Robust optimization for airline scheduling and vehicle routing

by

Da Lu

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Management Sciences

Waterloo, Ontario, Canada, 2014

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Robust optimization is an emerging modeling approach to make decisions under uncertainty. It provides an alternative framework to stochastic optimization where operational parameters are random and do not assume any probability distribution. In this thesis, we study three important problems in routing and scheduling under uncertainty, namely, the crew pairing problem, the shortest path problem with resource constraints, and the vehicle routing problem with time windows. We present robust optimization models and propose novel solution approaches, and perform extensive numerical testing to validate the models and solutions.

The crew pairing problem finds a set of legal pairings with minimum cost to cover a set of flights. An optimal solution for the deterministic case, however, is often found to be far from optimal or even infeasible when implemented due to the several uncertainties inherent to the airline industry. We present a robust crew pairing formulation where time between flights may vary within an interval. The robust model determines a solution that minimizes crew cost and provides protection against disruptions with a specified level. A column generation approach is presented to solve the robust crew pairing problem. The robust model and the solution approach are tested on a set of instances based on an European airline. The solutions are more robust than the deterministic ones under simulated disruptions.

The shortest path problem with resource constraints (SPPRC) is an important problem that appears as a subproblem in many routing and scheduling problems. The second study in the thesis focuses on the robust SPPRC where both cost and resource consumptions are random. The robust SPPRC determines a minimum cost path that is feasible when a number of variations occur for each resource. We present a mixed-integer programming (MIP) model that is equivalent to the robust SPPRC model, and develop graph reduction techniques and two solution methods. The first solution method is a sequential algorithm that solves a series of deterministic SPPRC. The second is a modified label-setting algorithm that uses a new dominance rule. Numerical testing shows that the modified label-setting algorithm outperforms the sequential algorithm and the MIP model.

The third problem studied is the vehicle routing problem with time windows under uncertain customer demands. The robust model determines a set of routes with minimum cost such that each customer is served exactly once within the time window and each route is feasible when a number of customers change their demands. We propose a branch-and-price-and-cut algorithm and a novel separation strategy to determine valid inequalities that make use of data uncertainty. The model and solution methodology are tested on instances generated based on the Solomon instances. The robust solutions pro-

vide significant protection against random changes in customer demands compared to the deterministic solutions.

Acknowledgements

I would like to thank all the people who made this possible.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1. Airline planning and vehicle routing

The airline industry is a highly competitive business that is characterized by large fixed costs and low profit margins [eco]. Decision makers need to make the best use of resources, such as aircraft, personnel and airport slots, to survive the high competition. The cost for personnel represents the second largest cost for airlines [Cohn and Lapp, 2010]. Therefore, airlines are continuously seeking to increase the utilization of personnel while maintaining high standards for safety and welfare. Operations research has a long history of successfully helping airlines to determine legal working schedules for personnel at minimum cost under a deterministic setting. However, working schedules are rarely implemented as planned due to a variety of disruptions, such as bad weather, mechanical breakdown, flight delay and cancellations, and labor and passenger issues, which results in a cost increase that ranges from 2% to 3% of annual revenue [Sab]. Therefore, the challenge of handling uncertainty is revealing new frontiers for researchers in this area.

Another optimization problem that is closely related to the airline industry is vehicle routing. The problem can be generically described as serving a set of customers by a fleet of vehicles, and has applications in many service industries, such as maintenance, courier, consumer products, and transportation. There exist many variants that consider different restrictions and assumptions inspired from the real world, such as vehicle capacity, service time windows, backhauls, and multiple tasks with pickup and delivery. However, due to the variability of model input, such as increased travel time due to congestion, customer demand changes, and increased service times, the uncertainties translate into increased operational cost and decreased customer satisfaction. Therefore, both practitioners and researchers face the challenge of developing routing and scheduling approaches that protect against uncertainty in operations.

Uncertainty presents a hurdle to the application of optimization tools to real world problems. The variation in operational parameters can severely degrade solution quality, making the proposed solutions being far from optimal or even infeasible when implemented. Many studies show that the decisions made under a deterministic setting are sensitive to the changes that may occur during execution [Birge and Louveaux, 2011; Herroelen and Leus, 2005; Sahinidis, 2004].

Optimization offers two frameworks to model uncertainty: stochastic optimization and robust optimization. In this thesis, we use robust optimization to model three problems in routing and scheduling, namely the crew pairing problem, shortest path problem with resource constraints, and vehicle routing problem with time windows. We first introduce stochastic and robust optimization frameworks, then detail the contributions and organization of the thesis.

1.2. Stochastic optimization

Stochastic optimization assumes that the uncertain parameters follow a probability distribution that is known or that can be estimated. Stochastic optimization looks for a solution that is feasible under all possible parameter realizations and performs best on average. Two basic types of models in stochastic optimization are recourse models and chance-constrained models. The decision making process in recourse models is divided into two or more stages where the decision at each stage is determined based on the information available up to the current stage. These decisions jointly optimize an objective that evaluates the quality of the solution. For example, a two-stage linear programming recourse model takes the following form:

$$\begin{aligned} \min \quad & \boldsymbol{c}^\top \boldsymbol{x} + \mathbb{E}\left[\boldsymbol{Q}(\boldsymbol{x}, \boldsymbol{\xi})\right] \\ \text{s.t.} \quad & \boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}, \\ & \boldsymbol{x} \geq \mathbf{o}, \end{aligned} \tag{1.1}$$

where $\boldsymbol{Q}(\boldsymbol{x}, \boldsymbol{\xi})$ is optimized by the second-stage problem:

$$\begin{aligned} \min_{\boldsymbol{y}} \quad & \boldsymbol{q}^\top \boldsymbol{y}, \\ \text{s.t.} \quad & \boldsymbol{T}\boldsymbol{x} + \boldsymbol{W}\boldsymbol{y} \leq \boldsymbol{h}, \end{aligned} \tag{1.2}$$

where $\boldsymbol{x} \in \mathbb{R}^m$ is the first-stage decision vector determined before the uncertain parameter vector $\boldsymbol{\xi} = (\boldsymbol{q}, \boldsymbol{T}, \boldsymbol{W}, \boldsymbol{h})$ realizes, and $\boldsymbol{y} \in \mathbb{R}^n$ is the second-stage decision vector made after $\boldsymbol{\xi}$ becomes available. $\boldsymbol{c}^\top \boldsymbol{x} + \mathbb{E}\left[\boldsymbol{Q}(\boldsymbol{x}, \boldsymbol{\xi})\right]$ optimizes the expected objective value by taking the advantage of the probabilistic property of $\boldsymbol{\xi}$. The model first determines a here-and-now decision at the first stage before the uncertain parameters are realized. In the second stage, when parameters become known, it provides a recourse decision accordingly.

The chance-constrained model generates a here-and-now decision that satisfies all constraints with a pre-specified probability. The interested readers are referred to Birge and Louveaux [2011], Shapiro et al. [2009] and Shapiro and Philpot. A chance constraint is represented by

$$\mathbf{P}\{\boldsymbol{A}(w)\boldsymbol{x} \leq \boldsymbol{b}(w)\} \geq \alpha, \tag{1.3}$$

where $\boldsymbol{x}$ is the decision variable vector, $\boldsymbol{A}(w) \in \mathbb{R}^{n \times m}$ and $\boldsymbol{b}(w) \in \mathbb{R}^n$ are stochastic parameters whose values depend on the realization w, and α is the confidence level that specifies the probability of $\boldsymbol{A}(w)\boldsymbol{x} \leq \boldsymbol{b}(w)$.

1.3. Robust optimization

Robust optimization originates from a different modeling paradigm. Instead of assuming a certain knowledge of the probabilistic attributes of parameters, it only requires the set of all possible parameter realizations, which is referred to as the support set of the parameters. Robust optimization selects an optimal decision that is feasible under all possible realizations. The generic form of a robust optimization problem is:

$$\begin{aligned} \min \quad & f_0(\boldsymbol{x}) \\ \text{s.t.} \quad & f_i(\boldsymbol{x}, \boldsymbol{a_i}) \leq 0, \quad \forall \boldsymbol{a_i} \in \mathcal{U}_i, i = 1, ..., n, \end{aligned} \tag{1.4}$$

where $\boldsymbol{x} \in \mathbb{R}^m$ is the decision variable vector, $\boldsymbol{a_i} \in \mathbb{R}^m$ is a realization of the uncertain parameters within support $\mathcal{U}_i \subseteq \mathbb{R}^m$ for constraint i, and f_0 and f_i are functions that describe the objective function and constraint $i, i = 1, ..., n$. Problem (1.4) is called the robust counterpart of the deterministic problem.

The four basic types of support studied in the literature are ellipsoidal, polyhedral, cardinality constrained and norm uncertainties. A generic form of robust linear optimization model is:

$$\begin{aligned} \min \quad & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{s.t.} \quad & \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}, \quad \forall \boldsymbol{a_1} \in \mathcal{U}_1, ..., \boldsymbol{a_n} \in \mathcal{U}_n, \end{aligned} \tag{1.5}$$

where $A \in \mathbb{R}^{n \times m}$ is a uncertain matrix, where the ith row is $\boldsymbol{a}_i, i = 1, ..., n$. Next, we describe the four types of uncertainty support and identify the form of their equivalent formulations.

Polyhedral uncertainty

Polyhedral uncertainty is described by

$$\mathcal{U} = \prod_{i=1}^{n} \left(\mathcal{U}_i = \{\boldsymbol{a_i} | \boldsymbol{D_i a_i} \leq \boldsymbol{d_i}\} \right), \tag{1.6}$$

where n convex hulls jointly encode the support. Problem (1.5) with polyhedral uncertainty can be reformulated as a linear program [Bertsimas et al., 2011].

Ellipsoidal uncertainty

The ellipsoidal uncertainty generalizes the polyhedral uncertainty and is able to well approximate many complicated convex sets. Let an ellipsoidal structure $\mathcal{U}$ in $\mathbb{R}^K$ be defined as

$$U = \{\Pi(\boldsymbol{u}) | \|\boldsymbol{Q_e u}\| \leq 1\}, \tag{1.7}$$

where $\boldsymbol{u} \rightarrow \Pi(\boldsymbol{u})$ is an affine embedding of certain $\mathbb{R}^L$ into $\mathbb{R}^k$ and $\boldsymbol{Q_e}$ is an $M \times L$ matrix [Ben-Tal and Nemirovski, 1999]. An ellipsoidal uncertainty is described by the intersection of a finite number of ellipsoidal structures:

$$\mathcal{U} = \bigcap_{e=0}^{k} U(\Pi_e, \boldsymbol{Q_e}) \tag{1.8}$$

with the additional requirement that $\mathcal{U}$ is bounded and there is at least one matrix $\boldsymbol{A} \in \mathcal{U}$ which belongs to the "relative interior" of every $U(\Pi_e, Q_e), e = 1, ...k$. The resulting Problem (1.5) with ellipsoidal uncertainty can be equivalently reformulated as a conic quadratic program [Ben-Tal and Nemirovski, 1999].

Norm uncertainty

Bertsimas et al. [2004] generalize the ellipsoidal uncertainty to any arbitrary norm p. Particularly, the uncertainty is defined as

$$\mathcal{U} = \{\boldsymbol{A} | \|\boldsymbol{M}(\text{vec}(\boldsymbol{A}) - \text{vec}(\bar{\boldsymbol{A}}))\|_p \leq \Delta\} \tag{1.9}$$

where $\text{vec}(\boldsymbol{A})$ is a vector constructed by concatenating the rows of $\boldsymbol{A}$, matrix $\boldsymbol{M}$ is invertible, and $\bar{\boldsymbol{A}}$ is any constant matrix. Bertsimas et al. [2004] show that when $p = 1$ or ∞, Problem (1.5) with norm uncertainty can be reformulated as a linear program.

Cardinality constrained uncertainty

Bertsimas and Sim [2003] propose a support structure that is described by a family of hyperrectangles with a fixed number of dimensions. To be specific, let a_{ij} be the jth parameter of constraint i. Parameter a_{ij} varies in interval $\left[\underline{a}_{ij}, \underline{a}_{ij} + h_{ij}\right]$ where $\underline{a}_{ij}$ is the nominal value of a_{ij} and h_{ij} is the maximum deviation from the nominal value. The cardinality constrained uncertainty is defined as

$$\mathcal{U} = \prod_{i=1}^{n} \left(\mathcal{U}_i = \bigcup_{S \in \{S_i | S_i \subseteq \{1,\ldots,m\}, |S_i| \leq \Gamma_i\}} \left\{ u_i \;\middle|\; \begin{array}{l} a_{ij} \in \left[\underline{a}_{ij}, \underline{a}_{ij} + h_{ij}\right], \forall j \in S, \\ a_{ij} = \underline{a}_{ij}, \forall j \notin S \end{array} \right\} \right). \tag{1.10}$$

Problem (1.5) with cardinality constrained uncertainty has an equivalent linear program [Bertsimas and Sim, 2003].

Compared to the other uncertainty structures, the cardinality constrained uncertainty provides a simple way to characterize the uncertainty support when there is not enough information for an accurate description on the structure. The decision maker controls the level of robustness by choosing appropriate cardinalities. The resulting model can provide a solution with an acceptable trade-off between optimality and robustness. In this thesis, we develop models based on cardinality constrained uncertainty.

1.4. Summary and Contributions of the thesis

The thesis studies three important problems in routing and scheduling, i.e. the crew pairing problem, the shortest path problem with resource constraints, and the vehicle routing problem with time windows. The focus is on modeling uncertainty, and building effective solution methodologies based on robust optimization with cardinality constrained uncertainty.

The thesis is organized as follows. Chapter 2 studies the robust crew pairing problem under cardinality constrained uncertainty. In contrast to the deterministic crew pairing problem, uncertainty in flight delays is modelled by allowing the time between two flight departures to vary in an interval. The model permits the decision maker to control the level of robustness by specifying a maximum number of disruptions allowed in a pairing. We propose a solution methodology based on column generation where the subproblem is a robust shortest path problem with a resource constraint. Unlike the robust combinatorial optimization problem in Bertsimas and Sim [2003] where only objective coefficients are uncertain, both objective and resource constraint coefficients of the subproblem are uncertain. We prove that the sequential approach proposed by Bertsimas and Sim [2003] that generates and solves a series of deterministic problems may be extended to solve the robust shortest path problem with a resource constraint. The robust crew pairing model and methodology are tested on a set of instances generated from a real problem and the solutions are compared against the deterministic solutions under simulated disruptions. The results show that the proposed solution approach successfully solves the instances within reasonable time and that the solutions are more robust than the deterministic solutions.

In Chapter 3, the robust shortest path problem with multiple resource constraints is studied. The resulting robust model is more challenging to solve and the sequential approach becomes computationally expensive for large size instances. To address the challenge, we develop a set of graph reduction techniques and propose a new dominance rule that exploits robust information. The solution method is tested on a set of instances and its effectiveness is supported by the results.

Chapter 4 studies the vehicle routing problem with time windows under customer demand uncertainty. The demand uncertainty is described by the cardinality constrained support, which is more challenging than the polyhedral support studied in Gounaris et al. [2013]. We propose a branch-and-price-and-cut algorithm where a novel separation strategy for valid cutting planes is used. The robust model and solution approach are tested on a set of instances derived from the Solomon instances. The robust solutions show superior protection against uncertainty, compared to the deterministic solutions under simulated scenarios.

1.5. Structure of the thesis

This thesis is organized as follows. In Chapter 2, we propose the formulation and solution methodology for the robust crew pairing problem based on cardinality constrained uncertainty. Chapter 3 generalizes the robust shortest path problem with single resource constraint to the case with multiple resource constraints, and presents a set of graph reduction procedures and a new dominance rule. Chapter 4 proposes a robust vehicle routing problem with time windows where customer demands are uncertain. A branch-and-price-and-cut algorithm and a novel cutting-plane separation procedure are developed to solve the problem. Chapter 5 concludes the thesis and discusses future research directions.

Chapter 2

The robust crew pairing problem

This chapter is based on Lu and Gzara [2014b], published online, Journal of Global Optimization.

2.1. Motivation and objective

Planning airline operations is a complex task that involves flight schedule design, aircraft assignment, aircraft routing, and crew scheduling. The complexity stems from the large number of decisions, strict safety rules set by aviation regulating bodies, and a highly uncertain operating environment. In real-time operations, an aircraft may malfunction, a crew member may call-in sick, a passenger may show up late, or the weather conditions may be too dangerous to fly. For example the volcano ash ejected from the eruption of Eyjafjallajökull paralyzed the aviation system of Europe in 2010, affecting thousands of flights. The freezing rain in January 2014 in Toronto, Canada, led to hundreds of delays and cancellations. Any of these factors and others will cause a flight delay or cancellation, and successive flights on the schedule are likely to be affected. Between August 2012 and January 2013, about 19% of flights were delayed according to 16 air carriers reporting to the Bureau of Transportation Statistics. Among these, 6.5% are due to late aircraft arrivals. Delays in domestic passenger flights in the U.S.A. cost the U.S. economy $31.2 billion in 2007, including $8.3 billion in direct costs to airlines, $16.7 billion in indirect costs to passengers, $200 million in lost demand, and $4.0 billion in forgone GDP [Ball et al., 2010]. In 2011, 103 million minutes of delay caused by the U.S. national aviation control system were reported by Airlines for America (A4A) for U.S. passenger airlines over 77 U.S. airports. This delay figure is estimated to incur a cost of $7.732 billion to the airlines with $1.636 billion as crew pay [A4A].

Operations Research has made significant contributions to the airline industry through modeling and solving aircraft and crew scheduling problems. Models developed in the last 50 years led to better planning and huge savings for the industry [Clarke and Ryan, 2001]. While most literature assumes a deterministic context with known and fixed flight and connection times with no account of delay, numerous studies show that schedules made under such assumptions are sensitive to disruptions that may occur during execution [Birge and Louveaux, 2011; Herroelen and Leus, 2005; Sahinidis, 2004]. In particular, crew schedules are affected because of the strict regulations related to crew safety, fatigue, labor agreements, etc. Disrupted crew schedules are more costly for airlines than planned since crews are paid for the maximum of the scheduled and the actual workloads. Furthermore, a crew may no longer fly after reaching the maximum allowable workload, and a stand-by crew must be called in to cover the rest of the schedule. The additional cost of these irregular operations may lead to about 5% increase in the operational crew cost compared to the cost in the planning stage [Shebalov and Klabjan, 2006]. In addition, crew cost is the second largest expense for airlines after fuel cost [Yen and Birge, 2000]. Improving crew scheduling by building schedules that are less susceptible to delays and disruptions may potentially result in significant savings.

As will be defined and discussed in the next section, a core problem solved when building crew schedules is the crew pairing problem. In this chapter we focus on the crew pairing problem under an uncertain disruption setting. We aim at building robust schedules that will remain feasible and will minimize the cost of disruption under certain delay patterns. The work makes three significant contributions. To the best of our knowledge, we present the first robust crew pairing formulation with cardinality constrained uncertainty. In this framework, uncertain parameters vary randomly within an interval. The robust model minimizes the nominal crew cost and the additional cost due to delay and ensures that selected pairings remain feasible after a disruption occurs. The robustness of the solutions is controlled by a user-defined parameter referred to as protection level that limits the number of delays. Since data uncertainty is modeled by intervals, the distribution of delay is not required. Instead, the planner sets the range for delay. The robust formulation has nonlinear terms in the objective function and in the resource constraint and cannot be solved directly. We develop a solution methodology based on decomposition to isolate the nonlinear terms in the subproblem. We propose for the first time the robust shortest path problem with a resource constraint as a subproblem. We then prove that the nonlinear subproblem may be solved as a series of deterministic auxiliary optimization problems. The solution methodology for the subproblem extends the results of Bertsimas and Sim

[2003] on the shortest path problem to the more difficult resource constrained case with robustness in the objective function and in the constraint. Finally, we perform extensive testing and simulation using random and real instances to conclude that the proposed model and solution methodology solve in competitive time and lead to more robust crew pairings.

The chapter is organized as follows. Section 2.2 introduces airline planning in general. Then we narrow the scope of the discussion down to crew scheduling. Section 2.3 provides a review of the literature on the crew pairing problem under uncertainty. Section 2.4 introduces a robust formulation of the crew pairing problem with random delays. The solution process based on Lagrangian relaxation and Dantzig-wolfe decomposition is detailed in Section 2.5. The robust shortest path problem with a resource constraint is discussed in Section 2.6. Section 2.7 reports on the computational testing on real data from a European airline and on simulation results devised to compare the robust and deterministic solutions. The conclusion is presented in Section 2.8.

2.2. Introduction to airline planning and crew pairing

This section first provides a general introduction to the planning activities involved in airline operations. Then the crew pairing problem is defined and the modeling and solution techniques are discussed.

2.2.1 Airline planning

Airline planning is composed of several stages of decision making as shown in Figure 2.1. The decision making problems at each stage are complex and present a challenge to solve due to the large-scale and the several restrictions and requirements that need to be taken into account. Therefore, these stages are dealt with sequentially despite being dependent. The first stage deals with schedule design. The goal is to construct a cyclic daily or weekly schedule of flights to be flown repeatedly for a given planning period, usually one month. Inputs to flight schedules include forecasts of potential resources (e.g. fleet capacity, assigned time slots at airports, etc.), forecasts of market demand, competitors' service situation, and the desired market initiatives. The output is a generic service plan with no specific aircraft or crew assignment. The second stage solves the fleet assignment problem which determines what type of aircraft is used for each flight. The inputs in this stage are flight schedules, number of aircrafts by fleet type, turn around time (i.e. the amount of time an aircraft spends on ground between flights) by fleet type at each airport, operating cost and potential revenue of flights by fleet type. The output is an assignment of fleet types to scheduled flights such that profit is maximized, every flight is covered by exactly one fleet type, and total aircraft resources are not exceeded. The next stage is the maintenance routing or aircraft routing problem, in which each aircraft is assigned to a sequence of flights so that maintenance checks can be adequately and regularly carried out at specific airports depending on the fleet type. Such maintenance is necessary for a safe and high utilization of aircrafts. The inputs and outputs for each stage are summarized in Table 2.1. After the three stages are completed, crew scheduling is then addressed.

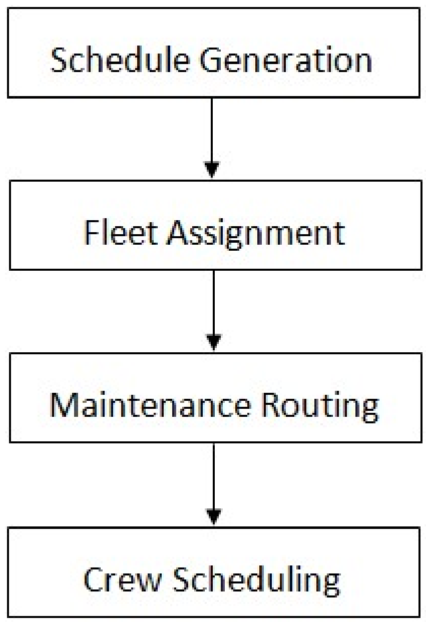

Figure 2.1: Airline planning.

Table 2.1: The inputs and outputs for each airline planning stage.

Planning stage	Inputs	Outputs
Schedule generation	Forecasts of potential resources, market demand, competitors' service situation, and desired market initiatives	A generic flight schedule
Fleet assignment	Flight schedule, number of aircrafts by fleet type, turn-around time by fleet type at each airport, operating cost and potential revenue of flights by fleet type	An assignment of fleet types to scheduled flights
Maintenance routing	Flight schedule with fleet type, the status of each aircraft, turn-around time by fleet type at each airport, regulation rules	An assignment of tasks for each aircraft
Crew scheduling	Flight schedule with fleet type, number of crew members required by fleet type, total number of crew members, status of each crew member, regulation rules, crew cost structure	An assignment of tasks for each crew member

2.2.2 Crew pairing

Crew schedules are built by first forming crew pairings as legal sequences of duties. A duty is a sequence of flight legs that spans a work day; and a pairing is a sequence of flight legs that spans multiple days with legal rules enforced and is assigned to one crew. The crew pairing problem constructs a set of pairings such that each scheduled flight is covered exactly once at minimum cost. The crew pairing problem is important as it possesses properties that are common to all crew scheduling problems. Moreover, it is one of the most complex scheduling problems due to its complicated cost structure and the numerous restrictions imposed.

Pairing cost and work rules

There are a number of rules restricting the set of feasible duties and pairings. These rules usually come from three sources. First, governing agencies issue regulations for safety purposes, such as the maximum flying time in a pairing. Second, labor unions often impose requirements concerning work conditions, such as safety and service requirements on accommodation and ground transportation. Third, airlines have additional constraints such as fair balancing of workload among different crew bases. All of these constraints lead to a huge computational challenge. For example, the restrictions imposed on duty include minimum and maximum sit time between sequential flights, maximum elapsed time and flying time in a duty. Rules imposed on pairing include maximum number of duties in a pairing, minimum and maximum amount of rest between duties, and the maximum time away from crew base [Barnhart et al., 2003].

The crew cost for a duty is usually the maximum of three quantities, i.e. the flying time, a fraction of the total elapsed time, and the minimum guaranteed number of hours. With this payment structure, crews are protected against very short duties or duties with extensive idle time between flights. The cost of a pairing comprises of two parts. The first part is the maximum of three quantities, that are the sum of the costs of all duties involved in the pairing, a fraction of the total elapsed time of the pairing, and a minimum guaranteed time per pairing. The second part of a pairing cost consists of extra costs such as meals and accommodation.

