation to the logistics of regional blood banking.* Ph.D. dissertation, Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, IL, 1976.

[55] I. H. Osman: *Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem.* Annals of Operations Research, Vol. 41, pages 421-451, 1993.

[56] I. H. Osman, N. Christofides: *Simulated annealing and descent algorithms for capacitated clustering problems.* EURO-XI, Beograd, 1989.

[57] S. N. Parragh, K. F. Doerner, R. F. Hartl: *A survey on pickup and delivery problems; Part I: Transportation between customers and depot.* Institute of Business Administration, Februray, Vienna, 2008.

[58] G. C. Pflug: *Some remarks on the value-at-risk and the conditional value-at-risk*. S. P. Uryasev, ed. Probabilistic Constrained Optimization: Methodology and Applications. Kluwer, Norwell, MA, pages 278-287, 2000.

[59] J. Renaud, F.F. Boctor, G. Laporte: *A fast composite heuristic for the symmetric traveling salesman problem*. INFORMS Journal on Computing, Vol. 8, pages 158-164, 1988.

[60] R. T. Rockafellar and S. P. Uryasev: *Conditional value-at-risk for general loss distributions*. Journal of Banking and Finance, Vol. 26, pages 1443-1471, 2002.

[61] A. Ruszczyński: *A regularized decomposition method for minimizing a sum of polyhedral functions*. Mathematical Programming 35, pages 309-333, 1986.

[62] A. Ruszczyński, A. Shapiro (Eds.): *Handbooks in Operations Research and Management Science, 10: Stochastic Programming*. Elsevier, Amsterdam, 2003.

[63] A. Schrijver: *Theory of Linear and Integer Programming*. Wiley-Interscience Publication, John Wiley & Sons Ltd., Essex, 1986.

[64] R. Séguin: *Problèm stochastiques de tournées de véhicules*. Ph.D. Thesis, Département d'informatique et de recherche opérationelle, Publication CRT-979, Centre de recherche sur les transports, Université de Montréal, 1994.

[65] A. Shapiro, D. Dentcheva, A. Ruszczyński: *Lectures on Stochastic Programming: Modeling and Theory*. MPS-SIAM Series on Optimization. SIAM, Philadelphia, 2009.

[66] J. Silberholz, B. Golden: *The Generalized Traveling Salesman Problem: a new Genetic Algorithm approach*. Extending the Horizons: Advances in Computing, Optimization, and Decision Technologies, pages 165-181, 2007.

[67] L. V. Snyder, and M. S. Daskin: *A random-key genetic algorithm for the generalized traveling salesman problem*. European Journal of Operational Research, Vol. 174, pages. 38-53, 2006.

[68] M. M. Solomon : *Vehicle Routing with Time Windows Benchmark Problems*. Homepage: `http://web.cba.neu.edu/~msolomon/problems.htm`, March 24, 2005.

[69] W. Stewart, B. Golden: *A Lagrangean relaxation heuristic for vehicle routing*. European Journal of Operational Research, Vol. 15, S. 84-88, 1984.

[70] W. Stewart, B. Golden: *Stochastic vehicle routing: A comprehensive approach*. European Journal of Operational Research, Vol. 14, pages 371-385, 1983.

[71] P.M. Thompson, H.N. Psaraftis: *Cyclic transfer algorithms for multivehicle routing and scheduling problems*. Operations Research, Vol. 41, pages 935-946, 1993.

[72] P. Toth, D. Vigo: *The Vehicle Routing Problem*. SIAM Monographs on Discrete Mathematics and Applications, Philadelphia, 1-26, 2002.

[73] A. Van Breedam: *An analysis of the behavior of heuristics for the vehicle routing problem for a selection of problems with vehicle-related, customer-related, and time-related constraints*. Ph.D. dissertation, University of Antwerp, 1994.

[74] R. Van Slyke, R.J.-B. Wets: *L-shaped linear programs with applications to optimal control and stochastic programming*. SIAM J. Appl. Math., Vol. 17, pages 638-663, 1969.

[75] A. Wren: *Computers in Transport Planning and Operation*. Ian Allan, London, 1971.

[76] A. Wren, A. Holliday: *Computer scheduling of vehicles from one or more depots to a number of delivery points*. Operational Research Quaterly, Vol. 23, pages 333-344, 1972.

[77] P. Yellow: *A computational modification to the savings method of vehicle scheduling*. Operational Research Quaterly, Vol. 21, pages 281-283, 1970.