Formulation and solution methodology

The classical formulation for the crew pairing problem is a set-partitioning type where the set of flights in $\mathcal{F}$ are to be covered by pairings in $\mathcal{P}$. Each pairing $p \in \mathcal{P}$ has a cost c_p, and $a_{ip} = 1$ if flight $i \in \mathcal{F}$ is covered by pairing $p \in \mathcal{P}$ and 0 otherwise. The decision variable x_p equals 1 if pairing $p \in \mathcal{P}$ is selected, and 0 otherwise. The objective for the crew pairing problem is to minimize the total cost of selected pairings such that each flight is covered by a pairing, resulting in the following set partitioning formulation:

$$\text{[P]:} \quad \min \sum_{p \in \mathcal{P}} c_p x_p \tag{2.1}$$

$$\text{s.t.} \quad \sum_{p \in \mathcal{P}} a_{ip} x_p = 1 \quad i \in \mathcal{F} \tag{2.2}$$

$$x_p \in \{0, 1\} \qquad p \in \mathcal{P} \tag{2.3}$$

In practice, it is prohibitive to enumerate the set of feasible pairings $\mathcal{P}$. Instead, the set-partitioning formulation is solved implicitly by branch-and-price. At each node of the branch-and-bound tree, the LP relaxation of problem [P] defined over a subset of the pairings is solved iteratively until no promising pairings are generated [Barnhart et al., 2003; Desaulniers et al., 1997, 1998; Desrosiers et al., 1995]. The subproblem is a shortest path problem with resource constraints (SPPRC), which can be solved by a label-setting algorithm. The labels of a node are cost-resource vectors. The cost-resource vector records those nonlinear costs and resources that can be built into the network, e.g. the total flying time, the number of duties in a pairing, etc. In the case of multi-resource or nonlinear cost, the solution methodology of SPPRC needs to keep all nondominated paths to intermediate nodes. A detailed discussion of SPPRC is found in Desrosiers et al. [1995],Desaulniers et al. [1997] and Desaulniers et al. [1998].

To find the optimal integer solution, the column generation algorithm is embedded in a branch-and-bound scheme, i.e. commonly known as the branch-and-price algorithm. Details of applying branch-and-price to a general IP can be found in Barnhart et al. [1998]. In the airline crew pairing problem, the traditional branching rule, i.e. forcing a pairing to be selected or not, is difficult to implement due to the fact that the subproblem will need to forbid more and more paths from being generated as the algorithm proceeds deeper along the branch-and-bound tree. This means the subproblem may have to find the $(k+1)$th constrained shortest path if k paths are forbidden, which is time consuming. A better branching rule is based on the strategy suggested by Ryan and Foster [1981]. The branching strategy says that if $Y \in \{0,1\}^{n \times m}$ is a 0-1 matrix, and a basic solution $\theta \in \mathbb{R}^m$ to $Y\theta = 1$ is fractional, i.e. at least one of the components of θ is fractional, then there exist two rows r and s of the master problem such that $0 < \sum_{k: y_{rk}=1, y_{sk}=1} \theta_k < 1$ where y_{ik} denotes the ith row entry of column k. Based on this, the branching rule of the airline crew pairing problem forces a pair of flights to be either covered by the same pairing or by two different pairings. Moreover, since it is much easier to force two flights to appear consecutively or not, the branching rule is adjusted accordingly so that flights r and s are either consecutive in a pairing or not. This is sometimes referred to as the follow-on strategy. Desaulniers et al. [1997] implemented the branch-and-price technique in a model that solves the crew pairing problem for Air France.

Network structure for pairing generator

Pairings are generated based on the idea of multi-commodity network flows where crews are commodities moving between airports. There are two types of networks used in the literature to model the crew pairing problem. The flight network is composed of flight legs, source nodes, sink nodes and connection arcs. The network is time and space based, i.e. each node in the network defines a time at a particular airport, except the source and sink nodes. A flight leg is represented by an arc connecting a node at the departure airport at departure time and a node at the arrival airport at arrival time. A pair of nodes are connected by a connection arc if such connection is feasible both in time and space, i.e. the two nodes should correspond to the same airport, and the time between them is long enough for required rest. The cost of an arc is defined as the time interval between its two nodes, except for the arcs having source node or sink node at one end. The source nodes and sink nodes do not have time attribute, so they can connect to any departure node and arrival node with a cost of zero as long as the space feasibility is satisfied. Figure 2.2 gives an example of a flight network corresponding to the one-day schedule shown in Table 2.2.

Table 2.2: An example of one-day schedule.

Flight nb.	Departure airport	Arriving airport	Departure time	Arrival time
1	A	B	9am	10am
2	B	A	11am	11:30am
3	A	B	12:30pm	1pm
4	B	C	1:30pm	2pm
5	C	A	3pm	4pm

An alternative network structure uses duties as its building blocks. In this network, there are three types of nodes, which are source nodes, sink nodes and duty nodes. The

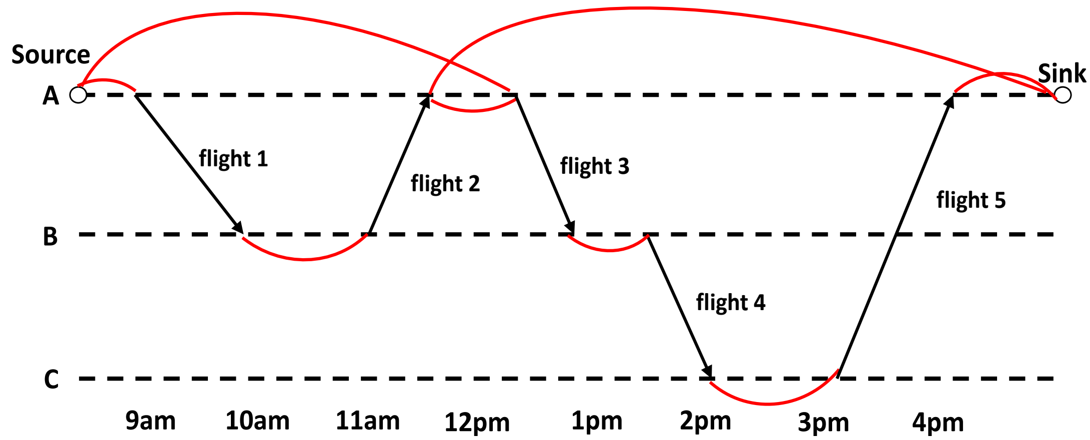

Figure 2.2: An example of flight network.

duties are constructed by enumerating all feasible combinations of scheduled flight legs for one workday. Such an enumeration is possible since flights are space and time based, and there are minimum and maximum sit time constraints. Two duty nodes are connected as long as the successive one originates at the airport where its predecessor terminates, and the rest period from the end of the first duty to the start of the second duty satisfies regulatory rules. The source and sink nodes have the same features as those in flight network, i.e. the arcs concerning them are only constrained by space feasibility with a time cost of zero. Figure 2.3 shows a duty network using a subset of duties corresponding to a three-day schedule with flights in each day replicating the schedule in Table 2.2. Table 2.3 shows all feasible duties, in which an input 1 represents duty i covers flight j, and 0 otherwise. Note that those nodes without any arcs coming in or going out can be eliminated to reduce the network size. Also, due to the time attribute of the nodes in these two types of network, the networks are directed and acyclic.

The choice of network structure depends on the geographic range of operation and

Table 2.3: All possible duties based on schedule in Table 2.2.

Duty/Flight	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
2	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0
3	0	0	1	1	1	0	1	1	1	1	1	1	0	0	0
4	0	0	0	1	1	0	0	1	1	0	1	1	1	1	0
5	0	0	0	0	1	0	0	0	1	0	0	1	0	1	1

the complexity of work rules. Specifically, duty network is typically used for international problems where the operational network tends to be sparse, while flight network is commonly used for domestic operations because the hub-and-spoke network leads to a tremendous number of feasible duties. Duty network intrinsically satisfies more rules than flight network because the construction of duties takes into account some rules defining duty feasibility, such as the maximum allowable flying time in a duty. There are, however, many rules not captured by either duty or flight network. Therefore, only a subset of the paths generated from the networks represents legal pairings.

2.3. Literature review

While most literature focuses on deterministic aircraft and crew scheduling problems and assumes that scheduled flights depart and arrive on time, there is a growing body of literature that models stability and recoverability of the schedules. Stability refers to the ability of a solution to remain feasible/optimal when delays occur. This requires incorporating protection measures during the planning process using a priori optimization methodologies. On the other hand, recoverability refers to the ability to recover from infeasibility during execution of the schedules. Re-optimization methodologies provide

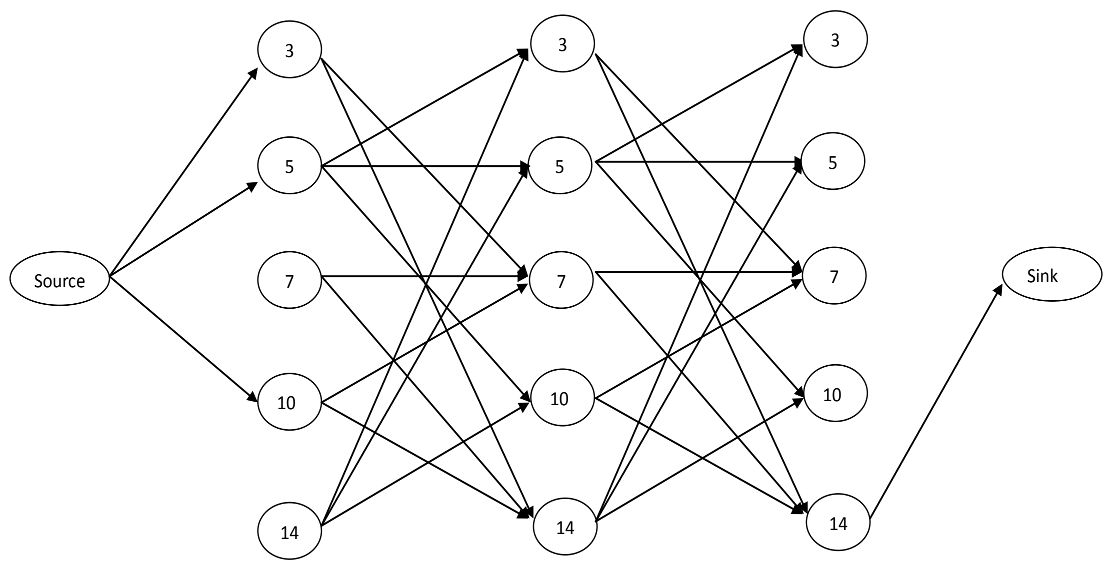

Figure 2.3: An example of duty network for a three-day schedule using duties 3, 5, 7, 10, and 14 in Table 2.2.

recovery mechanisms during execution or implementation stage [Barnhart et al., 2003; Eggenberg, 2009; Kohl et al., 2007]. The term "robust" is often used in the literature to refer to schedules that are less susceptible to changes and disruptions. To avoid confusion with robust optimization, we use the term "stable" to refer to such schedules and reserve the term "robust" to refer to schedules obtained from robust optimization modeling.

A determining factor in this context is the amount of information available to characterize uncertainty, leading to three types of models: deterministic, stochastic, and robust.

2.3.1 Deterministic optimization

Deterministic models assume that problem parameters are deterministic but incorporate measures that are argued to lead to stable solutions. Such measures include increasing crew move-ups, limiting aircraft change, and increasing connection times to allow for recovery when delays occur. Shebalov and Klabjan [2006] propose a model that improves a given schedule by maximizing the number of move-ups where a move-up is a potential swap of the remaining duties of two crews who have the same base and finish their pairings on the same day. An expected benefit is that delays may be avoided if crews can swap schedules. Another strategy to increase opportunities for aircraft and crew swaps is proposed by Smith and Johnson [2006]. They limit the number of fleet types that serve an airport when solving the fleet assignment problem. Gao et al. [2009] extend this strategy to account for crew connections within the fleet assignment problem and limit both the number of fleet types and crew bases that serve an airport. Weide et al. [2010] penalize the number of times crews swap aircraft and iteratively solve a crew pairing problem and an aircraft routing problem with modified objectives. The objective of the crew pairing problem is to minimize the sum of crew cost and penalties on aircraft changes; and the objective of the aircraft routing problem is to maximize the total weights of short connections used by both crew pairing solution and aircraft routing solution. Ehrgott and Ryan [2002] penalize short connection times where crews have to change aircraft and solve a biobjective crew scheduling problem. The first objective is to minimize crew pairing cost and the second objective is to minimize total penalty over all crews. Tekiner et al. [2009] model a crew pairing problem where some of the flights are not scheduled beforehand but may be added during execution. They propose techniques to insert these flights within crew schedules while minimizing disruptions by allowing a limited number of deadheads or through long connection times.

2.3.2 Stochastic optimization

Stochastic models assume that random parameters follow a known probability distribution and use either expected values or probabilistic measures. Sohoni et al. [2011] propose two stochastic mixed-integer models where block-time (elapsed time between the time an aircraft pushes back from the departure gate and arrives at the destination gate) follows a truncated normal distribution. The models reallocate slack time at selected connections. In one model, the objective maximizes profit while satisfying service levels on flights and passengers. In the other model, the weighted service levels are maximized with a guaranteed profit level. They develop algorithms that iteratively generate cuts to approximate the nonlinearity in these two models. Schaefer et al. [2000] use a deterministic crew pairing model where the pairing costs are expected values determined by simulation. The model assumes that the pairing cost is independent of other pairings and the simulation uses push-back recovery strategy to restore from disruption, i.e. the disrupted flights are delayed until all of the resources are available. Yen and Birge [2000] use a similar approach to Ehrgott and Ryan [2002], but study the stochastic case where arrival and departure times are random and are modeled by a set of scenarios. Tam et al. [2011] compare the two-stage stochastic model of Yen and Birge [2000] and the biobjective model of Ehrgott and Ryan [2002] using data from Air New Zealand. They conclude that both models improve crew schedule stability and observe that the biobjective model is suitable under moderate delay cost, while the stochastic model is suitable under large delay cost. Dunbar et al. [2012] minimize propagated delay for aircraft and crew by iteratively solving aircraft routing and crew pairing problems. When solving the subproblems, they incorporate aircraft delay to calculate propagated crew delay and vice versa. Lan et al. [2006] present two methods with different objectives. The first method tries to reduce delay propagation where delays follow a log-normal distribution. A second passenger-based approach minimizes the number of disrupted passengers by re-timing departure time of flights within a small time window.

2.3.3 Robust optimization

Robust models do not make any assumptions on the distribution of problem parameters, but assume that parameters vary within a known interval, and model worst-case behavior. Worst-case based optimization was introduced in the 70s [Soyster, 1973]. More recent studies include those by Ben-Tal and Nemirovski [1998], Ben-Tal and Nemirovski [1999] and Ben-Tal and Nemirovski [2000]. One drawback of this modeling approach is that robustness comes with a significant increase in cost. Bertsimas and Sim [2004] propose to control the robustness of the solution by bounding the worst case by a parameter. Bertsimas and Sim [2003] focus on discrete optimization and show that certain types of robust discrete problems can be solved as a series of deterministic problems. To be specific, constraint and objective coefficients vary within an interval. To control the conservatism of the robust solutions, the number of coefficients that may deviate from their nominal value in each constraint is limited by an upper bound called protection level. The model finds a solution that optimizes the objective function in the worst case with a given protection level and guarantees to be feasible when the number of random coefficients in each robust constraint is limited by the corresponding protection level.

To the best of our knowledge, no previous work studies the crew pairing problem under data uncertainty from the point of view of robust optimization. We use the modeling framework of Bertsimas and Sim [2003] and propose for the first time the robust crew pairing problem where both the objective function and resource constraints include a robust term. In this framework, data uncertainty is modeled by intervals and does not require the distribution of delays as input for the model. Instead, the user determines the reasonable ranges for the delays. The cost of pairing flights varies randomly in an interval for which the lower limit is referred to as the nominal cost and the length of the interval is referred to as the maximum deviation. The level of robustness of the solution is controlled by specifying a protection level which sets a maximum number of deviations from the nominal cost on any crew pairing. The proposed model includes nonlinear robust terms that account for the cost of deviation in the objective function and resource constraints.

2.4. Robust crew pairing formulation

The crew pairing problem is defined on a set of crews, K, and a flight network $G = (N, A)$; where N is the set of nodes and A is set of arcs. A flight network is typically used to schedule domestic operations due to the large number of feasible duties [Barnhart et al., 2003; Minoux, 1984; Desrosiers et al., 1991; Klabjan et al., 2001a; Shebalov and Klabjan, 2006], while a duty network is typically used to model international problems [Barnhart et al., 1995; Desaulniers et al., 1997; Klabjan et al., 2001b]. A hybrid of the two networks is used in Makri and Klabjan [2004]. There are three types of nodes in N: crew origin nodes denoted by set O, crew destination nodes denoted by set D, and flight nodes denoted by $N\backslash\{O, D\}$. Flight nodes are time and location dependent and are associated with the departure time and departing airport. The base of crew $k \in K$ is denoted by the origin and destination nodes o^k and d^k, respectively. The set A consists of three types of arcs. Arc (o^k, i) exists if crew k can start its schedule with flight i, and arc (i, d^k) exists if crew k can end its schedule with flight i. Arc (i, j) exists if flight i can be followed by flight j and the minimum connection time is satisfied. Arc (i, j) accounts for flight leg and flight connection and is referred to as connection (i, j) in the rest of the chapter. Let $G^k = (N^k, A^k)$ be the subgraph of G associated with crew k, constructed by removing the nodes and arcs that can not be visited by crew k from G.

According to Lavoie et al. [1988], a function of total time away from base reasonably represents pairing cost. We use total time away from base as a measure of pairing cost and refer to time and cost interchangeably. Also, we consider random arc costs that vary in an interval $[c_{ij}, c_{ij} + h_{ij}]$. The nominal cost c_{ij} is what would be the cost of arc (i, j) in the deterministic case. It is the advertised time in flight schedules, and it is the time between the departure of flight i and the departure of flight j, including flying time of flight i and the connection time between flights i and j. The parameter h_{ij} is the maximum deviation from the nominal cost of arc (i, j), and includes the delay of flight i and the additional time spent connecting between flights i and j. Deviation then represents the total delay between the departure of flights i and j. Without loss of generality, we assume the nominal cost and maximum deviation of arcs (o^k, j) and (i, d^k) are zero. Let $T_k \subseteq A^k$ be the set of arcs $(i, j) \in A^k$ with $h_{ij} > 0$, and let $\mathit{\Gamma}_k$ be the maximum number of arcs that may experience deviation for any pairing of crew k, referred to as the protection level. The protection level is a user defined parameter to control the level of robustness of the solution. In practice system operators may experiment with several protection levels and analyze the trade-offs between system costs and robustness of the solution. The maximum duration, $\mathcal{D}(\mathit{\Gamma}_k)$, of a pairing is a function of $\mathit{\Gamma}_k$ and is limited by the maximum allowable time crew k may spend away from base, denoted by b^k. This is a strict rule that is modeled as a resource constraint.

The robust crew pairing formulation finds solutions that remain feasible with respect to the resource constraints when up to $\mathit{\Gamma}_k$ arcs experience the maximum deviation on any pairing for crew $k \in K$. The objective is to find a schedule that minimizes the total crew pairing cost when up to $\mathit{\Gamma}_k$ arcs experience the maximum deviation on any pairing for crew $k \in K$. The robust crew pairing formulation is

$$
\begin{aligned}
\text{[P1]:} \quad \min \quad & \sum_{k \in K} \sum_{(i,j) \in A^k} c_{ij} y_{ij}^k + \sum_{k \in K} \beta_k(y^k) && && (2.4) \\
\text{s.t.} \quad & \sum_{k \in K} \sum_{i:(i,j) \in A^k} y_{ij}^k = 1 && j \in N \backslash \{O, D\} && (2.5) \\
& \sum_{j:(o^k,j) \in A^k} y_{o^k,j}^k = 1 && k \in K && (2.6) \\
& \sum_{i:(i,j) \in A^k} y_{ij}^k - \sum_{i:(j,i) \in A^k} y_{ji}^k = 0 && j \in N^k \backslash \{o^k, d^k\}, k \in K && (2.7) \\
& \sum_{i:(i,d^k) \in A^k} y_{i,d^k}^k = 1 && k \in K && (2.8) \\
& \mathcal{D}(\mathit{\Gamma}_k) \leq b^k && k \in K && (2.9) \\
& y_{ij}^k \in \{0, 1\} && (i,j) \in A^k, k \in K && (2.10)
\end{aligned}
$$

where y_{ij}^k is a binary decision variable which equals 1 if arc $(i,j) \in A^k$ is selected in a pairing for crew k and 0 otherwise, y^k is a vector of decision variables y_{ij}^k,

$$\mathcal{D}(\Gamma_k) = \sum_{(i,j)\in A^k} c_{ij} y_{ij}^k + \beta_k(y^k), \tag{2.11}$$

and

$$\text{[BETA]:} \quad \beta_k(y^k) = \max_{\{S_k | S_k \subseteq T_k, |S_k| \leq \Gamma_k\}} \left\{ \sum_{(i,j)\in S_k} h_{ij} y_{ij}^k \right\}. \tag{2.12}$$

The inner maximization problem [BETA] finds a subset S_k of the arcs such that $|S_k| \leq \Gamma_k$, and the sum of the maximum deviations on the arcs in S_k is maximized. The objective function (2.4) minimizes the robust crew pairing cost as the sum of the nominal cost and the maximum deviation cost with a given number of allowable deviations Γ_k for each crew $k \in K$. Constraints (2.5) make sure that each flight is covered by exactly one crew. Constraints (2.6) to (2.8) are flow balance constraints. Constraints (2.9) are the robust resource constraints that treat the maximum time-away-from-base as a resource limit. A feasible pairing with respect to this constraint is one where the duration of a pairing for crew k under the maximum deviation of at most Γ_k arcs is less than or equal to the resource limit b_k. The model assumes a push-back recovery strategy where a flight is delayed until all of the resources are available.

Model [P1] differs from the deterministic formulation of crew pairing in that it includes nonlinear terms in the objective function and constraints. In addition to size, the nonlinearities represent the main difficulty in [P1]. The deterministic crew pairing problem is of large-scale and so is the robust case. Moreover as robustness is introduced into the problem, the problem becomes nonlinear due to the inner maximization problem $\beta_k(y^k)$. Consequently, it is not possible to solve [P1] directly. One alternative is to transform [P1] into a MIP by dualizing the inner maximization as shown by Bertsimas and Sim [2003]. However, this results in $(|K|+1)(|A|+1)$ additional continuous variables and $(|K|+1)|A|$ additional constraints. Moreover, it is shown by Bertsimas and Sim [2003] that the robust counterpart of an NP-hard combinatorial problem is also NP-hard. Clearly, the deterministic crew pairing problem is NP-hard, and so is [P1]. As for the deterministic crew pairing problem, decomposition is the most successful exact approach. Applying Lagrangian relaxation to [P1] does not solve the nonlinearity issue but moves all nonlinearities to the subproblem. It turns out that dealing with the nonlinear terms within the subproblem is less challenging. Next we discuss how to decompose [P1] using Lagrangian relaxation and relate that to Dantzig-Wolfe decomposition to obtain feasible solutions. Then we discuss how to linearize $\beta_k(y^k)$ and show that the subproblem can be solved as a series of deterministic nominal problems.

2.5. Solution methodology

Applying Lagrangian relaxation on the linking constraints (2.5) using variables $\alpha_j, j \in N \backslash \{O, D\}$ results in the relaxed problem:

$$
\begin{aligned}
\text{[LR]:} \quad \min \quad & \sum_{k\in K}\sum_{(i,j)\in A^k} c_{ij}y_{ij}^k + \sum_{k\in K}\beta_k(y^k) + \\
& \sum_{j\in N\setminus\{O,D\}} \alpha_j \left(1 - \sum_{k\in K}\sum_{i:(i,j)\in A^k} y_{ij}^k\right) && (2.13)\\
\text{s.t.} \quad & \text{constraints (2.6) to (2.10)},
\end{aligned}
$$

which further decomposes into $|K|$ subproblems, one for each crew $k \in K$.

$$
\begin{aligned}
[\text{SP}_k]: \quad RC_k(\alpha) = \ \min & \sum_{(i,j)\in A^k} c_{ij}y_{ij}^k + \beta_k(y^k) - \sum_{j\in N\setminus\{O,D\}}\sum_{i:(i,j)\in A^k} \alpha_j y_{ij}^k && (2.14)\\
\text{s.t.} \quad & \sum_{j:(o^k,j)\in A^k} y_{o^k,j}^k = 1 && (2.15)\\
& \sum_{i:(i,j)\in A^k} y_{ij}^k - \sum_{i:(j,i)\in A^k} y_{ji}^k = 0 \qquad j \in N^k\setminus\{o^k, d^k\} && (2.16)\\
& \sum_{i:(i,d^k)\in A^k} y_{i,d^k}^k = 1 && (2.17)\\
& \sum_{(i,j)\in A^k} c_{ij}y_{ij}^k + \beta_k(y^k) \le b^k && (2.18)\\
& y_{ij}^k \in \{0,1\} \qquad (i,j) \in A^k. && (2.19)
\end{aligned}
$$

Without the term $\beta_k(y^k)$, the subproblem would be a shortest path problem with resource constraints. We refer to $[\text{SP}_k]$ as a robust shortest path problem with resource constraints defined on an acyclic network G^k. If a feasible crew schedule exists in the network G^k, the feasible region of subproblem [SP$_k$] is nonempty and bounded for all α. Let H_k, indexed by h, be the set of feasible paths for subproblem [SP$_k$], and let us represent path h by a binary vector y^{kh} where $y_{ij}^{kh} = 1$ if arc (i, j) is on path h. The Lagrangian master problem is given by:

$$
\begin{aligned}
\text{[MP]:} \quad \max \quad & \sum_{j \in N \setminus \{O,D\}} \alpha_j + \sum_{k \in K} \theta_k && && (2.20) \\
\text{s.t.} \quad & \theta_k \leq \sum_{(i,j) \in A^k} c_{ij} y_{ij}^{kh} + \beta_k(y^{kh}) - \\
& \quad \sum_{j \in N \setminus \{O,D\}} \sum_{i:(i,j) \in A^k} \alpha_j y_{ij}^{kh} && h \in H_k, k \in K && (2.21) \\
& \theta_k \text{ free} && k \in K && (2.22) \\
& \alpha_j \text{ free} && j \in N \setminus \{O, D\}. && (2.23)
\end{aligned}
$$

We use a cutting plane method to solve [MP], where the solution of the subproblem [SP$_k$] gives a cutting plane in the form of constraint (2.21) while the solution of [MP] proposes new variables α_j to the subproblem [SP$_k$] in each iteration. The algorithm terminates when the lower bound $\sum_{j \in N \setminus \{O,D\}} \alpha_j + \sum_{k \in K} RC_k(\alpha)$ is equal to the upper bound $\sum_{j \in N \setminus \{O,D\}} \alpha_j + \sum_{k \in K} \theta_k$.

To generate feasible solutions to [P1], we use dual information generated when the Lagrangian master problem is solved. Define decision variable $\lambda_k^h \in [0, 1]$ which is the fraction of crew k that is assigned a pairing corresponding to solution y^{kh}, and let parameter $a_{ik}^h = 1$ if flight i is covered in solution y^{kh}, and $w_k^h = \sum_{(i,j) \in A^k} c_{ij} y_{ij}^{kh} + \beta_k(y^{kh})$. The dual of [MP] is the Dantzig-Wolf master problem:

$$\text{[DW]:} \quad \min \sum_{k \in K} \sum_{h \in H_k} w_k^h \lambda_k^h \tag{2.24}$$

$$\text{s.t.} \sum_{k \in K} \sum_{h \in H_k} a_{ik}^h \lambda_k^h = 1 \qquad i \in N \backslash \{O, D\} \tag{2.25}$$

$$\sum_{h \in H_k} \lambda_k^h = 1 \qquad k \in K \tag{2.26}$$

$$0 \leq \lambda_k^h \leq 1 \qquad k \in K, h \in H_k. \tag{2.27}$$

At each iteration of the cutting plane algorithm, solving the relaxed Lagrangian master problem is the same as solving a restricted Dantzig-Wolfe master problem defined on the same subset of paths, i.e., a restriction of [DW] defined on a subset of pairings. When $\lambda_k^h \in \{0, 1\}$, the binary [DW] master problem is the set-partitioning formulation equivalent to [P1]. When the cutting plane algorithm terminates, the Lagrangian master problem is solved and the Lagrangian bound is obtained. At this point, a set of pairings, i.e., subproblem solutions, are generated. We use these pairings to form a restricted [DW], enforce binary requirements on λ_k^h, and solve the resulting model to obtain a feasible solution and a valid upper bound on [P1]. The experimental results presented in Section 6 show that the feasible solutions are optimal in most cases, and are very close to optimal in the remaining cases.

2.6. Robust shortest path with single resource constraints

When solving the deterministic crew pairing problem using column generation, the subproblem is a deterministic SPPRC which is known to be NP-hard [Ball, 1995]. The subproblem in our case is a robust SPPRC, and it reduces to the deterministic problem when $\Gamma = 0$. Hence, it is also NP-hard. A robust formulation of the shortest path problem is presented in Bertsimas and Sim [2003] where the objective function coefficients are uncertain. This is different from the problem we treat in that it has resource constraints and has robust terms both in the objective function and constraints. Next, we prove that the solution methodology proposed by Bertsimas and Sim [2003] may be extended and that [SP$_k$] may be solved as a series of nominal problems.

Let c and $\bar{c}$ be the vectors of arc costs c_{ij} and modified arc costs $c_{ij} - \alpha_j$ for all arcs $(i,j) \in A^k$. For ease of exposition, we drop index k, define $n = |A^k|$, introduce index $t = 1, ..., n$ on the arcs $(i,j) \in A^k$, and rewrite [SP$_k$] in a compact form:

$$[\text{P2}]: \quad Z = \min \quad \bar{c}^\top y + \max_{\{S|S\subseteq T, |S|\leq \Gamma\}} \sum_{t\in S} h_t y_t \tag{2.28}$$

$$\text{s.t.} \quad y \in Y \tag{2.29}$$

$$c^\top y + \max_{\{S|S\subseteq T, |S|\leq \Gamma\}} \sum_{t\in S} h_t y_t \leq b \tag{2.30}$$

where Y is the feasible set defined by flow balance constraints (2.15), (2.16), (2.17) and the binary constraint (2.19). Without loss of generality, we assume that index t is ordered such that $h_1 \geq h_2 \geq ... \geq h_n$, and $h_{n+1} = 0$. Taking advantage of the special structure of [P2], we show in Theorems 2.1 and 2.2 that it may be solved by solving at most $n + 1$ deterministic resource-constrained shortest path problems.

Theorem 2.1. *Solving problem [P2] is equivalent to solving problem [P3]:*

$$\text{[P3]:} \quad Z = \min \quad \bar{c}^\top y + \min\left\{U^1(y), \ldots, U^{d-1}(y), \ldots, U^{n+1}(y)\right\} \tag{2.31}$$

$$s.t. \quad y \in Y \tag{2.32}$$

$$c^\top y + \min\left\{U^1(y), \ldots, U^{d-1}(y), \ldots, U^{n+1}(y)\right\} \leq b \tag{2.33}$$

where

$$U^d(y) = \begin{cases} \Gamma h_1, & d = 1, \\ \Gamma h_{d-1} + \sum\limits_{t=1}^{d}(h_t - h_d)y_t, & d = 2, \ldots, n, \\ \sum\limits_{t=1}^{n} h_t y_t, & d = n + 1. \end{cases} \tag{2.34}$$

Proof of Theorem 2.1. For a given $\overline{y} \in Y$, the inner maximization is a bounded knapsack problem:

$$\text{[KN]:} \quad U(\overline{y}) = \max \quad \sum_{t \in T} h_t \overline{y}_t u_t \tag{2.35}$$

$$\text{s.t.} \quad \sum_{t \in T} u_t \leq \Gamma \tag{2.36}$$

$$u_t \in \{0, 1\} \quad t = 1, ..., n. \tag{2.37}$$

Since $\varGamma$ is integer, the binary requirement on u_t may be relaxed, and u_t becomes a continuous variable in $[0, 1]$. [KN] becomes:

$$\text{[KN]:} \quad U(\overline{y}) = \max \quad \sum_{t=1}^{n} h_t \overline{y}_t u_t \tag{2.38}$$

$$\text{s.t.} \quad \sum_{t=1}^{n} u_t \leq \varGamma \tag{2.39}$$

$$0 \leq u_t \leq 1 \quad t = 1, ..., n. \tag{2.40}$$

Define $v \geq 0$ as the dual variable for constraint (2.39), $p_t \geq 0, t = 1, ..., n$ as the dual variable for constraints (2.40). The dual problem of [KN] is:

$$\text{[DKN]:} \quad U(\overline{y}) = \min \quad \varGamma v + \sum_{t=1}^{n} p_t \tag{2.41}$$

$$\text{s.t.} \quad p_t + v \geq h_t \overline{y}_t \quad t = 1, ..., n \tag{2.42}$$

$$p_t, v \geq 0 \qquad t = 1, ..., n. \tag{2.43}$$

Observing that constraints (2.42) are equivalent to $p_t \geq h_t \overline{y}_t - v$, and $p_t \geq 0, t = 1, ..., n$, we can write p_t as:

$$p_t = \max\{h_t \overline{y}_t - v, 0\}, \quad t = 1, ..., n. \tag{2.44}$$

Since $\overline{y}_t$ is binary, equality (2.44) can be written as:

$$p_t = \max\{h_t - v, 0\} \overline{y}_t, \quad t = 1, ..., n. \tag{2.45}$$

Replacing p_t with equality (2.45), [DKN] is reformulated as:

$$\text{[DKN]:} \quad U(\overline{y}) = \min \quad \mathit{\Gamma} v + \sum_{t=1}^{n} (\max\{h_t - v, 0\})\overline{y}_t \tag{2.46}$$

$$\text{s.t.} \quad v \geq 0. \tag{2.47}$$

Recall that h_t is ordered such that $h_1 \geq h_2 \geq \cdots \geq h_n \geq h_{n+1} = 0$. Partitioning the feasible range of v into $n+1$ intervals, $\max\{h_t - v, 0\}\overline{y}_t$ can be linearized over each interval and [DKN] can be solved by exploring all these intervals. To be specific, for $v \in [h_1, \infty)$,

$$\max\{h_t - v, 0\}\overline{y}_t = 0, t = 1, \ldots, n+1,$$

and for $v \in [h_d, h_{d-1}]$, $d = 2, \ldots, n+1$,

$$\max\{h_t - v, 0\}\overline{y}_t = (h_t - v)\overline{y}_t, t = 1, \ldots, d-1,$$

while

$$\max\{h_t - v, 0\}\overline{y}_t = 0, t = d, \ldots, n+1.$$

Based on this, the following equation holds:

$$\sum_{t=1}^{n} (\max\{h_t - v, 0\})\overline{y}_t = \begin{cases} 0, & \text{if } v \in [h_1, \infty), \\ \sum\limits_{t=1}^{d-1} (h_t - v)\overline{y}_t, & \text{if } v \in [h_d, h_{d-1}], d = n+1, \ldots, 2. \end{cases} \tag{2.48}$$

Using equation (2.48), [DKN] decomposes into $n+1$ subproblems, where in each subproblem v is defined on a specified interval. Let us use index $d = 1, \ldots, n+1$ to denote these subproblems. For $d = 1$,

$$\text{[SubDKN}^1\text{]:} \quad U^1(\overline{y}) = \min \quad \Gamma v \tag{2.49}$$

$$\text{s.t.} \quad v \in [h_1, \infty). \tag{2.50}$$

In this case $\sum_{t=1}^{n}(\max\{h_t - v, 0\})\overline{y}_t = 0$ and it is removed from the objective function. Then, the optimal solution of [SubDKN^1] is obtained at $v = h_1$. For $d = 2, \ldots, n+1$, subproblem d is given by

$$\text{[SubDKN}^d\text{]:} \quad U^d(\overline{y}) = \min \quad \Gamma v + \sum_{t=1}^{d-1}(h_t - v)\overline{y}_t \tag{2.51}$$

$$\text{s.t.} \quad v \in [h_d, h_{d-1}]. \tag{2.52}$$

Rearranging the objective function (2.51), [SubDKN^d] becomes:

$$\text{[SubDKN}^d\text{]:} \quad U^d(\overline{y}) = \min \quad (\Gamma - \sum_{t=1}^{d-1}\overline{y}_t)v + \sum_{t=1}^{d-1}h_t\overline{y}_t \tag{2.53}$$

$$\text{s.t.} \quad v \in [h_d, h_{d-1}]. \tag{2.54}$$

Since $\sum_{t=1}^{d-1} h_t\overline{y}_t$ is a constant, the linear objective function (2.53) optimizes v over interval $[h_d, h_{d-1}]$. The optimal solution is obtained at $v = h_d$ or $v = h_{d-1}$. Consequently, subproblem d becomes

$$[\text{SubDKN}^d]: \quad U^d(\overline{y}) = \min\left\{\mathit{\Gamma} h_{d-1} + \sum_{t=1}^{d-1}(h_t - h_{d-1})\overline{y}_t, \mathit{\Gamma} h_d + \sum_{t=1}^{d-1}(h_t - h_d)\overline{y}_t\right\}$$

$$= \min\left\{\mathit{\Gamma} h_{d-1} + \sum_{t=1}^{d-1}(h_t - h_{d-1})\overline{y}_t, \mathit{\Gamma} h_d + \sum_{t=1}^{d}(h_t - h_d)\overline{y}_t\right\}. \quad (2.55)$$

Since subproblems [SubDKN^d], $d = 1, ..., n+1$, partition the feasible region of [DKN], the optimal solution of [DKN] is the minimum solution over all [SubDKN^d], that is:

$$[\text{DKN}]: \quad U(\overline{y}) = \min \left\{\mathit{\Gamma} h_1, \ldots, \mathit{\Gamma} h_{d-1} + \sum_{t=1}^{d-1}(h_t - h_{d-1})\overline{y}_t, \mathit{\Gamma} h_d + \sum_{t=1}^{d}(h_t - h_d)\overline{y}_t, \ldots, \right.$$
$$\left. \mathit{\Gamma} h_n + \sum_{t=1}^{n}(h_t - h_n)\overline{y}_t, \sum_{t=1}^{n} h_t\overline{y}_t\right\}. \quad (2.56)$$

For ease of exposition, let us redefine $U^d(\overline{y})$ as follows:

$$U^d(\overline{y}) = \begin{cases} \mathit{\Gamma} h_1, & d = 1, \\ \mathit{\Gamma} h_d + \sum\limits_{t=1}^{d}(h_t - h_d)\overline{y}_t, & d = 2, \ldots, n, \\ \sum\limits_{t=1}^{n} h_t\overline{y}_t, & d = n+1. \end{cases} \quad (2.57)$$

Then,

$$[\text{DKN}]: \quad U(\overline{y}) = \min \left\{U^1(\overline{y}), \ldots, U^d(\overline{y}), \ldots, U^{n+1}(\overline{y})\right\}. \quad (2.58)$$

Replacing the inner maximization in [P2] with $U(\overline{y})$ defined as in (2.58), [P2] becomes:

$$\begin{aligned} \text{[P3]:} \quad Z = \min \quad & \bar{c}^\top y + \min\left\{U^1(y), \ldots, U^d(y), \ldots, U^{n+1}(y)\right\} && (2.59)\\ \text{s.t.} \quad & y \in Y && (2.60)\\ & c^\top y + \min\left\{U^1(y), \ldots, U^d(y), \ldots, U^{n+1}(y)\right\} \leq b. && (2.61) \end{aligned}$$

□

Theorem 2.1 applies the techniques of Bertsimas and Sim [2003] to transform the inner maximization into an inner minimization. The following theorem states that the nonlinear problem [P3] can be solved as a series of linear problems.

Theorem 2.2. *Solving [P3] is equivalent to solving problem [P4]:*

$$\text{[P4]:} \quad Z = \min_{d=1,\ldots,n+1} \left\{Z^d\right\} \qquad (2.62)$$

where Z^d is the objective function value of the nominal problem:

$$\begin{aligned} [N^d]: \quad Z^d = \min \quad & \bar{c}^\top y + U^d(y) && (2.63)\\ s.t. \quad & y \in Y && (2.64)\\ & c^\top y + U^d(y) \leq b && (2.65) \end{aligned}$$

for $d = 1, ..., n+1$.

Proof of Theorem 2.2. To show the equivalence, we first prove that the feasible regions of [P3] and [P4] are identical, i.e. any feasible solution of [P3] is feasible to at least one nominal problem [N^d], and any infeasible solution of [P3] is infeasible to all nominal problems [N^d] for $d = 1, \dots, n+1$.

- Suppose $\hat{y}$ is feasible to [P3], then $\hat{y} \in Y$ and there exists $U^d(\hat{y})$ such that $c^\top \hat{y} + U^d(\hat{y}) \leq b$. It follows that $\hat{y}$ is feasible to at least one nominal problem [N^d], $d = 1, \dots, n+1$.

- Now, suppose $\hat{y}$ is infeasible to [P3], then

$$\hat{y} \notin Y \text{ or } \; c^\top \hat{y} + \min\{U^1(\hat{y}), \dots, U^d(\hat{y}), \dots, U^{n+1}(\hat{y})\} > b.$$

  If $\hat{y} \notin Y$ then $\hat{y}$ is infeasible to [N^d], $d = 1, \dots, n+1$.
  Else $c^\top \hat{y} + \min\{U^1(\hat{y}), \dots, U^d(\hat{y}), \dots, U^{n+1}(\hat{y})\} > b$, then

$$c^\top \hat{y} + U^d(\hat{y}) \geq c^\top \hat{y} + \min\{U^1(\hat{y}), \dots, U^d(\hat{y}), \dots, U^{n+1}(\hat{y})\} > b, d = 1, \dots, n+1.$$

  It follows that $\hat{y}$ is infeasible to [N^d], $d = 1, \dots, n+1$.

Therefore, the feasible regions of [P3] and [P4] are identical. Then, we prove that the objective value of any feasible solution of [P3] is the same as its objective value in [P4]. Suppose $\hat{y}$ is a feasible solution of [P3], then

$$\begin{aligned} Z(\hat{y}) &= \bar{c}^\top \hat{y} + \min\left\{U^1(\hat{y}), \dots, U^d(\hat{y}), \dots, U^{n+1}(\hat{y})\right\} \\ &= \min\left\{U^1(\hat{y}) + \bar{c}^\top \hat{y}, \dots, U^d(\hat{y}) + \bar{c}^\top \hat{y}, \dots, U^{n+1}(\hat{y}) + \bar{c}^\top \hat{y}\right\} \\ &= \min\left\{Z^1(\hat{y}), \dots, Z^d(\hat{y}), \dots, Z^{n+1}(\hat{y})\right\}, \end{aligned}$$

which is exactly the objective value of [P4]. The equivalence between [P3] and [P4] is proved. □

Theorems 2.1 and 2.2 state that the robust SPPRC is solved as $n+1$ nominal problems. To be precise, at most $n+1$ nominal problems need to be solved. For the nominal problems d and $d+1$, their objective function and resource constraints are the same when $h_d = h_{d+1}$, i.e. $Z^d = Z^{d+1}$ and $c^\top \hat{y} + U^d(\hat{y}) = c^\top \hat{y} + U^{d+1}(\hat{y}) \leq b$, $d = 1, ..., n$. This means that only those nominal problems with distinct values of h_d need to be solved. Moreover, solving [N^{d+1}] after [N^d] requires updating the cost of arcs $t = 1, \ldots, d$ to obtain the network for [N^{d+1}]. If the optimal path for [N^d] does not use any arc $t = 1, \ldots, d$, this path remains optimal to [N^{d+1}].

Define $B^d = \{t | y_t^d = 1, t = 1, ..., |A|\}$ as the set of arcs used by the optimal path y^d of problem [N^d]. Problem [P4] is solved using the following algorithm:

Algorithm (**SubAlgo**):

Initialize: $Z^* = \infty$, $y^* = \varnothing$, $d = 1$.

Step 1: Solve nominal problem [N^1] with optimal solution y^1.

If $Z^1 < Z^*$, update $Z^* = Z^1$, $y^* = y^1$, $d = d + 1$.

Step 2: While $d < n + 1$

2.a If $h_d \neq h_{d-1}$ and $B^{d-1} \cap \{1, ..., d-1\} \neq \varnothing$, go to Step 2.b, else go to Step 2.c.

2.b Solve nominal problem [N^d]. If $Z^d < Z^*$, update $Z^* = Z^d$ and $y^* = y^d$.

2.c Update $d = d + 1$.

End While

The algorithm sequentially creates and solves a set of nominal problems where the arc costs are modified according to the current h_d. Step 2.a forbids solving a nominal problem that is either exactly the same as the previous one or that will not improve the solution. The nominal problems [N^d] are shortest path problems solved using a label-setting algorithm. The solution of the subproblem is the best among all nominal problems' solutions and is updated during the process. Compared to the deterministic case, Algorithm **SubAlgo** solves at most n additional shortest path problems.

2.7. Computational tests

The computational tests are conducted on a workstation with Xeon processor and 8GB RAM. The complete methodology and the label-setting algorithm are implemented in C++. The relaxed Lagrangian master problem and the binary Dantzig-Wolfe master problem are solved using CPLEX 12.2. The testing is carried out in two parts. In the first part, we build 89 instances based on a flight schedule of a European airline to test the effectiveness of the solution methodology in terms of solution quality and CPU time. We also investigate the impact of problem parameters such as the protection level Γ, the number of crews available, and the maximum deviation h_{ij}. In the second part, we use simulation to evaluate the robust schedules and compare to deterministic ones.

2.7.1 Testing based on data from a European airline

The European airline under study operates 46 flight legs per day that span 7 airports. The planning period is four days, which is typical in the airline industry. The airline uses the same base for its crew, and enforces a minimum connection time of 30 minutes, and a maximum layover time of 24 hours. The nominal arc cost c_{ij} between nodes i and j is defined as the elapsed time in minutes between scheduled departure of flight i and scheduled departure of flight j. The maximum deviation h_{ij} denotes the delay between flights i and j, and is uniformly generated from $\{30, 45, 60, 75, 90, 105, 120\}$ in minutes. The maximum time-away-from-base is 5760 minutes or 4 days, which means every crew must return to base within 4 days.

To investigate the effects of the number of crews $|K|$ and the protection level Γ, we generate 89 instances with varying values for $|K|$ and Γ. The number of crews has an impact on the number of flights in a pairing. Fewer crews lead to longer pairings since all flights have to be covered by a pairing. However, there should be enough crews for the instances to remain feasible with respect to the planning period of 4 days. On the other hand, crews may not be assigned to pairings when the number is large. Taking this into account, the number of crews $|K|$ is set to $\{20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30\}$. The protection level sets a limit on the number of flights that could experience delay in a pairing. So, reasonable values for the protection level should not exceed the length of the pairing. Analyzing the data we have, we found that the longest pairing generated covers 22 flights. Then, the protection level is set to $\{1, 2, 3, 4, 5, 10, 15\}$, except for $|K| = 20$ where it is increased by an increment of 1 between 1 and 19. Tables 2.5 and 2.6 report on the Lagrangian bound, the IP solution cost, the optimality gap, the percentage increase in the IP solution for each value of Γ compared to $\Gamma = 1$, the number of columns generated, and the CPU time in seconds used in each part of the algorithm. The optimality gap is calculated using the Lagrangian bound and the IP feasible solution obtained by solving the binary DW master problem defined on the subset of pairings generated in the process of calculating the bound. The notation is listed in Table 2.4 and summary results are given in Table 2.7.

Table 2.4: List of notation.

LRB:	The Lagrangian bound.
IP:	The IP solution cost.
Gap:	The optimality gap in percentage.
Inc:	The percentage increase in IP objective value.
NbCol:	The number of robust feasible pairings generated.
MST:	CPU time used in solving the master problem.
SubT:	CPU time used in solving the subproblem.
Tot:	CPU time used to calculate Lagrangian bound.
IPT:	CPU time used to calculate IP solution.

Tables 2.5 and 2.6 are organized so that the instance is solved repeatedly by fixing the number of crews available $|K|$ and varying the protection level Γ. The value of Γ is an indication of the level of uncertainty expected during operation as it sets the number of arcs in a pairing that may experience delay, and can be estimated using historical data. In our experiments we vary Γ and study its impact on the cost. As Γ increases, the number of arcs that may experience delay increases which is expected to lead to more costly pairings. This is confirmed in columns **IP** in Tables 2.5 and 2.6, which show that cost increases by about 13% when Γ increases from 1 to 15 for all values of $|K|$. On the other hand, Figure 2.4 plots the IP solution cost as a function of $|K|$ for $\Gamma = 1, 10, 15$ and shows that for the same protection level, the IP solution cost decreases as more crews become available. In other words when more crews are available, the number of flights delayed has a lower impact on the overall operation. As more crews become available, the number of crew pairings increases and the number of flights in a pairing decreases. This results in a

decrease in the total nominal cost as well as a decrease in the robust cost because shorter pairings limit the number of flights that may be delayed in a pairing. This also explains why all crews available are used in the IP solution.

There is no strong evidence on whether the quality of the Lagrangian bound improves or deteriorates as Γ increases. While the optimality gap fluctuates with no particular trend, it is 0 in 50% of the cases and is on average 0.03%. The variation in the gap is probably due to the IP solution and not to the Lagrangian bound. On the other hand, CPU time usage increases with Γ. The increase in time is mainly used in the solution of the master problem. Figure 2.5 plots CPU time and number of columns as a function of Γ for the first 19 instances with $|K| = 20$. It shows that the number of columns generated increases as Γ increases. CPU time fluctuates when Γ is small and shows a decreasing trend as Γ increases. When Γ is large, more partial paths become infeasible, which makes the label setting algorithm take less time. Figure 2.6 plots the average CPU time and the average number of columns in function of $|K|$. The plot indicates that instances become easier as $|K|$ increases.

2.7.2 Evaluation of robust modeling by simulation

We run a total of 24000 simulation runs to evaluate crew pairings built by the robust formulation, and compare to those built by deterministic modeling. We use 20 instances defined on the same network and deterministic flight information as in Section 2.7.1 for $|K| = 20$ and $\Gamma = 0, ..., 19$, and we generate the maximum delay h_{ij} in three ways to reflect high, low, and time of day dependent delay. The maximum delay h_{ij} is set in multiples

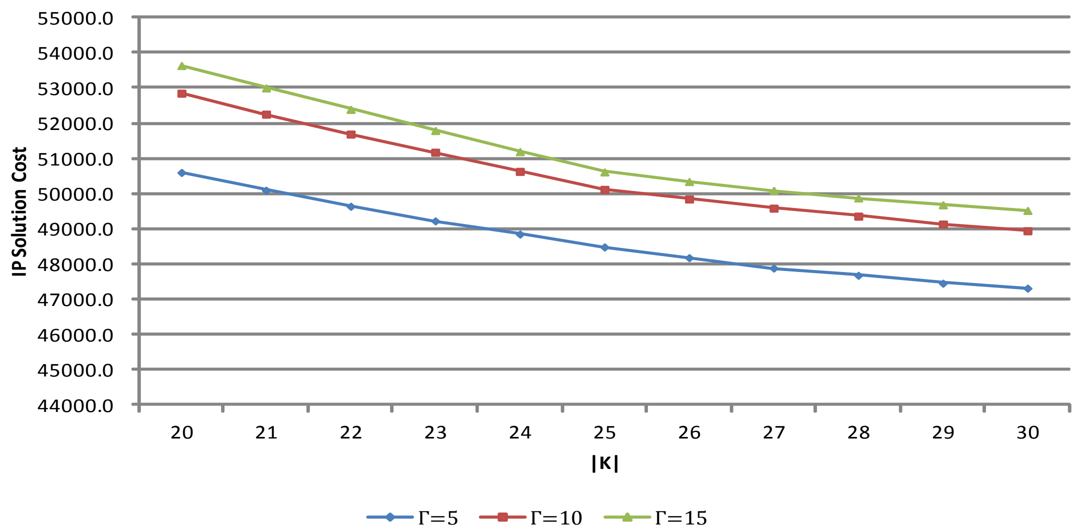

Figure 2.4: Trade-off between IP solution cost and number of crews for $\Gamma = 5, 10, 15$.

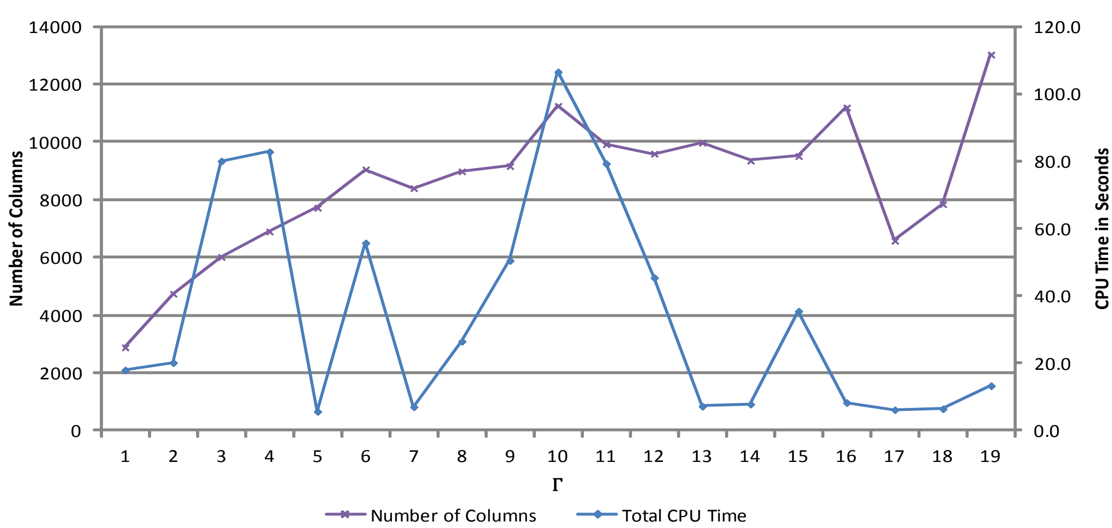

Figure 2.5: CPU time and number of columns in function of Γ for $|K| = 20$

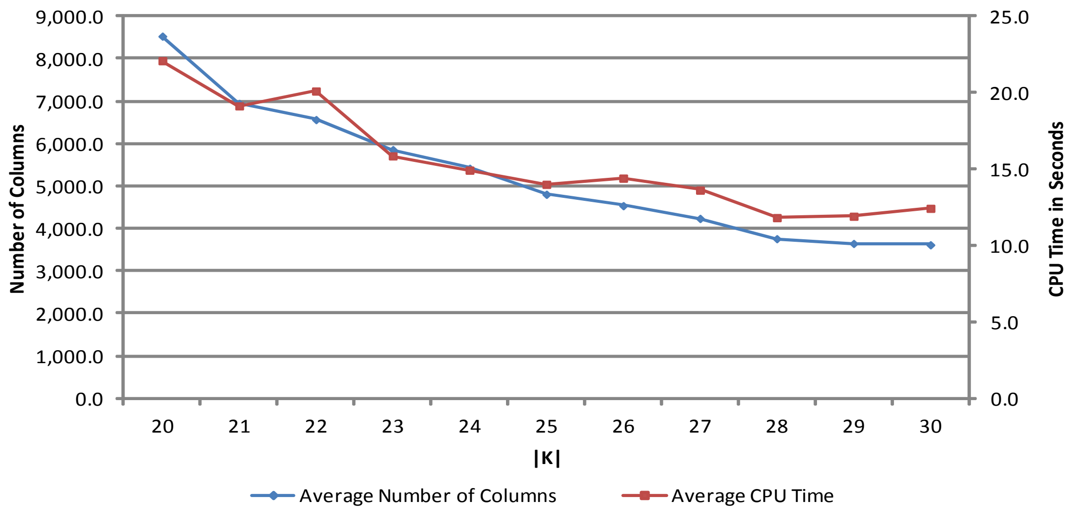

Figure 2.6: Average CPU time and average number of columns in function of $|K|$.

of 15 minutes and takes values from $H_1 = \{15, 30, 45, ..., 105, 120\}$ in the low delay case and from $H_2 = \{15, 30, 45, ..., 225, 240\}$ in the high delay case. For time of day dependent delay, the study of more than 6 million flights in the database of BTS in 2013 [Silver] reveals that the average flight delay exceeds 15 minutes at approximately 12:30pm, peaks at 6pm, and drops below 15 minutes around 9:30pm. We set the time between 12:30pm and 9:30pm as a high delay period and generate h_{ij} from from H_2 if the arrival time of flight i and the departure time of flight j is between 12:30pm and 9:30pm. Otherwise, h_{ij} takes value from H_1. At this point we have 60 instances that we solve using the proposed algorithm to obtain 60 optimal schedules. While h_{ij} determines the range of deviation on arc (i, j), μ_{ij} is the actual deviation used in the simulated scenario. It is generated based on either the Uniform distribution or the Triangular distribution with three levels of skewness. To determine whether connection (i, j) is disrupted, we generate a number from $\{0, 1\}$ with equal probabilities, where 1 means the connection is disrupted and 0 means it is not. For disrupted connection (i, j), the actual delay μ_{ij} is randomly generated on the interval $(0, h_{ij}]$ following either a Uniform or a Triangular distribution. The Uniform distribution is referred to as **Dist1**. For the Triangular distribution on interval $(0, h_{ij}]$, the skewness is determined by the mode of the distribution set to $0.625h_{ij}$, $0.75h_{ij}$, and $0.875h_{ij}$, respectively. A higher mode leads to a distribution skewed more to the left. We refer to the three triangular distributions with modes $0.625h_{ij}$, $0.75h_{ij}$, and $0.875h_{ij}$ as **Dist2**, **Dist3**, and **Dist4**, respectively. The 60 schedules are simulated 100 times each with each of the 4 distributions leading to 24000 runs.

We calculate two statistics to evaluate the quality of the solutions. Total cost is the sum of the pairing cost over 20 crews. Cumulative delay is the sum of the cumulative delay over all connections in the solution. For example, if connection arc (i, j) is selected by pairing p, the cumulative delay on the arc is μ_{ij} plus the cumulative delays on all preceding connections in the pairing. Figure 2.7 plots the average total cost across 100 scenarios where each line corresponds to one of the four distributions under low h_{ij}. Figures 2.8 and 2.9 show similar plots under high h_{ij} and time of day dependent h_{ij}, respectively. Figures 2.10, 2.11 and 2.12 plot the average cumulative delay under low h_{ij}, high h_{ij} and time of day dependent h_{ij}, respectively. Figures 2.13 and 2.14 plot the percent savings in average total cost and percent change in average cumulative delay with respect to $\Gamma = 0$, respectively. When $\Gamma = 0$, [P1] reduces to the deterministic crew pairing problem, and we refer to its solution as the nominal solution. Table 2.8 reports on the average, minimum, and maximum percent savings (Avg_Sav, Min_Sav, Max_Sav) on total cost of the robust solution over the nominal solution, and the average decrease, minimum, and maximum percent change (Avg_Dec, Min_Chg, Max_Chg) of cumulative delay of robust solution over nominal solution.

Except for one instance with protection level $\Gamma = 1$ under Dist 3, the robust solution always achieves lower average total cost than the nominal solution. Recall that cost stands for the duration of crew pairings. Therefore, robust pairings are always shorter on average when delay occurs. This holds for the four distributions of actual delay and under low, high, and time of day dependent delay. Average simulated cost shows a stable value when Γ is high enough, 14 and up in this case. This could be related to the number of flights in a crew paiting. When crew pairings are around 14 flights long, setting Γ higher than 14 will not increase the randomness of the data. In other words, the number of connections in a crew pairing that may experience delay is capped by Γ as well as by the number of flights in the schedule. Robust solutions provide up to 2% savings in total costs for high h_{ij}, 1% for low h_{ij}, and 1.3% for time of day dependent h_{ij}.

Average cumulative delay increases for small values of Γ and decreases as Γ increases. The decrease in delay becomes significant for high values of Γ. This says that schedules built assuming more randomness experience smaller propagated delay than a schedule built assuming no or low randomness. The decrease in average cumulative delay may be significant and is up to 43% for high h_{ij}, up to 27% for time of day dependent h_{ij}, and up to 29% for low h_{ij}. In practice since Γ is a parameter in the robust model, one could solve the robust model for several values of Γ and further evaluate the optimal schedules using historical data.

Comparing the twelve lines in Figures 2.13 and 2.14, there is no significant difference in

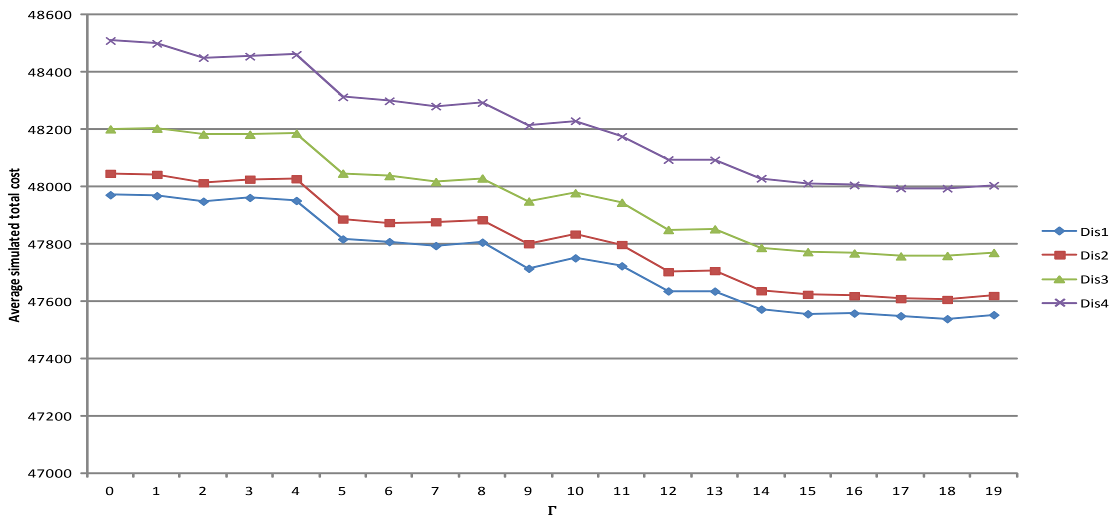

Figure 2.7: Low h_{ij}: Average total cost.

the shapes which indicates that the distribution and the skewness of delay does not have an impact on the robustness of the solution. When delay is significant (i.e., under high delay), the savings in total cost are achieved even with low levels of randomness determined by Γ. While for the case of low delay h_{ij} and time of day dependent delay h_{ij}, the savings become more significant when randomness is high (i.e., high Γ).

2.8. Conclusion

In this chapter, we presented a new robust formulation for the crew pairing problem. Unlike previous work, we assumed that flight and connection times are random and vary within an interval with no additional probabilistic assumptions. Robustness is achieved by protecting against infeasibility with a specified protection level and by finding minimum

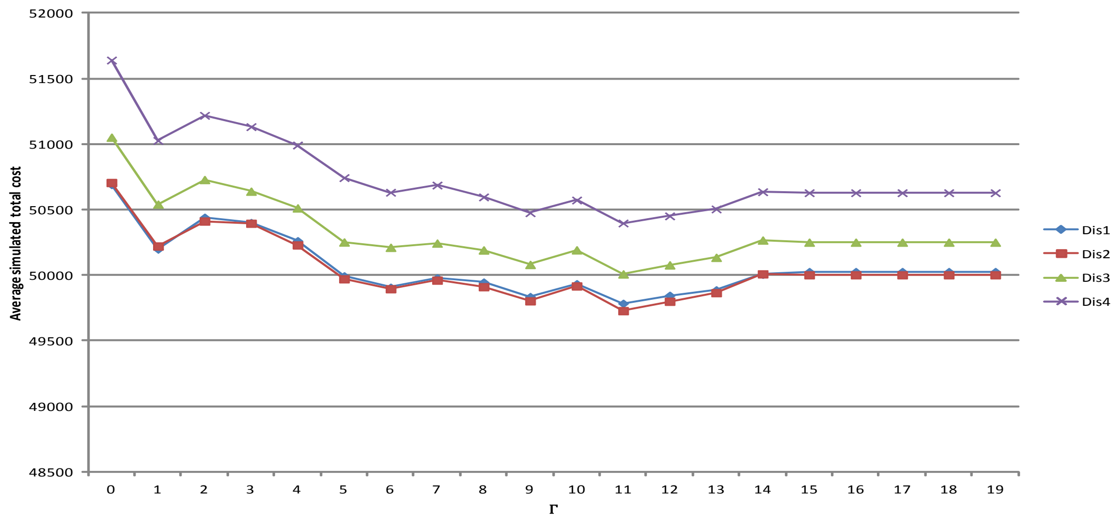

Figure 2.8: High h_{ij}: Average total cost.

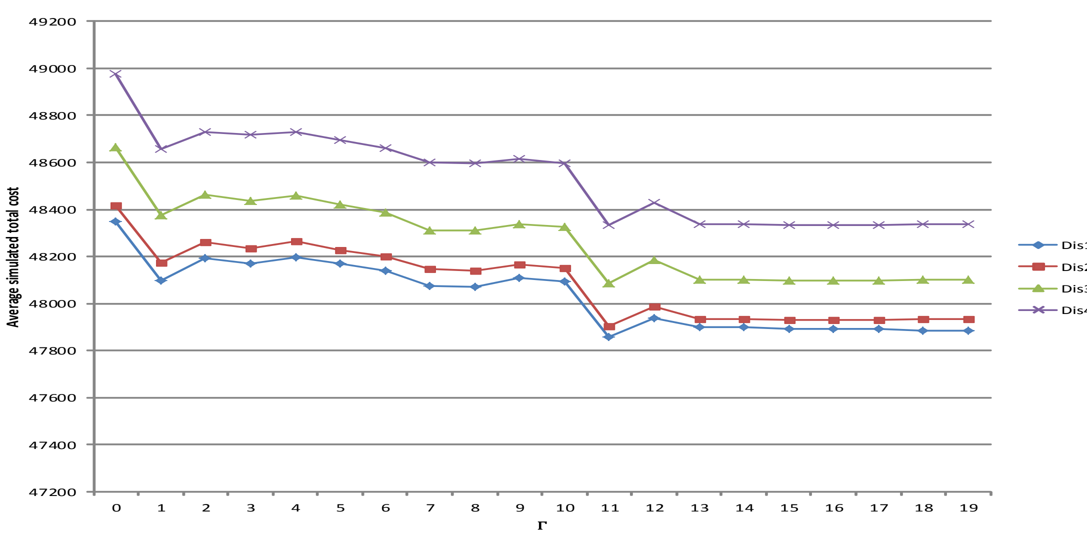

Figure 2.9: Time of day dependent h_{ij}: Average total cost.

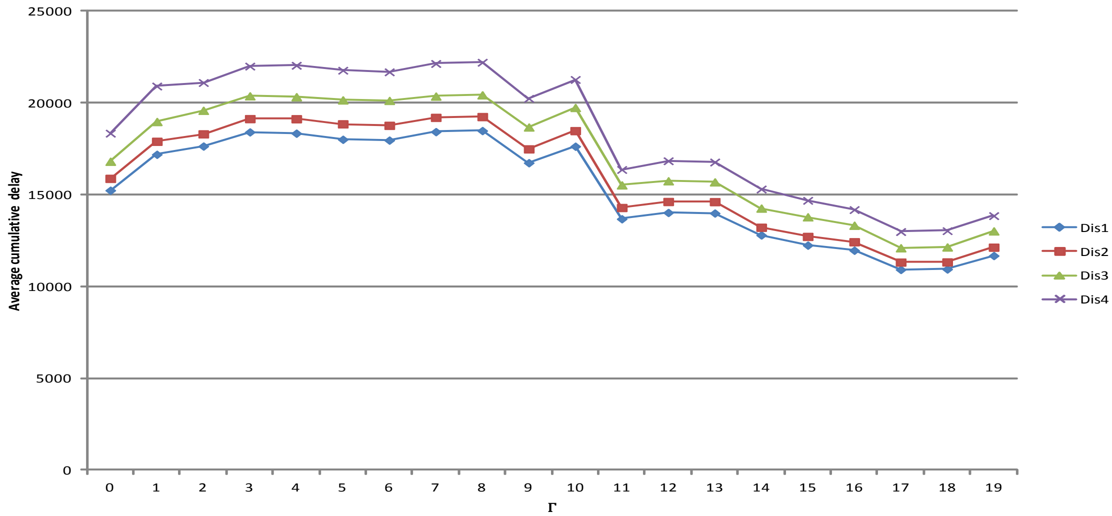

Figure 2.10: Low h_{ij}: Average cumulative delay.

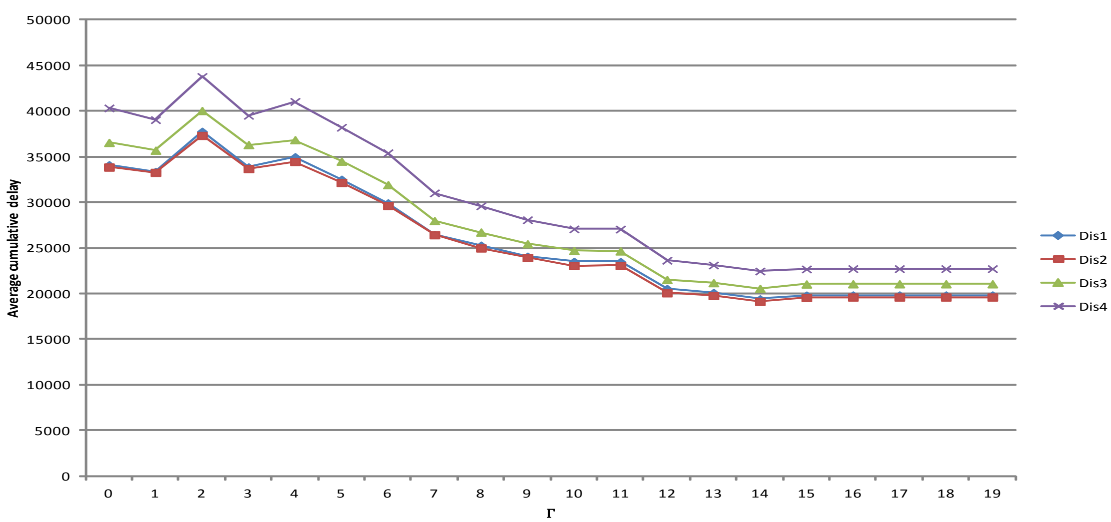

Figure 2.11: High h_{ij}: Average cumulative delay.

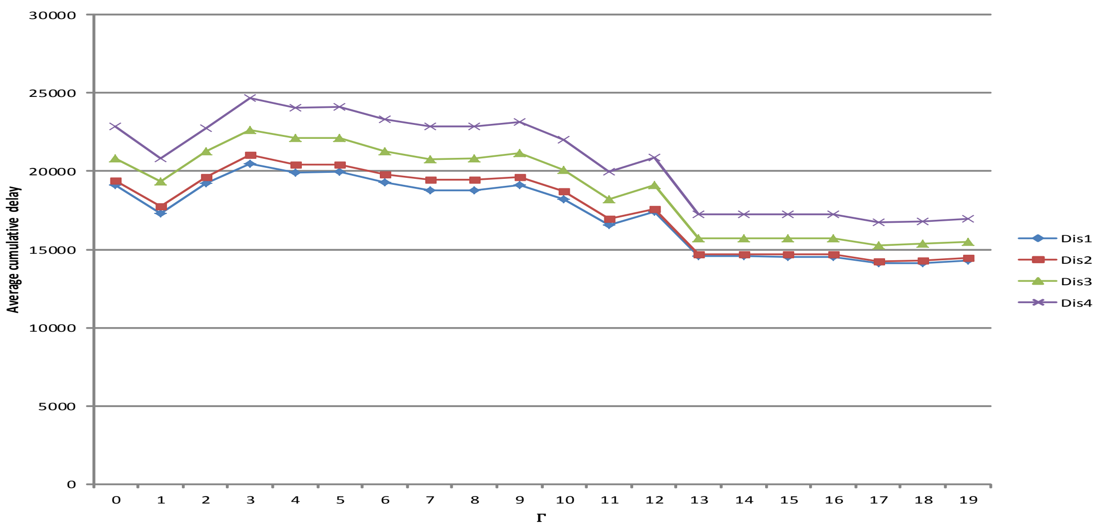

Figure 2.12: Time of day dependent h_{ij}: Average cumulative delay.

cost solutions in the worst case given that protection level. The resulting robust model is an integer, multi-commodity flow problem with nonlinear robust terms in the objective function and in the resource constraints. After applying Lagrangian relaxation, we defined the robust shortest path problem with resource constraints as subproblem and proved that it may be solved as a series of linear resource constrained shortest path problems. We used dual information to find feasible integer solutions after calculating the Lagrangian bound. Tests on industry instances showed that the solution methodology finds optimal or near-optimal solutions in reasonable time with modest computational requirements. Based on extensive simulations of robust and deterministic schedules, we found that robust schedules lead to lower total cost when delays occur and to significant savings in cumulative delays experienced by passengers.

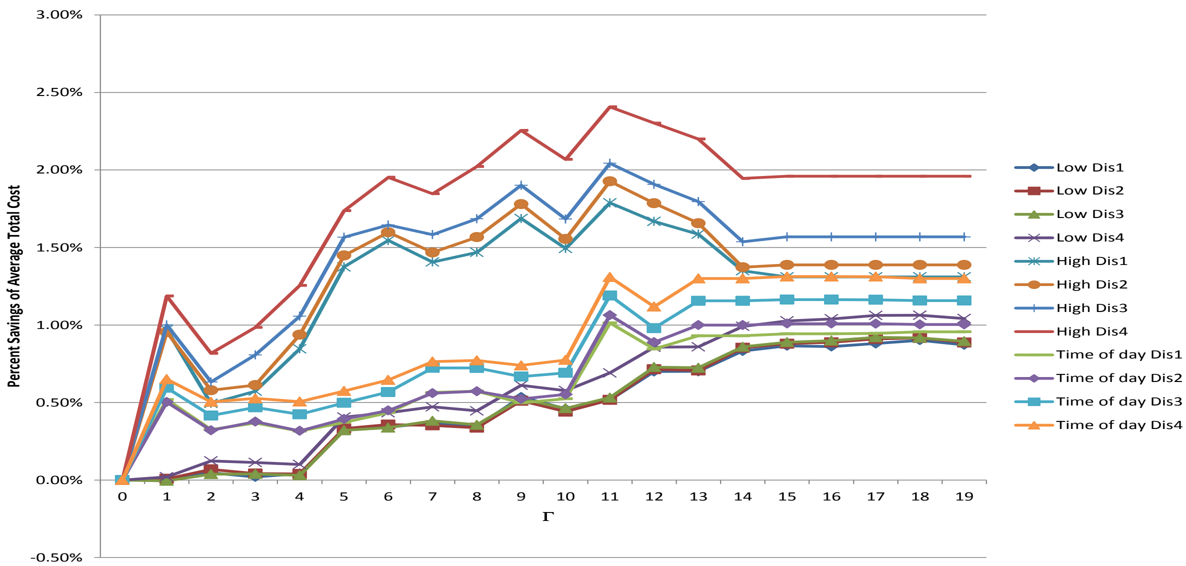

Figure 2.13: Percent savings of total cost under 4 distributions.

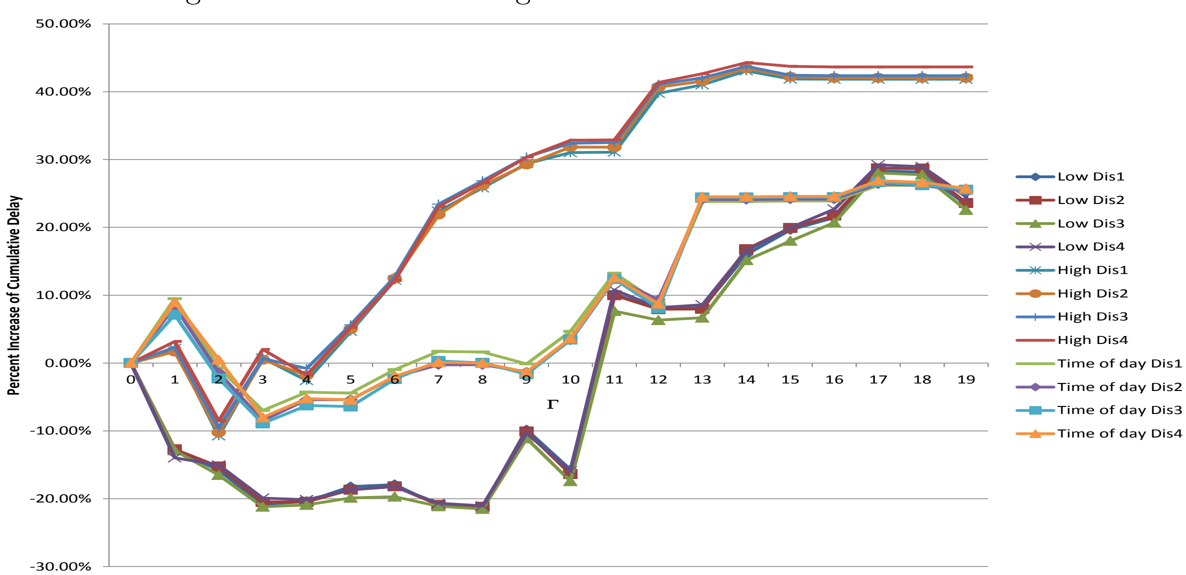

Figure 2.14: Percent change of average cumulative delay under 4 distributions.

Table 2.5: Test results of instances 1 to 40 based on real data.

$\|K\|$	Γ	**LRB**	**IP**	**Gap**	**Inc**	**NbCol**	**MST**	**SubT**	**Tot**	**IPT**
20	1	47165.0	47185.0	0.04	0.00	2887	4.6	0.6	6.1	18.03
	2	48382.5	48385.0	0.01	2.54	4744	11.2	0.8	13.5	20.19
	3	49293.3	49340.0	0.09	4.57	6020	11.7	0.9	14.5	80.08
	4	49995.0	50000.0	0.01	5.97	6901	12.3	0.9	15.2	82.85
	5	50605.0	50605.0	0.00	7.25	7723	16.2	1.1	19.6	5.71
	6	51153.3	51180.0	0.05	8.47	9052	19.0	1.2	22.5	55.76
	7	51655.0	51655.0	0.00	9.47	8403	19.5	1.3	22.9	7.10
	8	52103.6	52105.0	0.00	10.43	8985	17.2	1.3	20.9	26.54
	9	52478.3	52480.0	0.00	11.22	9167	20.4	1.4	24.4	50.54
	10	52797.5	52855.0	0.11	12.02	11252	24.6	1.7	29.3	106.56
	11	53077.5	53120.0	0.08	12.58	9925	19.3	1.5	23.5	79.32
	12	53293.3	53300.0	0.01	12.96	9599	18.1	1.5	22.6	45.46
	13	53440.0	53440.0	0.00	13.26	9977	26.6	1.8	30.8	7.24
	14	53555.0	53555.0	0.00	13.50	9357	22.4	1.6	26.3	7.89
	15	53640.0	53640.0	0.00	13.68	9513	19.9	1.6	23.9	35.51
	16	53685.0	53685.0	0.00	13.78	11196	28.1	2.0	32.7	8.32
	17	53715.0	53715.0	0.00	13.84	6582	12.7	1.2	15.8	6.22
	18	53745.0	53745.0	0.00	13.90	7848	16.6	1.5	20.3	6.49
	19	53745.0	53745.0	0.00	13.90	13030	29.7	2.2	35.3	13.34
21	1	46630.0	46670.0	0.09	0.00	2900	5.7	0.5	7.1	40.07
	2	47910.0	47910.0	0.00	2.66	4881	13.2	0.9	15.8	3.41
	3	48828.8	48855.0	0.05	4.68	6753	17.3	1.0	20.4	114.73
	4	49515.0	49515.0	0.00	6.10	6991	17.4	1.0	20.6	5.87
	5	50100.0	50100.0	0.00	7.35	9151	21.7	1.3	26.3	7.75
	10	52230.0	52260.0	0.06	11.98	8417	14.5	1.4	18.2	64.04
	15	53000.0	53005.0	0.01	13.57	9570	21.2	1.6	25.7	39.40
22	1	46110.0	46110.0	0.00	0.00	3165	7.9	0.6	9.7	12.42
	2	47455.0	47470.0	0.03	2.95	3797	10.5	0.7	12.6	21.52
	3	48370.0	48370.0	0.00	4.90	6764	20.3	1.0	23.7	5.89
	4	49060.0	49060.0	0.00	6.40	5728	11.8	1.0	14.7	15.06
	5	49645.0	49645.0	0.00	7.67	7393	18.2	1.0	23.7	21.67
	10	51675.0	51695.0	0.04	12.11	8833	20.3	1.3	24.0	82.76
	15	52390.0	52395.0	0.01	13.63	10303	28.1	1.8	32.5	40.18
23	1	45670.0	45670.0	0.00	0.00	2733	7.2	0.6	8.7	2.11
	2	47050.0	47050.0	0.00	3.02	3840	9.8	0.7	11.8	2.82
	3	47982.6	48010.0	0.06	5.12	4838	10.7	0.8	13.1	43.62
	4	48650.0	48670.0	0.04	6.57	5619	11.3	0.8	13.7	51.99
	5	49230.0	49230.0	0.00	7.80	6854	14.9	1.0	18.0	59.43
	10	51140.0	51180.0	0.08	12.06	7628	16.1	1.2	19.6	63.07
	15	51787.5	51795.0	0.01	13.41	9524	22.0	1.6	26.2	42.07

Table 2.6: Test results of instances 41 to 90 based on real data.

$\lvert K \rvert$	Γ	**LRB**	**IP**	**Gap**	**Inc**	**NbCol**	**MST**	**SubT**	**Tot**	**IPT**
24	1	45255.0	45260.0	0.01	0.00	2688	6.0	0.6	7.7	14.83
	2	46665.0	46665.0	0.00	3.10	3730	11.7	0.7	13.7	8.87
	3	47607.1	47620.0	0.03	5.21	4369	10.8	0.7	13.0	32.47
	4	48260.0	48285.0	0.05	6.68	6012	12.7	0.9	15.6	40.92
	5	48820.0	48855.0	0.07	7.94	6553	12.9	1.0	15.9	62.80
	10	50622.5	50635.0	0.02	11.88	6087	11.9	1.0	14.7	50.82
	15	51195.0	51195.0	0.00	13.11	8520	20.2	1.5	24.1	6.55
25	1	44847.5	44870.0	0.05	0.00	2673	6.9	0.6	8.4	19.01
	2	46330.0	46330.0	0.00	3.25	3546	9.9	0.6	11.7	2.82
	3	47235.0	47235.0	0.00	5.27	3826	9.3	0.6	11.2	3.45
	4	47888.3	47980.0	0.19	6.93	5699	13.4	0.8	15.9	259.27
	5	48436.7	48490.0	0.11	8.07	5562	12.5	0.9	15.1	62.62
	10	50125.0	50125.0	0.00	11.71	5920	14.9	1.0	17.7	6.44
	15	50620.0	50620.0	0.00	12.81	6442	14.8	1.3	18.1	0.58
26	1	44488.8	44520.0	0.07	0.00	2229	5.6	0.5	6.9	19.96
	2	46015.0	46015.0	0.00	3.36	3664	13.0	0.7	15.0	2.88
	3	46915.0	46915.0	0.00	5.38	4043	12.5	0.7	14.5	3.60
	4	47571.3	47705.0	0.28	7.15	5082	11.9	0.8	14.3	102.50
	5	48126.7	48185.0	0.12	8.23	5226	12.7	0.8	15.3	62.36
	10	49850.0	49850.0	0.00	11.97	5252	12.4	0.9	15.0	27.60
	15	50345.0	50345.0	0.00	13.08	6295	16.6	1.3	19.9	5.95
27	1	44137.5	44190.0	0.12	0.00	2425	7.0	0.5	8.3	32.39
	2	45715.0	45715.0	0.00	3.45	2862	7.1	0.5	8.7	2.29
	3	46645.0	46645.0	0.00	5.56	3910	12.2	0.7	14.3	9.93
	4	47287.0	47290.0	0.01	7.02	4856	13.4	0.8	15.8	24.07
	5	47858.3	47890.0	0.07	8.37	4550	9.2	0.7	11.5	30.87
	10	49587.5	49590.0	0.01	12.22	6120	18.8	1.1	21.9	25.41
	15	50095.0	50095.0	0.00	13.36	4963	12.2	1.1	14.9	3.84
28	1	43805.0	43805.0	0.00	0.00	2540	8.2	0.5	9.7	5.24
	2	45445.0	45445.0	0.00	3.74	2626	6.4	0.5	7.9	2.16
	3	46412.5	46420.0	0.02	5.97	3936	13.2	0.7	15.3	19.48
	4	47040.0	47125.0	0.18	7.58	3790	9.7	0.7	11.9	30.48
	5	47625.8	47685.0	0.12	8.86	4355	9.9	0.7	12.1	43.05
	10	49351.7	49365.0	0.03	12.69	5132	12.7	1.0	15.4	30.96
	15	49880.0	49880.0	0.00	13.87	3965	8.2	0.9	10.5	3.40
29	1	43675.0	43705.0	0.07	0.00	2473	7.3	0.5	8.8	10.14
	2	45280.0	45280.0	0.00	3.60	2722	7.4	0.5	8.9	1.86
	3	46195.0	46195.0	0.00	5.70	3258	10.2	0.5	11.9	3.00
	4	46870.0	46870.0	0.00	7.24	4321	13.5	0.7	15.8	4.06
	5	47460.0	47460.0	0.00	8.59	4141	10.6	0.7	12.8	11.60
	10	49140.0	49140.0	0.00	12.44	4152	9.7	0.8	11.9	3.86
	15	49685.0	49685.0	0.00	13.68	4492	10.7	1.0	13.3	4.08
30	1	43585.0	43585.0	0.00	0.00	2175	6.4	0.4	7.7	1.72
	2	45185.0	45185.0	0.00	3.67	2986	9.3	0.5	10.9	2.34
	3	46070.0	46070.0	0.00	5.70	3542	11.1	0.6	13.0	3.11
	4	46748.3	46770.0	0.05	7.31	3394	8.7	0.6	10.6	6.29
	5	47310.0	47320.0	0.02	8.57	3937	9.3	0.6	11.4	17.71
	10	48950.0	48950.0	0.00	12.31	4512	13.3	0.7	15.7	3.57
	15	49520.0	49525.0	0.01	13.63	4795	14.6	1.0	17.7	21.13

Table 2.7: Average statistics over all instances.

$\|K\|$	**LRB**	**IP**	**Gap**	**Inc**	**NbCol**	**MST**	**SubT**	**Tot**	**IPT**
20	51975.0	51986.1	0.02	10.2	8534.8	18.4	1.4	22.1	34.9
21	49744.8	49759.3	0.03	6.6	6951.9	15.8	1.1	19.1	39.3
22	49243.6	49249.3	0.01	0.0	6569.0	16.7	1.1	20.1	28.5
23	48787.2	48800.7	0.03	0.0	5862.3	13.2	1.0	15.9	37.9
24	48346.4	48359.3	0.03	6.8	5422.7	12.3	0.9	14.9	31.0
25	47926.1	47950.0	0.05	6.9	4809.7	11.7	0.8	14.0	50.6
26	47616.0	47647.9	0.07	7.0	4541.6	12.1	0.8	14.4	32.1
27	47332.2	47345.0	0.03	7.1	4240.9	11.4	0.8	13.6	18.4
28	47080.0	47103.6	0.05	7.5	3763.4	9.8	0.7	11.8	19.3
29	46900.7	46905.0	0.01	7.3	3651.3	9.9	0.7	11.9	5.5
30	46766.9	46772.1	0.01	7.3	3620.1	10.4	0.6	12.4	8.0
Average	48338.1	48352.6	0.03	6.1	5269.8	12.9	0.9	15.5	27.8

Table 2.8: Comparison of nominal and robust solutions.

		Low h_{ij}			High h_{ij}			Time of day dependent h_{ij}		
		Avg_Sav	**Min_Sav, Γ**	**Max_Sav, Γ**	**Avg_Sav**	**Min_Sav, Γ**	**Max_Sav,Γ**	**Avg_Sav**	**Min_Sav, Γ**	**Max_Sav,Γ**
Total cost	Dis1	0.51	0.01, 1	0.90, 18	1.03	0.49, 2	1.79, 11	0.68	0.32, 4	1.01, 11
	Dis2	0.51	0.01, 1	0.92, 18	1.38	0.58, 2	1.93, 11	0.71	0.32, 4	1.06, 11
	Dis3	0.52	-0.01, 1	0.92, 18	1.51	0.63, 2	2.04, 11	0.85	0.42, 4	1.19, 11
	Dis4	0.63	0.02, 1	1.06, 18	1.83	0.82, 2	2.41, 11	0.95	0.50, 2	1.31, 17
		Low h_{ij}			High h_{ij}					
		Avg_Dec	**Min_Chg, Γ**	**Max_Chg, Γ**	**Avg_Dec**	**Min_Chg, Γ**	**Max_Chg, Γ**	**Avg_Dec**	**Min_Chg, Γ**	**Max_Chg, Γ**
Cumulative delay	Dis1	-0.56	-21.34, 8	28.32, 17	25.22	-10.47, 2	43.08, 14	10.28	-7.01, 3	26.22, 18
	Dis2	-0.48	-21.19, 8	28.71, 17	25.51	-10.24, 2	43.45, 14	9.65	-8.50, 3	26.40, 17
	Dis3	-1.55	-21.52, 8	28.02, 17	26.07	-9.56, 2	43.74, 14	9.45	-8.89, 3	26.64, 17
	Dis4	-0.24	-21.05, 8	29.21, 17	26.56	-8.54, 2	44.29, 14	10.00	-8.02, 3	26.88, 17

Chapter 3

Robust shortest path problem with resource constraints

This chapter is based on the paper [Lu and Gzara, 2014a] submitted to Mathematical Programming.

3.1. Motivation and objective

The shortest path problem has been widely studied in operations research because of its theoretical and practical relevance. While the basic shortest path problem (SPP) is easy to solve and lies at the heart of network flows, extensions with additional restrictions on paths present challenges to solve. Shortest paths appear frequently in practice whenever there is interest to send flow from an origin to a destination in applications as diverse as vehicle routing and scheduling, airline operations planning, transportation, and telecommunications network planning. Shortest paths arise frequently as subproblems when solving optimization problems on networks by decomposition techniques like Lagrangian relaxation, column generation, and Benders decomposition. Several extensions of the shortest path problem have been studied. We refer interested readers to the survey by Desaulniers et al. [2005] for classification, modeling issues and solution methodologies for SPP with additional constraints. In particular, in the deterministic shortest path problem with resource constraints (SPPRC) each arc consumes a given amount of a given resource with limited availability. The deterministic SPPRC finds a minimum cost path feasible with respect to resource consumption constraints.

In this chapter, we study the robust SPPRC by introducing uncertain cost and resource consumption on arcs. More specifically, the uncertain parameters are defined by an interval of uncertainty where the realization of a parameter may take any value within the corresponding interval. The lower limit of the interval of uncertainty is referred to as the nominal value of the parameter and the deterministic SPPRC arises when all random parameters are at their nominal values. We present a robust model to find the minimum cost robust path that remains feasible when a subset of random parameters deviate from their nominal values. We first review the literature relevant to the deterministic SPPRC, we then review the relevant literature on robust optimization and summarize the contributions in this chapter.

The rest of the chapter is organized as follows. Section 3.5 defines the robust SPPRC and presents the robust formulation. The graph reduction techniques for the robust SPPRC are presented in Section 3.5.1. Section 3.6 shows an equivalent MIP reformulation, then an exact sequential algorithm is proven applicable to the problem. Section 3.7 presents the new dominance rules and the modified label-setting algorithm. Section 3.8 reports on the computational testing and Section 3.9 concludes the chapter.

3.2. Literature review on the deterministic SPPRC

The solution methodologies for the deterministic SPPRC include Lagrangian relaxation, labeling algorithm, and heuristics. Lagrangian relaxation is used by Handler and Zang [1980] who relax the resource constraint, resulting in a shortest path problem with Lagrangian length as subproblem. If the dual gap after solving the Lagrangian dual problem is nonzero, a kth-shortest path problem is solved using the algorithm of Yen [1971]. The parameter k is initialized at 2 and is increased gradually until the gap closes. Santos et al. [2007] improve the algorithm of Handler and Zang [1980] with a refined search direction based on the tightness of the resource limit. Beasley and Christofides [1989] use similar relaxation and close the dual gap through branch-and-bound. The branching scheme starts at the origin and builds a partial path. Each branch corresponds to a SPPRC from the end node of current partial path to the destination. The next branch to explore is determined based on the Lagrangian subproblem solution. Beasley and Christofides [1989] also propose graph reduction techniques based on the minimum consumption of each resource and the minimum Lagrangian length between a pair of nodes. Borndörfer et al. [2001] study a similar problem with an additional goal consumption for each resource and penalize any deviation from the goal.

Dynamic programming formulations leading to labeling algorithms form another exact methodology for the deterministic SPPRC. Desrochers [1988] proposes a label correcting algorithm that extends Pareto-optimal partial paths along the graph from origin to destination. Dumitrescu and Boland [2003] improve and compare a set of algorithms including Lagrangian relaxation, label-setting, and heuristics with a cost scaling technique. They improve the preprocessing by iteratively using the techniques in Beasley and Christofides [1989] until no more reduction was possible. For the label-setting algorithm, they strengthen the feasibility check by considering for each resource the sum of the incurred resource consumption of a partial path and the minimum resource requirement from the end of a partial path to the destination. Dumitrescu and Boland [2003] propose an exact weight-scaling algorithm that repeatedly performs the label-setting algorithm on a graph with resource consumptions on arcs and scaled resource upper bounds. During the process, the minimum cost path provides a lower bound on the original problem even if it is infeasible. As the scaled values ultimately converge to their original values, the algorithm is guaranteed to find an optimal solution. Lozano and Medaglia [2013] modify the label-setting algorithm by proposing three rules to manage and limit the size of the partial paths that are stored at each node.

Zhu and Wilhelm [2012] propose a three-stage algorithm to solve SPPRC on an acyclic graph. They reduce the graph using a technique similar to that of Dumitrescu and Boland [2003] in the first stage. Then they expand it to a graph corresponding to an unconstrained shortest path problem, and solve as SPP. Since the extended graph from one such transformation is reusable for different arc cost, the approach is suitable for column generation where cost on arc changes in the process.

Heuristic algorithms for the deterministic SPPRC are based on cost scaling. Hassin [1992] propose a polynomial approximation scheme for SPPRC with nonnegative integer costs on arcs and a single resource constraint. The algorithm starts from an upper bound UB and a lower bound LB and determines whether a resource feasible path with cost no more than $\sqrt{UB * LB}$ exists. This is done approximately by scaling arc costs and solving the scaled problem with a label-setting algorithm. With approximation factor ϵ, UB or LB is updated as $\sqrt{UB * LB}(1+\epsilon)$ or $\sqrt{UB * LB}$ depending on whether such path exists. The algorithm is suggested to terminate when $UB/LB \leq 2$. Lorenz and Raz [2001] propose a similar scaling approach and an initialization of UB that guarantees $UB/LB \leq |A|$.

A comparison of the computational results from the literature shows a dominance of the labeling algorithm over the other approaches. Beasley and Christofides [1989] are able to solve an instance with 500 nodes and 4868 arcs in 26.3 seconds while Lozano and Medaglia [2013] solve it in 0.013 seconds. Lozano and Medaglia [2013] compared their algorithm with Santos et al. [2007] on a set of instances. Their algorithm is able to solve the largest instance with 40000 nodes and 800000 arcs in 14.4 seconds, a speed up of 33.49 times against Santos et al. [2007]. Dumitrescu and Boland [2003] compared their modified label-setting algorithm against the heuristics by Hassin [1992] and Lorenz and Raz [2001]. Among the solved instances by the modified label-setting algorithm and the heuristics, the heuristics were only better in a few large instances. However, the modified label-setting algorithm solves more instances. Moreover, for the unsolved instances, Dumitrescu and Boland [2003] further refine the label pruning step by considering a Lagrangian lower bound on SPPRC from the resident node of a label to destination node. This improvement enabled the modified labeling algorithm to solve all instances.

3.3. Literature review on robust SPPRC and related problems

The literature on robust SPPRC is scant but a few studies on robust vehicle routing problems exit. More specifically, Sungur et al. [2008] are the first to study a robust capacitated vehicle routing problem (CVRP) with uncertain demands. They focus on three types of demand uncertainty, including the convex hull, box, and ellipsoid. However, their models assume worst case scenario. They formulate the corresponding models based on Miller-Tucker-Zemlin formulation with an instance added to the right hand side of the subtour elimination constraints, and identify conditions under which these problems can be solved by exact algorithms for the deterministic CVRP. Although they mention the possibility of introducing the cardinality constrained uncertainty to the vehicle routing problem with time windows for both travel time and arc cost under the framework of [Bertsimas and Sim, 2003], the solution methodology suggested is an extension of the sequential approach for robust combinatorial problems proposed by Bertsimas and Sim [2003]. A robust vehicle routing problem with time windows is recently studied by Agra et al. [2013]. They provide two formulations based on resource inequalities and path inequalities, and prove that it is sufficient to consider a subset of the extreme points of the uncertainty polytope, which is composed of all possible scenarios. When the uncertainty polytope falls into the framework of Bertsimas and Sim [2003], i.e. uncertainty is represented by cardinality constrained support, they further reduce the subset of the extreme points of the uncertainty polytope. The resource inequality formulation is solved implicitly by a column-and-row generation procedure and the path inequality formulation is solved by a cutting plane algorithm. A robust

CVRP with demand uncertainty is studied by Gounaris et al. [2013]. The demand realizations consist of a polyhedron, referred to as the support of the demands, and the model generates a set of routes that minimize the total traveling cost and remain feasible for any demand realization in the support. They provide robust counterparts of four deterministic CVRP formulations, including two-index vehicle flow formulation, Miller-Tucker-Zemlin formulation, commodity flow formulation, and vehicle assignment formulation. Eggenberg et al. [2007] study a robust shortest path problem outside the framework of robust optimization. They propose a model that sums the worst-case cost and the cost of traversing back to recover an infeasible path, and finds a solution that minimizes the cost difference between the worst case scenario and the best case scenario.

3.4. Contributions

We present the robust SPPRC with multiple resource constraints. The resource consumption and arc costs are uncertain and are given by an interval with no predefined probability distribution. The robustness is controlled by a protection level that limits the number of uncertain parameters in the objective function and in the robust constraints. We propose a set of preprocessing techniques based on Beasley and Christofides [1989] and Dumitrescu and Boland [2003], tailored for the robust version of the problem. More specifically, for graph reduction based on resource consumption and Lagrangian relaxation, we strengthen the inequalities to remove unnecessary arcs and nodes by considering the lower bound on the total variation against the deterministic case. Also, any feasible solution to the original problem during the process provides an upper bound and is used to strengthen the inequalities based on Lagrangian relaxation.

As mentioned by Sungur et al. [2008], the solution methodology that solves a robust combinatorial optimization problem by solving a sequence of nominal deterministic problems in Bertsimas and Sim [2003] is not restricted to the problem with uncertainty in objective coefficients. In fact, it can be extended to the case where both objective and constraint coefficients are uncertain. In this chapter, we show that when multiple resource constraints are subject to uncertainty, a sequential algorithm that solves a series of deterministic SPPRC may be applied to solve the robust SPPRC.

Finally, we derive new dominance rules and propose a modified label-setting algorithm for the robust SPPRC. When comparing two partial paths, one partial path dominated by the other one in a deterministic setting might lead to a better complete path in a robust context. This is because for each resource, the total resource consumption in a robust context not only depends on the deterministic data but also on a subset of the resource variations on the complete path, where the specified protection level determines the size of the subset. As a result, the variation information of a partial path is not enough to make a decision about dominance. We propose an extended dominance rule to compare two partial paths under the robust framework by developing for each resource and each partial path an upper bound on the sum of resource variations in a subset with determined size.

We test the sequential algorithm, the modified label-setting algorithm and direct solution of the MIP reformulation using CPLEX 12.4 on instances based on those in Beasley and Christofides [1989]. The results show the superiority of the modified label-setting algorithm over the sequential algorithm and direct solution of the MIP reformulation. Also, the results show that the scalability of the sequential algorithm is poor when the number of resources or the number of distinct variations increase.

3.5. The robust SPPRC

In this section, we first present a new formulation for the robust SPPRC. Then we generalize the preprocessing techniques of the deterministic SPPRC to reduce the graph for the robust SPPRC.

Let us define a network $G = (N, A)$, where N is the set of nodes indexed by $i = 1, ..., |N|$ and A is the set of arcs represented by $(i, j) \in A$. Let $o \in N$ denote the origin, and $d \in N$ denote the destination. Traveling in the network consumes a number of resources. Each resource has a limit on available capacity, representing the maximum amount of the resource that can be used on a feasible path. The cost of an arc is represented by a special resource with infinite capacity. Let R denote the set of resources indexed by $r = 1, ..., |R|$, where resource $r = 1$ is used for the cost. For each arc $(i, j) \in A$, t_{ijr} is the minimum consumption of resource r on arc (i, j), also referred to as nominal consumption. h_{ijr} is the maximum deviation from the nominal consumption. A realization of the consumption of resource r on (i, j), denoted by $\hat{t}_{ijr}$ is assumed to vary randomly in the interval $[t_{ijr}, t_{ijr} + h_{ijr}]$. When arc (i, j) shows no uncertainty in the consumption of resource r, h_{ijr} is set to 0. The limit on resource r is denoted by B_r. The decision variable y_{ij} takes value 1 if arc (i, j) is selected in the shortest path, and 0 otherwise. To protect the solution from uncertainty, a specified parameter Γ_r, called protection level, is defined for each resources r. It allows at most Γ_r arcs to deviate from the nominal values of resource r for any path at any given time. Given a path p involving arcs $S^p \subseteq A$, the worst deviation from the nominal consumption of resource r is determined by $\max_{\{S_r | S_r \subseteq S^p, |S_r| \leq \Gamma_r\}} \sum_{(i,j) \in S_r} h_{ijr}$. The consumption by path p of resource r given Γ_r is calculated as $\sum_{(i,j) \in S^p} t_{ijr} + \max_{\{S_r | S_r \subseteq S^p, |S_r| \leq \Gamma_r\}} \sum_{(i,j) \in S_r} h_{ijr}$ and is referred to as the robust consumption of resource r. As Γ_r increases, more variations are considered in the robust consumption of resource r. The robust consumption of resource $r = 1$ is defined as the robust cost. A path is considered robust if its robust consumption of resource r is no more than B_r at any node in the path for each resource $r \in R \setminus \{1\}$. The robust SPPRC is modeled as follows:

[P]

$$\min \sum_{(i,j) \in A} t_{ij1} y_{ij} + \max_{\{S_1 | S_1 \subseteq A, |S_1| \leq \Gamma_1\}} \sum_{(i,j) \in S_1} h_{ij1} y_{ij} \tag{3.1}$$

$$\text{s.t.} \sum_{j:(o,j) \in A} y_{o,j} = 1 \tag{3.2}$$

$$\sum_{i:(i,j) \in A} y_{ij} - \sum_{i:(j,i) \in A} y_{ji} = 0 \quad \forall j \in N \setminus \{o, d\} \tag{3.3}$$

$$\sum_{i:(i,d) \in A} y_{i,d} = 1 \tag{3.4}$$

$$\sum_{(i,j) \in A} t_{ijr} y_{ij} + \max_{\{S_r | S_r \subseteq A, |S_r| \leq \Gamma_r\}} \sum_{(i,j) \in S_r} h_{ijr} y_{ij} \leq B_r \quad r = 2, ..., |R| \tag{3.5}$$

$$y_{ij} \in \{0, 1\} \quad \forall (i,j) \in A. \tag{3.6}$$

The objective function (3.1) minimizes the total robust cost given protection level Γ_1. Constraints (3.2)-(3.4) are flow balance constraints. Constraint (3.5) requires that the robust consumption of resource r on a feasible path must be less than or equal to resource limit B_r. A feasible path for the robust SPPRC protects against uncertainty with a specified level for each resource. A shortest path for the robust SPPRC is a feasible path that has the minimum robust cost under protection level Γ_1.

The deterministic SPPRC is known to be NP-hard [Desaulniers et al., 2005], its robust counterpart is also NP-hard [Bertsimas and Sim, 2003]. Due to the difficulty of the robust SPPRC, reducing the graph facilitates tackling the problem. Beasley and Christofides [1989] present two types of graph reduction techniques for the deterministic SPPRC. One of the techniques determines whether a node/arc should be removed based on the minimum resource consumption required to traverse that node/arc. The other technique relaxes the resource constraints in a Lagrangian fashion. Then for each node/arc, it finds a lower bound on the cost of paths traversing that node/arc. The lower bound is compared against an upper bound to determine whether the node/arc may be removed. The two types of techniques are modified to reduce the graph of the robust SPPRC and presented in Section 3.5.1.

3.5.1 Reduction of graph G

When ignoring uncertainty, the cost and resource consumption happen at the nominal level and the nominal problem is a deterministic SPPRC given by:

[PR1]

$$\min \sum_{(i,j)\in A} t_{ij1} y_{ij} \tag{3.7}$$

s.t. (3.2) - (3.4), (3.6)

$$\sum_{(i,j)\in A} t_{ijr} y_{ij} \leq B_r \quad r = 2, ..., |R|. \tag{3.8}$$

Since constraint (3.5) is more restricting than constraint (3.8), and the robust term in (3.1) is nonegative, the optimal objective value of [PR1] provides a lower bound to [P]. We now derive bounds on the robust resource consumption and use them to improve the deterministic SPPRC lower bound. Define the shortest length l as the minimum number of arcs needed to go from o to d in network G without enforcing resource limits. Let $\hat{\Gamma}_r = \min\{l, \Gamma_r\}$. For any path p, the robust consumption of resource r is at least $\sum_{(i,j)\in S^p} t_{ijr} + H_r$, where $H_r = \min_{\{S_r|S_r\subseteq A, |S_r|=\hat{\Gamma}_r\}} \sum_{(i,j)\in S_r} h_{ijr}$ is a constant for resource r. Subtracting H_r from the corresponding resource limit B_r leads to the following modified deterministic SPPRC:

[PR2]

$$\min \sum_{(i,j)\in A} t_{ij1} y_{ij} + H_1 \tag{3.9}$$

s.t. (3.2) - (3.4), (3.6)

$$\sum_{(i,j)\in A} t_{ijr} y_{ij} \leq B_r - H_r \quad r = 2, ..., |R|. \tag{3.10}$$

Since

$$H_r \le \max_{\{S_r | S_r \subseteq A, |S_r| \le \Gamma_r\}} \sum_{(i,j) \in S_r} h_{ijr} y_{ij}, \quad r = 2, ..., |R|$$

any feasible path to [P] is also feasible to [PR2] and its objective value in [P] is no less than its objective value in [PR2]. Using these results, we derive two sets of reduction rules for graph G. The first set of rules is based on resource capacity constraints, and the second is based on Lagrangian bounds.

Resource based reduction

Removing the resource constraint (3.10) from [PR2], and the constant term H_1 from the objective function, we obtain a deterministic shortest path problem. Define D_{ij}^r as the minimum cost from node i to node j with consumption of resource r as the arc cost. Then node i satisfying the following inequality may be removed from graph G:

$$D_{oi}^r + D_{id}^r > B_r - H_r \quad r = 2, ..., |R|. \tag{3.11}$$

In inequality (3.11), if the minimum consumption of resource r from o to d through node i exceeds $B_r - H_r$, any path visiting node i is infeasible to [P]. Moreover, arc (i, j) satisfying the following inequality may be removed from graph G:

$$D_{oi}^r + t_{ijr} + D_{id}^r > B_r - H_r \quad r = 2, ..., |R|. \tag{3.12}$$

Inequality (3.12) states that if the minimum consumption of resource r from o to d through arc (i, j) is greater than $B_r - H_r$, any path traversing arc (i, j) is infeasible to [P]. The minimum cost from origin to all nodes in graph G can be calculated using a label-setting algorithm. Similarly, the minimum cost from all nodes to destination can be calculated on the reversed graph of G. Algorithm 3.1 summarizes the steps of the resource based reduction.

Algorithm 3.1 Resource based reduction

```
for r = 2, ..., |R| do
    Set t_{ij1} = t_{ijr}, ∀(i, j) ∈ A.
    Solve the resulting shortest path problem using Dijkstra's algorithm.
    for all i ∈ N\{o, d} do
        if D^r_{oi} + D^r_{id} > B_r − H_r, then
            Remove node i, and its out-going and in-coming edges.
        end if
        Update N and A.
    end for
    for all (i, j) ∈ A, do
        if D^r_{oi} + t_{ijr} + D^r_{jd} > B_r − H_r, then
            Remove edge (i, j).
        end if
        if node i has no out-going edges, then
            Remove node i, and its in-coming edges.
        end if
        if node j has no in-coming edges,  then
            Remove node j, and its out-going edges.
        end if
        Update N and A.
    end for
end for
```

Lagrangian cost based reduction

Introducing Lagrangian multipliers $\mu_r \geq 0$ to resource constraints (3.10), the relaxed problem of [PR2] is:

[PR3]

$$Z^R = \min \sum_{(i,j)\in A} t_{ij1} y_{ij} + H_1 + \sum_{r=2}^{|R|} \mu_r \left(\sum_{(i,j)\in A} t_{ijr} y_{ij} - B_r + H_r \right) \tag{3.13}$$

s.t. (3.2) - (3.4), (3.6).

If the solution of [PR3] is feasible to [P], it gives an upper bound

$Z^{UB} = \sum_{(i,j)\in A} t_{ij1} y_{ij} + \max_{\{S_1 | S_1 \subseteq A, |S_1| \leq \Gamma_1\}} \sum_{(i,j)\in S_1} h_{ij1} y_{ij}.$

Define $\bar{t}_{ij1} = t_{ij1} + \sum_{r=2}^{|R|} \mu_r t_{ijr}$ as the Lagrangian cost for arc $(i,j) \in A$. The minimum Lagrangian cost from node i to node j is denoted as $L_{ij}(\mu)$ for a given set of multipliers μ_r. Any node i satisfying the following inequality may be removed from graph G:

$$L_{oi}(\mu) + L_{id}(\mu) + H_1 - \sum_{r=2}^{|R|} \mu_r (B_r - H_r) > Z^{UB}. \tag{3.14}$$

The left hand side of inequality (3.14) calculates the minimum objective value measured by function (3.13) among all paths from o to d through node i. If it is greater than the objective value of the current best solution to [P], any path traversing node i is not optimal to [P]. An arc $(i,j) \in A$ may be removed from G if the following inequality is satisfied:

$$L_{oi}(\mu) + \bar{t}_{ij1} + L_{jd}(\mu) + H_1 - \sum_{r=2}^{|R|} \mu_r (B_r - H_r) > Z^{UB}. \tag{3.15}$$

In inequality (3.15), if the minimum objective value measured by function (3.13) among all paths from o to d through arc (i,j) is greater than the upper bound on [P], the optimal solution to [P] does not involve arc (i, j). To determine a good set of multipliers μ_r, we use Kelley's cutting plane algorithm to solve the Lagrangian dual problem of [PR3]. The reduction based on Lagrangian cost is summarized in Algorithm 3.2.

Algorithm 3.2 Lagrangian based reduction

---

for all $i \in N \backslash \{o, d\}$ **do**
    if $L_{oi}(\mu) + L_{id}(\mu) + H_1 - \sum_{r=2}^{|R|} \mu_r(B_r - H_r) > Z^{UB}$, **then**
        Remove vertex i, and its out-going and in-coming edges.
    end if
    Update N and A.
end for
for all $(i, j) \in A$ **do**
    if $L_{oi}(\mu) + \bar{t}_{ij1} + L_{jd}(\mu) + H_1 - \sum_{r=2}^{|R|} \mu_r(B_r - H_r) > Z^{UB}$, **then**
        remove edge (i, j).
    end if
    if node i has no out-going edges, **then**
        Remove node i and its in-coming edges.
    end if
    if node j has no in-coming edges, **then**
        Remove node j and its out-going edges.
    end if
    Update N and A.
end for

The overall graph reduction procedure is detailed in Algorithm 3.3. It starts with the resource based reduction (Algorithm 3.1). Paths found during the resource based reduction are used to construct cuts to initialize the Lagrangian master problem. Then, Kelley's cutting plane algorithm is used to solve the Lagrangian master problem of [PR3]. Optimal Lagrangian multipliers μ_r are used to calculate the Lagrangian cost for each arc, and the Lagrangian based reduction (Algorithm 3.2) is applied. During the procedure, any path feasible to [P] is used to update Z^{UB}. If l increases or set A reduces in a reduced graph G, H_r may increase to make constraint (3.10) tighter. The resulting [PR2] may have an increased optimal objective value that in turn may further reduce graph G. Hence, the resource based reduction and the Lagrangian cost based reduction are applied iteratively until the graph can not be reduced any more.

Algorithm 3.3 Graph reduction

while G is reduced **do**
  Implement Algorithm 3.1:
    any path found in the process is kept for the initiation of Kelley's cutting plane algorithm.
    any path feasible to problem [P] is used to update Z^{UB};
  Implement Kelley's cutting plane algorithm to solve the Lagrangian dual problem of [PR3].
  Use the μ_r at the end of Kelley's cutting plane algorithm to calculate the Lagrangian cost for each arc.
  Implement Algorithm 3.2.
end while

Next, we reformulate [P] as an equivalent MIP, and we show that the resulting MIP can be solved using a sequential algorithm. The modified label-setting algorithm is presented in Section 3.7.

3.6. An equivalent MIP reformulation of the robust SPPRC

We now present a linear MIP reformulation that is equivalent to [P] and that may be solved by any MIP solver. Then we show the reformulation may be solved as a sequence of deterministic SPPRC on modified networks. Let Y be the feasible set defined by flow balance constraints (3.2)-(3.4) and (3.6), and $\mathbf{y} \in Y$ be the vector of y_{ij}. For ease of exposition, we define y_r^a, t_r^a and h_r^a as replicas of y_{ij}, t_{ijr} and h_{ijr} when arc (i, j) has the ath highest variation with respect to resource r. Without loss of generality, for each resource r, we sort the arcs using index $a = 1, ..., |A|$ such that $h_r^1 \geq h_r^2 \geq ... \geq h_r^{|A|-1} \geq h_r^{|A|}$, and $h_r^{|A|+1} = 0$. $\mathbf{y_r}$ and $\mathbf{t_r}$ are vectors of y_r^a and t_r^a, respectively. Define u_r^a as a binary variable that takes value 1 if $h_r^a y_r^a$ is selected to maximize the robust term and 0 otherwise. Then [P] is rewritten as follows:

[P1]

$$
\begin{aligned}
Z = \ & \min \mathbf{t}_1 \mathbf{y}_1 + \max_{\substack{0 \leq u_1^a \leq 1 \\ \sum_{a=1}^{|A|} u_1^a \leq \mathit{\Gamma}_1}} \sum_{a=1}^{|A|} h_1^a y_1^a u_1^a \\
& \text{s.t. } \mathbf{y} \in Y \\
& \qquad \mathbf{t_r} \mathbf{y_r} + \max_{\substack{0 \leq u_r^a \leq 1 \\ \sum_{a=1}^{|A|} u_r^a \leq \mathit{\Gamma}_r}} \sum_{a=1}^{|A|} h_r^a y_r^a u_r^a \leq B_r \quad r = 2, ..., |R|.
\end{aligned}
$$

The robust term is reformulated as $\max_{\substack{0 \leq u_r^a \leq 1 \\ \sum_{a=1}^{|A|} u_r^a \leq \mathit{\Gamma}_r}} \sum_{a=1}^{|A|} h_r^a y_r^a u_r^a$ in [P1]. Let v_r be the dual variable of constraint $\sum_{a=1}^{|A|} u_r^a \leq \mathit{\Gamma}_r$, and q_r^a be the dual variable of constraint $0 \leq u_r^a \leq 1$. Applying Theorem 2 of Bertsimas and Sim [2003] to [P1], we obtain an equivalent MIP formulation:

[P-MIP]

$$Z = \min \mathbf{t}_1\mathbf{y}_1 + \mathit{\Gamma}_1 v_1 + \sum_{a=1}^{|A|} q_1^a \tag{3.16}$$

$$\text{s.t. } \mathbf{y} \in Y$$

$$\mathbf{t}_r\mathbf{y}_r + \mathit{\Gamma}_r v_r + \sum_{a=1}^{|A|} q_r^a \leq B_r \quad r = 2, ..., |R| \tag{3.17}$$

$$v_r + q_r^a \geq h_r^a y_r^a \quad a = 1, ..., |A|, r = 1, ..., |R| \tag{3.18}$$

$$q_r^a, v_r \geq 0 \quad a = 1, ..., |A|, r = 1, ..., |R|. \tag{3.19}$$

The robust terms in [P1] are transformed into minimization problems by duality. The objective functions of the resulting dual problems are captured in the objective function of [P-MIP] and constraints (3.17), respectively. The constraints of the resulting dual minimization problems are added as constraints (3.18) and (3.19) to [P-MIP]. The latter may be solved directly using any deterministic MIP solver.

Moreover, we extend Theorem 3 of Bertsimas and Sim [2003] and show that [P] may be solved as a series of $|A| + 1$ problems:

$$Z = \min_{e=1,...,|A|+1} Z^e \tag{3.20}$$

where $e = 1, \ldots, |A| + 1$, Z^e is the minimum objective value of problem [I^e] defined as

[I^e]

$$Z^e = \min \mathbf{t}_1\mathbf{y}_1 + U^e(\mathbf{y}) \tag{3.21}$$

$$\text{s.t. } \mathbf{y} \in Y \cap \Omega \tag{3.22}$$

where

$$U^e(\mathbf{y}) = \begin{cases} \Gamma_1 h_1^1 & \text{if } e = 1, \\ \Gamma_1 h_1^e + \sum_{a=1}^{e} (h_1^a - h_1^e) y_1^a & \text{if } e = 2, \dots, |A|, \\ \sum_{a=1}^{|A|} h_1^a y_1^a & \text{if } e = |A| + 1. \end{cases}$$

and the polyhedron Ω is defined by the robust constraints (3.5). Note that for each problem [I^e], the objective function of [I^e] has no robust term and is similar to the objective function of deterministic SPPRC. However, the robust constraints still have robust terms.

Theorem 2.2 in Section 2.6 provides a sequential approach for the robust shortest path problem with one robust resource constraint. The robust terms in the objective function and in the resource constraint studied in Chapter 2 are the same. The following theorem and corollary generalize Theorem 2.2 to the case of multiple robust resource constraints. Moreover, the robust terms can be different due to either $h_{ijr_1} \neq h_{ijr_2}$ for $(i, j) \in A, r_1, r_2 \in R$ or $\Gamma_{r_1} \neq \Gamma_{r_2}$ for $r_1, r_2 \in R$, or both.

The following Theorem 3.1 linearizes the robust term in one of the robust constraints. This robust constraint is replaced by one of a set of nominal constraints, resulting in a sequence of robust subproblems. The optimal solution to the robust SPPRC can be obtained by solving these subproblems.

Theorem 3.1. *Problem [I^e] can be solved by solving $|A|+1$ problems with the $\overline{r}$th, $\overline{r} \in \{2, ..., |R|\}$, robust constraint replaced by a nominal constraint. Specifically, solving $[I^e]$ is equivalent to solving*

[I^e-Seq]

$$Z^e = \min_{w=1,\ldots,|A|+1} Z^e_w, \tag{3.23}$$

where for $w = 1, \ldots, |A|+1$,

[I^e_w]

$$Z^e_w = \min \; \mathbf{t}_1\mathbf{y}_1 + U^e(\mathbf{y}) \tag{3.24}$$

$$s.t. \; \mathbf{y} \in Y \cap \overline{\Omega} \tag{3.25}$$

$$\mathbf{t}_{\overline{r}}\mathbf{y}_{\overline{r}} + Q^w(\mathbf{y}) \leq B_{\overline{r}}, \tag{3.26}$$

where

$$Q^w(\mathbf{y}) = \begin{cases} \Gamma_{\overline{r}} h^1_{\overline{r}} & \text{if } w = 1, \\ \Gamma_{\overline{r}} h^w_{\overline{r}} + \sum_{a=1}^{w} (h^a_{\overline{r}} - h^w_{\overline{r}}) y^a_{\overline{r}} & \text{if } w = 2, \ldots, |A| \\ \sum_{a=1}^{|A|} h^a_{\overline{r}} y^a_{\overline{r}} & \text{if } w = |A|+1. \end{cases}$$

$h^a_{\overline{r}}$ is ordered such that $h^1_{\overline{r}} \geq h^2_{\overline{r}} \geq \cdots \geq h^{|A|}_{\overline{r}} \geq h^{|A|+1}_{\overline{r}} = 0$, and $\overline{\Omega}$ is the polyhedron defined by the robust constraints except the $\overline{r}$th robust constraint.

Proof of Theorem 3.1. The proof is presented in Appendix 6.3. □

In $[\mathrm{I}^e_w]$, the $\bar{r}$th robust constraint is replaced by nominal constraint (3.26). When Y is updated as $Y \cap \{y|\mathbf{t}_{\bar{r}}\mathbf{y}_{\bar{r}} + Q^w(\mathbf{y}) \leq B_{\bar{r}}\}$, $[\mathrm{I}^e_w]$ shares the same structure as $[\mathrm{I}^e]$ except that it has one less robust constraint. This leads to the following corollary.

Corollary 3.1. *If the robust SPPRC has n_r distinct variations across all arcs for resource $r \in R$, solving at most $\prod_{r \in R} n_r$ deterministic SPPRC solves the robust SPPRC.*

Proof of Corollary 3.1. According to Theorem 3.1, $[\mathrm{I}^e_w]$ can also be transformed to a sequence of robust problems with one less robust constraint. When the transformation is applied repeatedly for every robust subproblem, all the robust constraints may be replaced by nominal constraints. When a robust subproblem has n_r distinct deviations across all arcs in the rth robust constraint, the number of its subproblems is n_r. Then the robust SPPRC may be transformed into $\prod_{r \in R} n_r$ deterministic SPPRC. □

Note that an optimal solution $\bar{\mathbf{y}}_\mathbf{1}$ to $[\mathrm{I}^e]$ is also optimal to $[\mathrm{I}^{e+1}]$ if $h_1^{e+1} \neq h_1^e$ and $T^e \cap \{1, ..., e\} = \varnothing$, where $T^e = \{a|\bar{y}_1^a = 1, a = 1, ..., |A|\}$. This is because when the solution process moves from problem $[\mathrm{I}^e]$ to $[\mathrm{I}^{e+1}]$, only the costs of arcs $a = 1, \ldots, e$ are updated. So, if the optimal path found in problem $[\mathrm{I}^e]$ does not use any arc $a \in \{1, \ldots, e\}$, it is also optimal to $[\mathrm{I}^{e+1}]$.

This sequential approach to solve [P] is presented in Algorithm 3.4. It creates a sequence of deterministic SPPRC and solves the resulting nominal problems using a label-setting algorithm. For a problem with $|R|$ resources and n_r distinct variations in each resource $r \in R$, the total number of nominal problems to solve in the worst case is $\prod_{r=1}^{|R|} n_r$. Algorithm 3.4 grows fast when $|R|$ or n_r increases. Section 3.7 presents an algorithm that is insensitive to $|R|$ and n_r. It modifies the label-setting algorithm that uses a new dominance rule designed for the robust context.

Algorithm 3.4 The sequential algorithm

for $h_1^e = h_1^1, ..., h_1^{|A|+1}$ and $h_1^e \neq h_1^{e+1}, e = 1, ..., |A|$, **do**
  if current best solution has an arc with a cost greater than h_1^e, **then**
    Find arc $(i, j) \in A$ with arc cost deviation greater than h_1^e.
    Set $t_{ij1} = t_{ij1} + h_{ij1} - h_1^e$.
    for $\{t_{e,2}, ..., t_{e,|R|}\} \in \prod_{r=2}^{R} \{t_{e,r} \in \{t_{1,r}, ..., t_{|A|+1,r}\} | t_{e,r} \neq t_{e',r}, e' = 1, ..., |A| + 1, e' \neq e\}$,
do
      for $r = 2, ...|R|$ **do**
        for $(i, j) \in A$ **do**
          if F **then**ind all arc $(i, j) \in A$ with $t_{ijr} > t_{e,r},$.
            Set $t_{ijr} = t_{ijr} + t_{ijr} - t_{e,r}$.
          end if
        end for
      end for
      Implement Algorithm 1.1 and find the optimal path $\overline{\mathbf{y}}$.
      if the cost of $\overline{\mathbf{y}}$ is less than current best solution, **then**
        Update current best solution.
      end if
    end for
  end if
end for

3.7. The modified label-setting algorithm for the robust SPPRC

We first show that the dominance rule in the label-setting algorithm becomes invalid when arc consumption is allowed to vary in an interval. Define $\Lambda_i^p = \{\lambda_{ir}^p : r = 1, ..., |R|\}$ as the resource consumption vector of robust SPPRC, where λ_{ir}^p is the robust consumption of resource r for path p from origin o to node i. Let $C_i^p = \{c_{ir}^p : r = 1, ..., |R|\}$ be the resource consumption vector of deterministic SPPRC, where c_{ir}^p is the nominal consumption of resource r for path p from origin o to node i. Figure 3.1 shows a network of deterministic SPPRC with 2 resources. The origin and the destination in Figure 3.1 are nodes 1 and 4, respectively. On each arc, the first and the second numbers between parentheses represent the consumption of resources 1 and 2 on that arc. The limit on resource 2 is 20. There are two paths 1 and 2 from origin to destination consisting of arcs $\{(1,3),(3,4)\}$ and $\{(1,2),(2,3),(3,4)\}$ with labels $C_4^1 = [7,9]$ and $C_4^2 = [8,9]$, respectively. Path $\{(1,2),(2,3),(3,4)\}$ is the optimal path. Moreover, as C_3^1 dominates C_3^2, discarding C_3^2 at node 3 does not affect optimality. Figure 3.2 adds variations to the graph in Figure 3.1. A pair of numbers in curly brackets shows the variations of resources 1 and 2 on each arc. Setting the protection level Γ to 2 for both resources, then $\Lambda_3^1 = [12,11]$ and $\Lambda_3^2 = [17,14]$. Based on the dominance rule for deterministic SPPRC, Λ_3^2 is dominated by Λ_3^1, only one path to node 4 remains with label $\Lambda_4^1 = [22,19]$. However, the optimal path should be path 2 with $\Lambda_4^2 = [21,18]$. This shows that the dominance rule for SPPRC is invalid. The reason is that when reaching node 3, path 2 has already encountered the first two biggest variations for each resource, whereas path 1 has only encountered one of its two variations for each resource. Therefore, to derive a valid dominance rule, we need not only consider the information given by Λ_i^p but also the information from node i to the destination d.

Let $G_i = (N_i, A_i)$ denote a subgraph of G where N_i is the set of nodes reachable from

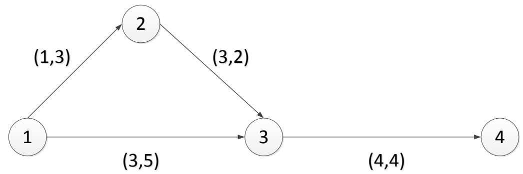

Figure 3.1: A network for nominal resource constrained shortest path problem.

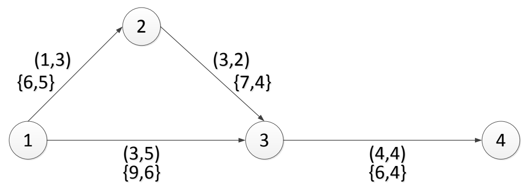

Figure 3.2: A network for robust resource constrained shortest path problem.

i, and A_i are the set of outgoing arcs of N_i. Given protection level Γ_r,

$$\Delta_{ir} = \arg\max_{S_{ir}} \sum_{(j,k)\in S_{ir}} h_{jkr} := \{S_{ir}|S_{ir} \subseteq A_i, |S_{ir}| \leq \Gamma_r\}$$

is a set of arcs whose variations of resource r are the highest Γ_r variations across G_i when $\Gamma_r \leq A_i$, or are the highest $|A_i|$ variations across G_i when $\Gamma_r > |A_i|$. Δ_{ir} can be determined using a modified breadth first search as shown in Algorithm 3.5.

For a path p from origin o to node i composed of arcs S^p, let us define S^p_r as a subset of S^p for each resource $r \in R$, and

$$\Phi^p_{ir} = \arg\max_{S^p_r} \sum_{(j,k)\in S^p_r} h_{jkr} := \{S^p_r|S^p_r \subseteq S^p, |S^p_r| \leq \Gamma_r\}$$

as the set of arcs such that the sum of their variations of resource r is maximized under protection level Γ_r. Then an upper bound on the total deviation of consumption of resource

Algorithm 3.5 Modified BFS for G_i with h_{jkr} as arc cost

Initialization:
Set h_{jkr} as the arc cost for each $(j,k) \in G_i$.
Create a vector V of size $\mathit{\Gamma}_r$.
Set all elements in V to 0.
Create a queue M.
Enqueue i onto M.
while M is not empty, **do**
    $j \leftarrow M$.dequeue(),
    mark j.
    for all $(j,k) \in G_i$ **do**
        for all $g \in V$ **do**
            if $h_{jkr} > g$, **then**
                insert h_{jkr} before g,
                pop out the end element of V,
                break.
            end if
        end for
        if k is not marked, **then**
            enqueue k onto M.
        end if
    end for
end while
Return $\mathit{\Delta}_{ir} = \{(j,k) | h_{jkr} > 0, h_{jkr} \in V\}$.

r under protection level Γ_r for a path to destination d extended from p is

$$\xi_{ir}^p = \max_{\{\Upsilon \subseteq \Phi_{ir}^p \cup \Delta_{ir}, |\Upsilon| \leq \Gamma_r\}} \sum_{(j,k)\in\Upsilon} h_{jkr},$$

where $\Phi_{ir}^p \cup \Delta_{ir}$ consists of a subset of arcs from path p and a subset of arcs from A_i. $\Xi_i^p = (\xi_{i1}^p, ..., \xi_{i|R|}^p)$ represents a vector that consists of the upper bound on the total variation of each resource consumption for a path to destination d extended from p.

Theorem 3.2. *Given paths* 1 *and* 2 *both from origin o to node i, and path* 3 *from node i to destination d, if $C_i^1 + \Xi_i^1 < \Lambda_i^2$, then path* 4 *composed of paths* 1 *and* 3 *dominates path* 5 *composed of paths* 2 *and* 3*. Label C_i^2 may be eliminated at node i.*

Proof of Theorem 3.2. For path 1 and $\forall r \in R$,

$$\begin{aligned}
\xi_{ir}^1 &= \max_{\{\Upsilon \subseteq \Phi_{ir}^p \cup \Delta_{ir}, |\Upsilon| \leq \Gamma_r\}} \sum_{(j,k)\in\Upsilon} h_{jkr} \\
&= \max_{\{\Upsilon | \Upsilon \subseteq S^1 \cup A_i, |\Upsilon| \leq \Gamma_r\}} \sum_{(j,k)\in\Upsilon} h_{jkr} && (3.27) \\
&\geq \max_{\{S_r^4 | S_r^4 \subseteq S^1 \cup S^3, |S_r^4| \leq \Gamma_r\}} \sum_{(j,k)\in S_r^4} h_{jkr} && (3.28) \\
&= \max_{\{S_r^4 | S_r^4 \subseteq S^4, |S_r^4| \leq \Gamma_r\}} \sum_{(j,k)\in S_r^4} h_{jkr}. && (3.29)
\end{aligned}$$

Equality (3.27) holds due to the definition of Φ_{ir}^p and Δ_{ir}. Inequality (3.28) holds because

$S^3 \subseteq A_i$. $\xi^1_{ir} \geq \max_{\{S^4_r | S^4_r \subseteq S^4, |S^4_r| \leq \Gamma_r\}} \sum_{(j,k) \in S^4_r} h_{jkr}$ leads to

$$c^1_{ir} + \max_{\{S^4_r | S^4_r \subseteq S^4, |S^4_r| \leq \Gamma_r\}} \sum_{(j,k) \in S^4_r} h_{jkr} \leq c^1_{ir} + \xi^1_{ir}. \tag{3.30}$$

Recall that $\lambda^p_{ir} = c^p_{ir} + \max_{\{S^p_r | S^p_r \subseteq S^p, |S^p_r| \leq \Gamma_r\}} \sum_{(i,j) \in S^p_r} h_{jkr}$, then

$$\lambda^2_{ir} \leq c^2_{ir} + \max_{\{S^5_r | S^5_r \subseteq S^2 \cup S^3 = S^5, |S^5_r| \leq \Gamma_r\}} \sum_{(j,k) \in S^5_r} h_{jkr} \quad r = 1, ..., |R|. \tag{3.31}$$

If $c^1_{i\overline{r}} + \xi^1_{i\overline{r}} < \lambda^2_{i\overline{r}}$ for resource $\overline{r} \in R$, $C^1_i + \Xi^1_i < \Lambda^2_i$ leads to the following set of inequalities:

$$c^1_{ir} + \xi^1_{ir} \leq \lambda^2_{ir} \leq c^2_{ir} + \max_{\{S^5_r | S^5_r \subseteq S^5, |S^5_r| \leq \Gamma_r\}} \sum_{(j,k) \in S^5_r} h_{jkr} \quad \forall r \in R \backslash \{\overline{r}\} \tag{3.32}$$

$$c^1_{i\overline{r}} + \xi^1_{i\overline{r}} < \lambda^2_{i\overline{r}} \leq c^2_{i\overline{r}} + \max_{\{S^5_{\overline{r}} | S^5_{\overline{r}} \subseteq S^5, |S^5_{\overline{r}}| \leq \Gamma_{\overline{r}}\}} \sum_{(j,k) \in S^5_{\overline{r}}} h_{jk\overline{r}}. \tag{3.33}$$

Hence,

$$c^1_{ir} + \max_{\{S^4_r | S^4_r \subseteq S^4, |S^4_r| \leq \Gamma_r\}} \sum_{(j,k) \in S^4_r} h_{jkr} \leq c^2_{ir} + \max_{\{S^5_r | S^5_r \subseteq S^5, |S^5_r| \leq \Gamma_r\}} \sum_{(i,j) \in S^5_r} h_{jkr} \quad \forall r \in R \backslash \{\overline{r}\} \tag{3.34}$$

$$c^1_{i\overline{r}} + \max_{\left\{S^4_{\overline{r}} | S^4_{\overline{r}} \subseteq S^4, |S^4_{\overline{r}}| \leq \Gamma_{\overline{r}}\right\}} \sum_{(j,k) \in S^4_{\overline{r}}} h_{jk\overline{r}} < c^2_{i\overline{r}} + \max_{\{S^5_{\overline{r}} | S^5_{\overline{r}} \subseteq S^5, |S^5_{\overline{r}}| \leq \Gamma_{\overline{r}}\}} \sum_{(i,j) \in S^5_{\overline{r}}} h_{jk\overline{r}}. \tag{3.35}$$

Moreover, as the nominal consumption of resource $r \in R$ on path 3 is $c^4_{dr} - c^1_{ir}$, which equals $c^5_{dr} - c^2_{ir}$, adding $c^4_{dr} - c^1_{ir}$ and $c^5_{dr} - c^2_{ir}$ to the left and right hand sides of inequalities

(3.34) and (3.35) results in

$$c^4_{dr} + \max_{\{S^4_r|S^4_r\subseteq S^4,|S^4_r|\le\Gamma_r\}} \sum_{(j,k)\in S^4_r} h_{jkr} = \lambda^4_{dr} \le c^5_{dr} + \max_{\{S^5_r|S^5_r\subseteq S^5,|S^5_r|\le\Gamma_r\}} \sum_{(j,k)\in S^5_r} h_{jkr} = \lambda^5_{dr} \quad \forall r \in R\backslash\{\bar{r}\} \tag{3.36}$$

$$c^4_{d\bar{r}} + \max_{\left\{S^4_{\bar{r}}|S^4_{\bar{r}}\subseteq S^4,|S^4_{\bar{r}}|\le\Gamma_{\bar{r}}\right\}} \sum_{(j,k)\in S^4_{\bar{r}}} h_{jk\bar{r}} = \lambda^4_{d\bar{r}} < c^5_{d\bar{r}} + \max_{\{S^5_{\bar{r}}|S^5_{\bar{r}}\subseteq S^5,|S^5_{\bar{r}}|\le\Gamma_{\bar{r}}\}} \sum_{(j,k)\in S^5_{\bar{r}}} h_{jk\bar{r}} = \lambda^5_{d\bar{r}}. \tag{3.37}$$

Hence $\Lambda^4_{dr} < \Lambda^5_{dr}$. As path 3 can be any path from i to d, C^1_i dominates C^2_i. □

The dominance rule in Theorem 3.2 is integrated within a modified label-setting algorithm for the robust SPPRC. The algorithm is detailed in Algorithm 3.6.

3.8. Numerical tests

In this section, we report on extensive numerical testing to compare the three approaches to solve the robust SPPRC. The first approach solves the MIP reformulation [P-MIP] directly as a linear mixed integer program. The second approach solves a sequence of nominal problems and the third approach uses the modified label-setting algorithm. Before any of the approaches is used, we first reduce the problem using the graph reduction algorithm of Section 3.5.1. Then each of the three approaches is used to solve the reduced instances. All algorithms are coded in C++, and MIP and LP models are solved using CPLEX 12.4. Testing is carried out on a workstation with Xeon processor 3.00GHz and 8GB RAM.

We construct 60 random instances, denoted by **LG**, based on 24 instances from Beasley and Christofides [1989], denoted by **BC** as shown in Table 3.1. For example, **LG** instances

Algorithm 3.6 Label setting algorithm for robust resource constrained shortest path problem

Initialization:
$p^* = 0$, $p = 0$, $C_0^p = (0, ..., 0)$ where $C_0^p \in \mathbb{R}^{|R|}$; mark C_0^p as not dominated
create a vector V_{ir}^p of size Γ_r initialized with all elements as 0 for all $r \in R$.
$L_0 = \{C_0^p\}$, $L_i = \{(\infty, B_2, ..., B_{|R|})\}, \forall i \in N \backslash \{o\}$; $Unprocessed = \{C_0^p\}$
while $Unprocessed \neq \varnothing$ **do**
Extract a label $C_i^{\tilde{p}}$ from $Unprocessed$
if $C_i^{\tilde{p}}$ is not dominated **then**
for all $j : (i, j) \in A$ **do**
$p = p + 1$; $C_j^p = f_{ij}(C_i^{\tilde{p}})$; mark C_j^p as not dominated.
for all $r \in R$ **do**
create V_{jr}^p as a copy of $V_{ir}^{\tilde{p}}$.
for all g in V_{jr}^p **do**
if $h_{ijr} > g$ **then**
insert h_{ijr} before g, pop out the end element in V_{jr}^p, break.
end if
end for
end for
$vars$: = sum of all elements in V_{j1}^p.
if $c_{j1}^p + vars > UB$ **then**
mark C_j^p as dominated.
else
for all $r = 2, ..., R$ **do**
$vars$: = sum of all elements in V_{jr}^p.
if $c_{jr}^p + vars > B_r$ **then**
mark C_j^p as infeasible,
break.
end if
end for
end if
if C_j^p is feasible and not dominated **then**
for all $C_j^{\bar{p}} \in L_j$ **do**
Perform dominance check for C_j^p and $C_j^{\bar{p}}$.
if $C_j^{\bar{p}} + \Xi_j^{\bar{p}} < \Lambda_j^p$ **then**
mark C_j^p as dominated and break.
else if $C_j^p + \Xi_j^p < \Lambda_j^{\bar{p}}$ **then**
mark $C_j^{\bar{p}}$ as dominated,
delete $V_{jr}^{\bar{p}}$ for all $r \in R$.
if $C_j^{\bar{p}}$ is processed **then**
$L_j = L_j \backslash \{C_j^{\bar{p}}\}$.
end if
end if
end for
if C_j^p is not dominated **then**
$L_j = L_j \cup \{C_j^p\}$.
if $j = d$ and $\lambda_{d1}^p < UB$ **then**
update UB and $p^* = p$.
end if
else
delete C_j^p,
delete V_{jr}^p for all $r \in R$.
end if
end if
end for
mark $C_i^{\tilde{p}}$ as processed.
else
delete $C_i^{\tilde{p}}$.
end if
end while
Return p^*

7, 13, 19 and 25 are based on **BC** instance 5. **BC** instances $1-4, 9-12, 17-20$ have one resource, while instances $5-8, 13-6, 21-24$ have 10 resources, in addition to cost. **BC** instances $1, 2, 5, 6, 9, 10, 13, 14, 17, 18, 21$ and 22 are generated using the scheme of Handler and Zang [1980]. This scheme is designed such that the optimal path has a low ranking when the unconstrained paths are ordered from lowest cost to highest cost. Specifically, the nodes in these instances are randomly generated on a square. The cost on arc (i, j) is an integer based on the Euclidean distance between nodes i and j. Resource consumption on an arc is determined by multiplying the reciprocal of arc cost by a uniformly generated random variable. As a result, resource consumption is inversely related to arc cost. In **BC** instances $3, 4, 7, 8, 11, 12, 15, 16, 19, 20, 23$ and 24, both cost and resource consumption are uniformly generated integers in $[0, 5]$. Arc (i, j) is determined by randomly generating i from $[1, |N|]$ and j from $[i+1, \min(|N|, i+|N|/4)]$.

Table 3.1: Relationship between **BC** instances and **LG** instances.

$\lvert R\rvert - 1$	1	1	10	2	3	4	10
	BC	**LG**	**BC**	**LG**			
	1	1	5	7	13	19	25
	2	2	6	8	14	20	26
	3	31	7	37	43	49	55
	4	32	8	38	44	50	56
	9	3	13	9	15	21	27
	10	4	14	10	16	22	28
	11	33	15	39	45	51	57
	12	34	16	40	46	52	58
	17	5	21	11	17	23	29
	18	6	22	12	18	24	30
	19	35	23	41	47	53	59
	20	36	24	42	48	54	60

In our tests, **LG** instances with $2, 3$ and 4 resources are obtained from **BC** instances with 10 resources where the first $2, 3$ and 4 resources are considered. In all 60 instances, the deviation h_{ijr} in cost and resource consumption is an integer generated in the interval $[0, t_{ijr}]$ following a uniform distribution. The protection level is varied between 2 and 5 resulting in a total of 240 instances. We report on the effect of graph reduction on the size of the instances and on the quality of the initial lower and upper bounds. Then, we compare the three solution approaches of the robust SPPRC.

3.8.1 Effects of graph reduction

Table 3.2 summarizes the results of the graph reduction algorithm. Average results are reported and grouped according to resource consumption. Specifically, instances with resource consumption inversely related to cost are denoted as ***Inverse***, and instances with uniformly generated resource consumption are denoted as ***Uniform***. **Avg_A%** denotes the average percentage of arcs remaining after graph reduction. **Avg_UBG%** refers to the average gap calculated as $100 * \frac{(UB-LB)}{LB}$ where UB and LB are the upper and lower bounds calculated by the graph reduction algorithm. **Avg_OPG%** refers to the average gap calculated as $100 * \frac{Opt-LB}{LB}$ where Opt refers to the optimal objective value obtained after solving the instance by the modified label-setting algorithm. The algorithm runs in less than 0.0005 seconds in all instances, so individual times are not reported. When resource consumption is inversely related to cost, the number of arcs of the reduced graph is 10.85% of the original graph. Reduction is less significant for uniform resource consumption with about 72% of the arcs remaining in the reduced graph.

The quality of the lower and upper bounds show similar trends. The LB is about

91.87% of the optimal objective value and 107.02% of the UB for ***Inverse*** instances. On the other hand, the LB is about 181.88% of the optimal objective value and 327.60% of the UB for ***Uniform*** instances.

These statistics suggest that ***Uniform*** instances have more dense networks and may be more difficult to solve. Because the performance of Lagrangian based reduction depends on the quality of the Lagrangian multipliers and the UB, we expect that having tight LB and UB tends to result in a smaller network after reduction. This is shown by the positive correlation between **Avg_A%** and **Avg_UBG%**. A single implementation of resource based reduction under protection level Γ removes at least as many vertices and arcs as it does under $\Gamma - 1$. This is because H_r increases when Γ increases. As a result, the right hand sides of inequalities (3.11) and (3.12) become more restricting. However, when Γ increases, Lagrangian cost based reduction may be stronger or weaker depending on the upper bound, H_r and μ_r. In Table 3.3, **Avg_Res%** and **Avg_Lag%** report the average percentage of arcs removed by resource based reduction and Lagrangian cost based reduction, respectively. The additional number of arcs removed from $\Gamma = 2$ to $\Gamma = 3$ is higher under the Lagrangian based reduction. The percentages are stable for higher values of Γ. This is because $\hat{\Gamma}_r = \min\{l, \Gamma_r\}$.

3.8.2 Comparison of solution approaches for the robust SPPRC

Tables 3.4 to 3.8 report on the size of the original graph given by the number of nodes $|N|$ and number of arcs $|A|$, the number of resources $|R| - 1$, the lower and upper bounds and the relative gap after graph reduction, the clock time used by the sequential algorithm, by

Table 3.2: Summary results on graph reduction procedure.

Inverse	Avg_UBG%	Avg_OPG%	Avg_A%
$\Gamma = 2$	80.93	80.75	5.26
$\Gamma = 3$	114.26	91.98	12.63
$\Gamma = 4$	116.26	97.18	12.75
$\Gamma = 5$	116.63	97.55	12.76
Average	107.02	91.87	10.85
Uniform			
$\Gamma = 2$	230.75	137.77	62.56
$\Gamma = 3$	344.03	201.52	74.38
$\Gamma = 4$	367.80	192.79	75.38
$\Gamma = 5$	367.80	195.45	75.46
Average	327.60	181.88	71.94

Table 3.3: Percentage of arcs removed by resource based reduction and Lagrangian cost based reduction.

	Inverse		Uniform	
	Avg_Res%	Avg_Lag%	Avg_Res%	Avg_Lag%
$\Gamma = 2$	52.66	42.09	10.52	26.92
$\Gamma = 3$	53.07	34.31	9.99	15.63
$\Gamma = 4$	53.08	34.17	9.96	14.67
$\Gamma = 5$	53.08	34.16	9.96	14.58

CPLEX on the original graph (**OG**) and on the reduced graph (**RG**), and the modified label-setting algorithm, and the optimal objective value (**Opt**). Tables 3.5 to 3.8 report on detailed statistics on all 60 instances with protection levels 2 to 5, respectively, and Table 3.4 gives average results.

Both the modified label-setting algorithm and CPLEX successfully solved all feasible instances, and determined that the rest are infeasible. The sequential algorithm fails to solve instances 49-60 for $\Gamma = 2, 3$. For instances 1-48, it uses 64.39 seconds and 343.42 seconds which is 487 and 1431 times slower than the modified label-setting algorithm. Since the sequential algorithm is dominated for $\Gamma = 2, 3$ and instances become more difficult for $\Gamma = 4, 5$, we omitted the sequential algorithm from the rest of the testing. However, the sequential algorithm solves a set of independent problems which could benefit from parallel implementation. The modified label-setting algorithm achieves significantly smaller computational times than CPLEX. Times vary between 0.001 and 0.047 with an average 0.02. The modified label-setting algorithm is on average 160 times faster than CPLEX. Looking at the average times for increasing Γ, there is no evidence that instances become harder with higher Γ both for modified label-setting algorithm and CPLEX. On the other hand, graph reduction reduces time of CPLEX by about 50%.

While instance 54 is infeasible in the original data, the other infeasible instances are infeasible because of the robust term. For all infeasible instances in Tables 3.5 to 3.8, we increase the resource capacities and rerun the testing. Table 3.9 reports on these instances and confirms the findings.

The results in Table 3.4 suggest again that ***Uniform*** instances are more difficult to solve than ***Inverse*** instances. The modified label-setting algorithm solves ***Inverse*** instances in an average of 0.0004 seconds which is about 76 times faster than the average time used to solve ***Uniform*** instances. The average times used by CPLEX on reduced graph are 0.136 seconds and 7.414 seconds for ***Inverse*** and ***Uniform*** instances, respectively. Again, the size of the network after graph reduction is an important factor affecting the difficulty of the instances.

3.9. Conclusion

In this chapter, we address the robust SPPRC where the cost and resource consumption parameters on an arc are defined by intervals and a protection level is specified for each random parameter. We first develop new graph reduction techniques based on resources and on Lagrangian relaxation. Then, we present an MIP reformulation of robust SPPRC and prove that it may be solved as a sequence of deterministic SPPRC. In addition, a modified label-setting algorithm is proposed. The algorithm relies on a new dominance rule. First, a valid upper bound on the total variation in the worst case is developed. This upper bound is added to the nominal resource consumption of the partial path to determine if it can dominate other paths. We use the MIP reformulation, the sequential algorithm, and the modified label-setting algorithm to solve 240 instances with up to 10 resources, 500 nodes and 4868 arcs and with varying protection levels. The results show that the graph reduction procedure helps reduce the overall solution time by half. The modified label-setting algorithm is far superior to the other two solution approaches. The scalability of the sequential algorithm is poor and thus not a preferable choice when there are many resources and when width of the intervals of uncertainty vary greatly.

Table 3.4: Summary of computational time.

		$\lvert R\rvert - 1$	Algorithm 3.6	Cplex OG	Cplex RG	Algorithm 3.4
$\Gamma = 2$	Inverse	1	0.000	0.412	0.028	0.475
		2	0.000	1.776	0.419	3.216
		3	0.000	2.508	0.052	52.585
		4	0.000	6.580	0.036	456.052
		10	0.000	15.520	0.017	
		Average	0.000	5.359	0.110	128.082
	Uniform	1	0.015	0.512	0.697	0.014
		2	0.002	1.147	0.243	0.022
		3	0.014	2.298	1.321	0.406
		4	0.034	3.349	1.987	2.339
		10	0.094	34.812	30.908	
		Average	0.032	8.424	7.031	0.695
	Average		0.016	6.891	3.571	64.389
$\Gamma = 3$	Inverse	1	0.000	0.430	0.062	0.957
		2	0.001	1.112	0.234	6.231
		3	0.001	1.977	0.089	122.837
		4	0.001	6.810	0.137	2614.169
		10	0.001	8.982	0.054	
		Average	0.001	3.862	0.115	686.048
	Uniform	1	0.026	0.612	0.636	0.015
		2	0.009	2.001	0.635	0.021
		3	0.029	3.615	4.188	0.863
		4	0.053	8.207	6.457	2.238
		10	0.024	24.237	25.163	
		Average	0.028	7.734	7.416	0.784
	Average		0.014	5.798	3.765	343.416
$\Gamma = 4$	Inverse	1	0.001	0.379	0.066	
		2	0.000	1.874	0.358	
		3	0.001	2.952	0.102	
		4	0.001	6.198	0.178	
		10	0.001	11.718	0.065	
		Average	0.001	4.624	0.154	
	Uniform	1	0.028	0.561	0.638	
		2	0.010	0.976	0.493	
		3	0.053	4.644	5.872	
		4	0.064	10.636	10.224	
		10	0.017	23.402	21.581	
		Average	0.034	8.044	7.762	
	Average		0.017	6.334	3.958	
$\Gamma = 5$	Inverse	1	0.000	0.451	0.066	
		2	0.001	2.110	0.356	
		3	0.000	2.911	0.110	
		4	0.001	6.598	0.225	
		10	0.000	13.274	0.065	
		Average	0.000	5.069	0.164	
	Uniform	1	0.028	0.771	0.668	
		2	0.016	0.865	0.651	
		3	0.053	5.974	4.934	
		4	0.062	10.057	9.490	
		10	0.019	22.004	21.497	
		Average	0.036	7.934	7.448	
	Average		0.018	6.501	3.806	

Table 3.5: Detailed results with with protection level $\Gamma = 2$.

		Original graph			Reduced graph							Time				
												Algorithm 3.6	CPLEX		Algorithm 3.4	Opt
	instance	$\|N\|$	$\|A\|$	$\|R\|-1$	$\|N\|$	$\|A\|$	Res%	Lag%	LB	UB	Gap%		OG	RG		
	1	100	955	1	13	19	6.91	91.10	133	181	36.09	0	0.167	0.005	0.009	181
	2	100	955	1	7	7	7.33	91.94	149	165	10.74	0	0.078	0.005	0.007	165
	3	200	2040	1	7	7	99.66	0.00	420	-	-	0	0.096	0.001	0.005	-
	4	200	2040	1	7	7	99.66	0.00	420	-	-	0	0.091	0.001	0.004	-
	5	500	4858	1	184	396	0.60	91.25	488.571	1061	117.16	0.001	0.803	0.136	1.839	937
	6	500	4858	1	55	81	0.89	97.45	654	977	49.39	0	1.234	0.017	0.988	977
	7	100	990	2	6	6	1.31	98.08	91	104	14.29	0	0.123	0.005	0.009	104
	8	100	990	2	62	184	4.24	77.17	89	182	104.49	0	0.297	0.094	6.538	159
	9	200	2080	2	10	13	25.43	73.94	271	385	42.07	0	0.414	0.006	0.016	385
	10	200	2080	2	195	1269	38.99	0.00	267.971	-	-	0.002	2.286	2.384	9.19	566
	11	500	4847	2	55	79	87.79	10.58	863	1878	117.61	0	1.325	0.014	1.514	1513
	12	500	4847	2	46	63	95.52	3.18	867	2194	153.06	0	6.209	0.009	2.03	2194
	13	100	990	3	6	6	1.52	97.88	91	104	14.29	0	0.143	0.006	0.014	104
	14	100	990	3	62	182	4.34	77.27	89	182	104.49	0	0.329	0.11	225.431	159
Inverse	15	200	2080	3	86	198	73.46	17.02	268	844	214.93	0.001	3.246	0.154	31.488	666
	16	200	2080	3	41	67	82.21	14.57	510	758	48.63	0	1.549	0.014	17.043	758
	17	500	4847	3	54	77	87.93	10.48	863	1878	117.61	0	3.516	0.017	14.308	1513
	18	500	4847	3	46	63	95.75	2.95	867	2194	153.06	0	6.263	0.011	27.226	2194
	19	100	990	4	9	11	2.32	96.57	103	127	23.30	0	0.31	0.007	3.298	127
	20	100	990	4	21	39	4.55	91.52	96.6444	159	64.52	0	0.373	0.014	407.645	159
	21	200	2080	4	68	138	72.31	21.06	450	844	87.56	0	4.905	0.08	456.505	740
	22	200	2080	4	41	68	89.86	6.88	509	928	82.32	0	5.017	0.02	256.119	815
	23	500	4847	4	47	63	90.06	8.64	863	1878	117.61	0	11.706	0.013	268.447	1878
	24	500	4847	4	102	161	96.68	0.00	858	-	-	0.001	17.171	0.082	1344.3	3599
	25	100	990	10	9	11	7.68	91.21	103	127	23.30	0	0.41	0.011	0	127
	26	100	990	10	9	11	14.04	84.85	108	159	47.22	0	0.968	0.012	0	159
	27	200	2080	10	16	19	99.09	0.00	393.98	-	-	0	2.196	0.004	0	-
	28	200	2080	10	0	0	100.00	0.00	-10000	-	-	0	1.009	0	0	-
	29	500	4847	10	40	53	91.79	7.12	863	1878	117.61	0	42.26	0.026	0	1878
	30	500	4847	10	71	109	97.75	0.00	858	-	-	0.001	46.277	0.047	0	-
	31	100	959	1	97	845	2.92	8.97	1.5	12	700.00	0.013	0.386	0.153	0.01	5
	32	100	959	1	88	541	3.65	39.94	2	10	400.00	0.016	0.239	0.17	0.008	7
	33	200	1971	1	98	216	2.94	86.10	6	8	33.33	0.001	0.186	0.028	0.008	8
	34	200	1971	1	194	1915	2.84	0.00	6	39	550.00	0.021	0.241	0.294	0.011	8
	35	500	4978	1	467	4461	5.99	4.40	6	16	166.67	0.025	1.027	2.866	0.027	11
	36	500	4978	1	443	2799	6.41	37.36	6	14	133.33	0.013	0.994	0.67	0.022	9
	37	100	999	2	89	834	10.41	6.11	3.375	12	255.56	0.005	0.324	0.313	0.031	9
	38	100	999	2	89	601	10.41	29.43	3.625	10	175.86	0.002	0.333	0.336	0.019	10
	39	200	1960	2	94	183	11.63	79.03	5.25	9	71.43	0	0.673	0.058	0.015	9
	40	200	1960	2	184	1767	9.85	0.00	5.75	-	-	0.002	0.69	0.728	0.037	12
	41	500	4868	2	59	96	8.38	89.65	4	6	50.00	0	1.836	0.012	0.016	6
	42	500	4868	2	49	83	11.65	86.65	4	6	50.00	0	3.023	0.01	0.016	6
	43	100	999	3	90	889	11.01	0.00	3.8202	35	816.18	0.027	1.211	1.521	0.074	10
	44	100	999	3	90	878	12.11	0.00	4.52778	-	-	0.015	1.251	1.292	0.974	16
Uniform	45	200	1960	3	184	1774	9.49	0.00	5.25	-	-	0.037	1.831	2.742	1.116	14
	46	200	1960	3	181	1734	11.53	0.00	5.75	-	-	0.007	2.127	2.333	0.197	16
	47	500	4868	3	103	178	11.40	84.94	4	7	75.00	0	3.604	0.03	0.037	7
	48	500	4868	3	36	49	12.86	86.13	4	7	75.00	0	3.764	0.008	0.036	7
	49	100	999	4	90	889	11.01	0.00	3.86654	-	-	0.037	1.633	1.773	2.531	14
	50	100	999	4	90	878	12.11	0.00	4.57734	-	-	0.005	1.201	1.242	1.841	16
	51	200	1960	4	182	1754	10.51	0.00	5.79245	-	-	0.154	5.523	4.954	5.408	18
	52	200	1960	4	181	1718	12.35	0.00	6.70245	-	-	0.007	3.638	3.912	2.205	22
	53	500	4868	4	97	160	12.78	83.94	4	7	75.00	0	5.088	0.03	0.977	7
	54	500	4868	4	36	49	13.99	85.00	4.25	7	64.71	0.001	3.011	0.01	1.069	7
	55	100	999	10	90	839	16.02	0.00	4.15904	-	-	0.009	2.13	3.67	0	-
	56	100	999	10	89	730	26.93	0.00	5.38219	-	-	0.002	1.86	1.048	0	-
	57	200	1960	10	180	1705	13.01	0.00	6.85392	-	-	0.006	7.419	8.635	0	-
	58	200	1960	10	177	1618	17.45	0.00	8.99812	-	-	0	5.588	5.397	0	-
	59	500	4868	10	469	4559	6.35	0.00	3.49231	-	-	0.522	110.31	122.89	0	-
	60	500	4868	10	469	4495	7.66	0.00	4.26087	-	-	0.026	81.565	43.806	0	-

Table 3.6: Detailed results with with protection level $\Gamma = 3$.

		Original graph			Reduced graph							Time				
												Algorithm 3.6	CPLEX		Algorithm 3.4	Opt
	instance	$\|N\|$	$\|A\|$	$\|R\|-1$	$\|N\|$	$\|A\|$	Res%	Lag%	LB	UB	Gap%		OG	RG		
	1	100	955	1	14	20	7.12	90.79	136	186	36.76	0	0.145	0.01	0.009	186
	2	100	955	1	100	892	6.60	0.00	131	434	231.30	0	0.207	0.178	0.99	203
	3	200	2040	1	7	7	99.66	0.00	420	-	-	0	0.093	0.001	0.005	-
	4	200	2040	1	7	7	99.66	0.00	420	-	-	0	0.089	0.001	0.006	-
	5	500	4858	1	244	614	0.68	86.68	489.571	1140	132.86	0.001	0.667	0.165	3.654	958
	6	500	4858	1	69	111	0.86	96.85	656	1024	56.10	0	1.377	0.015	1.076	1024
	7	100	990	2	95	486	1.01	49.90	89	219	146.07	0.001	0.173	0.099	15.592	128
	8	100	990	2	62	184	4.24	77.17	89	182	104.49	0	0.196	0.103	5.723	160
	9	200	2080	2	10	13	25.58	73.80	272	390	43.38	0	0.293	0.005	0.017	390
	10	200	2080	2	195	1269	38.99	0.00	267.971	-	-	0.002	2.532	1.164	11.482	628
	11	500	4847	2	55	79	87.83	10.54	876	1923	119.52	0	1.237	0.017	1.612	1552
	12	500	4847	2	68	103	95.40	2.48	872	2447	180.62	0	2.243	0.013	2.962	2447
	13	100	990	3	95	480	1.21	50.30	89	219	146.07	0.001	0.235	0.117	413.982	128
	14	100	990	3	62	182	4.34	77.27	89	182	104.49	0	0.252	0.131	191.239	160
Inverse	15	200	2080	3	102	267	75.91	11.25	269	925	243.87	0.001	2.062	0.192	36.072	722
	16	200	2080	3	57	107	85.72	9.13	512	951	85.74	0.001	3.061	0.063	31.004	951
	17	500	4847	3	54	77	87.97	10.44	881	1923	118.27	0	2.792	0.017	16.992	1552
	18	500	4847	3	67	100	95.67	2.27	872	2447	180.62	0	3.461	0.016	47.734	2447
	19	100	990	4	9	11	2.32	96.57	103	128	24.27	0	0.341	0.007	3.065	128
	20	100	990	4	40	86	5.86	85.45	94.6444	182	92.30	0	0.315	0.025	1938.42	182
	21	200	2080	4	83	185	75.48	15.63	451	925	105.10	0.001	4.787	0.176	738.652	925
	22	200	2080	4	46	76	90.24	6.11	512	951	85.74	0	3.616	0.019	305.115	951
	23	500	4847	4	246	529	89.09	0.00	858	-	-	0.003	20.873	0.55	11254.1	2462
	24	500	4847	4	102	161	96.68	0.00	858	-	-	0.001	10.927	0.045	1445.66	-
	25	100	990	10	9	11	7.68	91.21	103	128	24.27	0	0.403	0.013	0	128
	26	100	990	10	11	16	16.97	81.41	108	182	68.52	0	0.952	0.014	0	182
	27	200	2080	10	16	19	99.09	0.00	393.98	-	-	0	2.284	0.003	0	-
	28	200	2080	10	0	0	100.00	0.00	-10000	-	-	0	1.026	0	0	-
	29	500	4847	10	107	180	92.37	3.92	869	2462	183.31	0.002	27.786	0.258	0	2462
	30	500	4847	10	71	109	97.75	0.00	858	-	-	0.001	21.443	0.037	0	-
	31	100	959	1	98	877	2.82	5.74	1.5	13	766.67	0.036	0.271	0.251	0.009	7
	32	100	959	1	92	647	3.44	29.09	2	11	450.00	0.021	0.308	0.245	0.008	8
	33	200	1971	1	194	1915	2.84	0.00	6	42	600.00	0.04	0.631	0.927	0.011	10
	34	200	1971	1	194	1915	2.84	0.00	6	42	600.00	0.022	0.196	0.2	0.01	8
	35	500	4978	1	469	4539	5.97	2.85	6	17	183.33	0.026	1.387	1.635	0.033	12
	36	500	4978	1	454	3370	6.35	25.95	6	15	150.00	0.013	0.878	0.559	0.02	9
	37	100	999	2	90	870	10.41	2.50	3.375	13	285.19	0.013	0.509	1.403	0.024	10
	38	100	999	2	90	895	10.41	0.00	3.625	-	-	0.037	1.282	1.226	0.026	13
	39	200	1960	2	94	183	11.63	79.03	5.25	9	71.43	0	0.586	0.025	0.014	9
	40	200	1960	2	184	1767	9.85	0.00	5.75	-	-	0.002	0.891	1.027	0.021	12
	41	500	4868	2	142	279	9.16	85.11	3.28571	7	113.04	0	4.751	0.119	0.022	7
	42	500	4868	2	49	83	11.65	86.65	4	6	50.00	0.001	3.987	0.01	0.016	6
	43	100	999	3	90	889	11.01	0.00	3.8202	39	920.89	0.027	0.685	0.68	0.978	14
	44	100	999	3	90	878	12.11	0.00	4.52778	-	-	0.027	1.72	1.729	0.977	21
Uniform	45	200	1960	3	184	1774	9.49	0.00	5.25	-	-	0.03	1.868	2.822	1.07	15
	46	200	1960	3	181	1734	11.53	0.00	5.75	-	-	0.018	2.637	2.551	0.973	18
	47	500	4868	3	252	658	9.74	76.75	3.3125	8	141.51	0.001	4.365	0.363	0.045	8
	48	500	4868	3	469	4553	6.47	0.00	3.65625	-	-	0.068	10.412	16.984	1.133	17
	49	100	999	4	90	889	11.01	0.00	3.86654	-	-	0.02	2.362	1.98	1.37	15
	50	100	999	4	90	878	12.11	0.00	4.57734	-	-	0.003	1.398	1.463	1.974	-
	51	200	1960	4	182	1754	10.51	0.00	5.79245	-	-	0.103	7.986	6.079	2.567	21
	52	200	1960	4	181	1718	12.35	0.00	6.70245	-	-	0.007	5.116	3.878	1.996	-
	53	500	4868	4	238	611	12.18	75.27	3.32819	8	140.37	0.001	6.91	0.482	1.011	8
	54	500	4868	4	469	4553	6.47	0.00	3.71429	-	-	0.183	25.469	24.858	4.51	22
	55	100	999	10	90	839	16.02	0.00	4.15904	-	-	0.004	4.323	3.386	0	-
	56	100	999	10	89	730	26.93	0.00	5.38219	-	-	0.001	1.24	1.187	0	-
	57	200	1960	10	180	1705	13.01	0.00	6.85392	-	-	0.003	9.695	8.178	0	-
	58	200	1960	10	177	1618	17.45	0.00	8.99812	-	-	0	2.955	3.027	0	-
	59	500	4868	10	469	4559	6.35	0.00	3.49231	-	-	0.126	93.773	101.197	0	-
	60	500	4868	10	469	4495	7.66	0.00	4.26087	-	-	0.01	33.438	34.001	0	-

Table 3.7: Detailed results with with protection level $\Gamma = 4$.

		Original graph			Reduced graph							Time				Opt
												Algorithm 3.6	CPLEX		Algorithm 3.4	
	instance	$\|N\|$	$\|A\|$	$\|R\|-1$	$\|N\|$	$\|A\|$	Res%	Lag%	LB	UB	Gap%		OG	RG		
	1	100	955	1	14	20	7.12	90.79	136	186	36.76	0	0.147	0.005	186	
	2	100	955	1	100	892	6.60	0.00	131	450	243.51	0.001	0.276	0.147	203	
	3	200	2040	1	7	7	99.66	0.00	420	-	-	0	0.102	0.001	-	
	4	200	2040	1	7	7	99.66	0.00	420	-	-	0	0.095	0.001	-	
	5	500	4858	1	260	685	0.74	85.16	490.571	1160	136.46	0.001	0.667	0.222	959	
	6	500	4858	1	90	149	0.99	95.94	657	1063	61.80	0.001	0.985	0.02	1063	
	7	100	990	2	95	486	1.01	49.90	89	219	146.07	0	0.338	0.137	128	
	8	100	990	2	62	184	4.24	77.17	89	182	104.49	0	0.177	0.112	160	
	9	200	2080	2	10	13	25.58	73.80	272	390	43.38	0	0.303	0.005	390	
	10	200	2080	2	195	1269	38.99	0.00	267.971	-	-	0.002	2.898	1.866	636	
	11	500	4847	2	55	79	87.87	10.50	876	1935	120.89	0	2.419	0.013	1935	
	12	500	4847	2	68	103	95.40	2.48	872	2454	181.42	0	5.111	0.015	2454	
	13	100	990	3	95	480	1.21	50.30	89	219	146.07	0.001	0.535	0.187	128	
	14	100	990	3	62	182	4.34	77.27	89	182	104.49	0	0.235	0.138	160	
Inverse	15	200	2080	3	104	283	76.11	10.29	269	937	248.33	0.001	3.972	0.198	727	
	16	200	2080	3	57	107	85.72	9.13	515	954	85.24	0.001	5.576	0.056	954	
	17	500	4847	3	54	77	88.01	10.40	881	1935	119.64	0.001	2.93	0.017	1935	
	18	500	4847	3	67	100	95.67	2.27	872	2454	181.42	0.001	4.463	0.017	2454	
	19	100	990	4	9	11	2.32	96.57	103	128	24.27	0	0.284	0.008	128	
	20	100	990	4	40	86	5.86	85.45	94.6444	182	92.30	0	0.399	0.029	182	
	21	200	2080	4	84	190	75.67	15.19	453	937	106.84	0.001	5.402	0.258	937	
	22	200	2080	4	46	76	90.24	6.11	515	954	85.24	0	4.99	0.02	954	
	23	500	4847	4	246	529	89.09	0.00	858	-	-	0.003	15.847	0.709	2565	
	24	500	4847	4	102	161	96.68	0.00	858	-	-	0	10.266	0.041	-	
	25	100	990	10	9	11	7.68	91.21	103	128	24.27	0	0.39	0.011	128	
	26	100	990	10	11	16	16.97	81.41	108	182	68.52	0	0.813	0.014	182	
	27	200	2080	10	16	19	99.09	0.00	393.98	-	-	0	1.107	0.003	-	
	28	200	2080	10	0	0	100.00	0.00	-	-	-	0	1.053	0	-	
	29	500	4847	10	117	201	92.24	3.61	866	2565	196.19	0.002	33.846	0.324	2565	
	30	500	4847	10	71	109	97.75	0.00	858	-	-	0.001	33.098	0.035	-	
	31	100	959	1	98	899	2.92	3.34	1.5	14	833.33	0.034	0.339	0.318	7	
	32	100	959	1	92	713	3.34	22.31	2	12	500.00	0.026	0.268	0.301	9	
	33	200	1971	1	194	1915	2.84	0.00	6	44	633.33	0.037	0.478	0.722	10	
	34	200	1971	1	194	1915	2.84	0.00	6	44	633.33	0.028	0.204	0.215	8	
	35	500	4978	1	469	4539	5.97	2.85	6	17	183.33	0.027	1.172	1.515	12	
	36	500	4978	1	460	3846	6.37	16.37	6	16	166.67	0.013	0.902	0.757	9	
	37	100	999	2	90	870	10.41	2.50	3.375	13	285.19	0.014	0.583	1.284	11	
	38	100	999	2	90	895	10.41	0.00	3.625	-	-	0.033	0.579	0.712	13	
	39	200	1960	2	94	183	11.63	79.03	5.25	9	71.43	0.001	0.454	0.026	9	
	40	200	1960	2	184	1767	9.85	0.00	5.75	-	-	0.007	0.541	0.52	12	
	41	500	4868	2	280	827	8.05	74.96	3.28571	8	143.48	0.002	2.403	0.407	8	
	42	500	4868	2	49	83	11.65	86.65	4	6	50.00	0	1.298	0.009	6	
	43	100	999	3	90	889	11.01	0.00	3.8202	42	999.42	0.024	0.73	0.739	14	
	44	100	999	3	90	878	12.11	0.00	4.52778	-	-	0.029	1.413	1.338	-	
Uniform	45	200	1960	3	184	1774	9.49	0.00	5.25	-	-	0.027	2.228	3.594	15	
	46	200	1960	3	181	1734	11.53	0.00	5.75	-	-	0.017	2.666	2.648	19	
	47	500	4868	3	252	658	9.74	76.75	3.3125	8	141.51	0.001	2.71	0.314	8	
	48	500	4868	3	469	4553	6.47	0.00	3.65625	-	-	0.222	18.119	26.599	20	
	49	100	999	4	90	889	11.01	0.00	3.86654	-	-	0.017	2.531	2.767	15	
	50	100	999	4	90	878	12.11	0.00	4.57734	-	-	0.003	1.668	1.67	-	
	51	200	1960	4	182	1754	10.51	0.00	5.79245	-	-	0.103	8.156	7.047	25	
	52	200	1960	4	181	1718	12.35	0.00	6.70245	-	-	0.006	5.707	5.617	-	
	53	500	4868	4	238	611	12.18	75.27	3.32819	8	140.37	0.001	3.688	0.461	8	
	54	500	4868	4	469	4553	6.47	0.00	3.71429	-	-	0.253	42.066	43.78	-	
	55	100	999	10	90	839	16.02	0.00	4.15904	-	-	0.005	4.221	4.014	-	
	56	100	999	10	89	730	26.93	0.00	5.38219	-	-	0.001	0.407	0.34	-	
	57	200	1960	10	180	1705	13.01	0.00	6.85392	-	-	0.004	9.97	8.086	-	
	58	200	1960	10	177	1618	17.45	0.00	8.99812	-	-	0	1.385	1.215	-	
	59	500	4868	10	469	4559	6.35	0.00	3.49231	-	-	0.085	85.584	83.247	-	
	60	500	4868	10	469	4495	7.66	0.00	4.26087	-	-	0.009	38.845	32.585	-	

Table 3.8: Detailed results with with protection level $\Gamma = 5$.

		Original graph			Reduced graph							Time				
												Algorithm 3.6	CPLEX		Algorithm 3.4	Opt
	instance	$\|N\|$	$\|A\|$	$\|R\|-1$	$\|N\|$	$\|A\|$	Res%	Lag%	LB	UB	Gap%		OG	RG		
Inverse	1	100	955	1	14	20	7.12	90.79	136	186	36.76	0	0.086	0.006	186	
	2	100	955	1	100	892	6.60	0.00	131	455	247.33	0	0.085	0.135	203	
	3	200	2040	1	7	7	99.66	0.00	420	-	-	0	0.096	0.001	-	
	4	200	2040	1	7	7	99.66	0.00	420	-	-	0	0.087	0.001	-	
	5	500	4858	1	260	685	0.74	85.16	490.571	1160	136.46	0.001	0.821	0.233	959	
	6	500	4858	1	90	149	0.99	95.94	657	1063	61.80	0	1.529	0.02	1063	
	7	100	990	2	95	486	1.01	49.90	89	219	146.07	0	0.322	0.134	128	
	8	100	990	2	62	184	4.24	77.17	89	182	104.49	0	0.301	0.133	160	
	9	200	2080	2	10	13	25.58	73.80	272	390	43.38	0	0.318	0.005	390	
	10	200	2080	2	195	1269	38.99	0.00	267.971	-	-	0.002	1.495	1.827	636	
	11	500	4847	2	55	79	87.87	10.50	876	1935	120.89	0	4.049	0.016	1935	
	12	500	4847	2	68	103	95.40	2.48	872	2454	181.42	0.001	6.177	0.018	2454	
	13	100	990	3	95	480	1.21	50.30	89	219	146.07	0.001	0.393	0.134	128	
	14	100	990	3	62	182	4.34	77.27	89	182	104.49	0	0.325	0.147	160	
	15	200	2080	3	104	283	76.11	10.29	269	937	248.33	0.001	1.849	0.295	727	
	16	200	2080	3	57	107	85.72	9.13	515	954	85.24	0	3.902	0.044	954	
	17	500	4847	3	54	77	88.01	10.40	881	1935	119.64	0	4.101	0.019	1935	
	18	500	4847	3	67	100	95.67	2.27	872	2454	181.42	0	6.894	0.02	2454	
	19	100	990	4	9	11	2.32	96.57	103	128	24.27	0	0.183	0.008	128	
	20	100	990	4	40	86	5.86	85.45	94.6444	182	92.30	0	0.565	0.028	182	
	21	200	2080	4	84	190	75.67	15.19	453	937	106.84	0.001	6.666	0.326	937	
	22	200	2080	4	46	76	90.24	6.11	515	954	85.24	0	4.482	0.019	954	
	23	500	4847	4	246	529	89.09	0.00	858	-	-	0.003	17.958	0.933	2603	
	24	500	4847	4	102	161	96.68	0.00	858	-	-	0.001	9.731	0.038	-	
	25	100	990	10	9	11	7.68	91.21	103	128	24.27	0	0.364	0.01	128	
	26	100	990	10	11	16	16.97	81.41	108	182	68.52	0	1.279	0.014	182	
	27	200	2080	10	16	19	99.09	0.00	393.98	-	-	0	1.158	0.003	-	
	28	200	2080	10	0	0	100.00	0.00	-10000	-	-	0	1.042	0	-	
	29	500	4847	10	122	208	92.24	3.47	866	2603	200.58	0.002	48.279	0.326	2603	
	30	500	4847	10	71	109	97.75	0.00	858	-	-	0	27.519	0.034	-	
Uniform	31	100	959	1	98	899	2.92	3.34	1.5	14	833.33	0.032	0.377	0.365	7	
	32	100	959	1	92	713	3.34	22.31	2	12	500.00	0.029	0.188	0.284	9	
	33	200	1971	1	194	1915	2.84	0.00	6	45	650.00	0.037	0.505	0.478	10	
	34	200	1971	1	194	1915	2.84	0.00	6	46	666.67	0.029	0.435	0.313	8	
	35	500	4978	1	469	4539	5.97	2.85	6	17	183.33	0.026	1.922	1.308	12	
	36	500	4978	1	460	3846	6.37	16.37	6	16	166.67	0.017	1.199	1.259	10	
	37	100	999	2	90	895	10.41	0.00	3.375	43	1174.07	0.022	0.551	1.386	11	
	38	100	999	2	90	895	10.41	0.00	3.625	-	-	0.066	0.737	1.569	13	
	39	200	1960	2	94	183	11.63	79.03	5.25	9	71.43	0	0.543	0.027	9	
	40	200	1960	2	184	1767	9.85	0.00	5.75	-	-	0.007	0.524	0.475	12	
	41	500	4868	2	280	827	8.05	74.96	3.28571	8	143.48	0.001	1.917	0.435	8	
	42	500	4868	2	49	83	11.65	86.65	4	6	50.00	0	0.916	0.012	6	
	43	100	999	3	90	889	11.01	0.00	3.8202	43	1025.60	0.027	1.945	1.693	14	
	44	100	999	3	90	878	12.11	0.00	4.52778	-	-	0.03	1.76	2.251	-	
	45	200	1960	3	184	1774	9.49	0.00	5.25	-	-	0.03	2.157	2.312	16	
	46	200	1960	3	181	1734	11.53	0.00	5.75	-	-	0.019	3.287	2.937	20	
	47	500	4868	3	252	658	9.74	76.75	3.3125	8	141.51	0.001	3.396	0.354	8	
	48	500	4868	3	469	4553	6.47	0.00	3.65625	-	-	0.209	23.296	20.055	20	
	49	100	999	4	90	889	11.01	0.00	3.86654	-	-	0.017	2.107	2.27	15	
	50	100	999	4	90	878	12.11	0.00	4.57734	-	-	0.003	1.961	1.648	-	
	51	200	1960	4	182	1754	10.51	0.00	5.79245	-	-	0.103	5.015	8.715	25	
	52	200	1960	4	181	1718	12.35	0.00	6.70245	-	-	0.006	5.411	5.561	-	
	53	500	4868	4	238	611	12.18	75.27	3.32819	8	140.37	0.001	3.743	0.401	8	
	54	500	4868	4	469	4553	6.47	0.00	3.71429	-	-	0.241	42.106	38.343	-	
	55	100	999	10	90	839	16.02	0.00	4.15904	-	-	0.005	1.628	1.679	-	
	56	100	999	10	89	730	26.93	0.00	5.38219	-	-	0.001	0.464	0.339	-	
	57	200	1960	10	180	1705	13.01	0.00	6.85392	-	-	0.004	11.344	9.417	-	
	58	200	1960	10	177	1618	17.45	0.00	8.99812	-	-	0.001	1.267	1.344	-	
	59	500	4868	10	469	4559	6.35	0.00	3.49231	-	-	0.094	84.854	85.81	-	
	60	500	4868	10	469	4495	7.66	0.00	4.26087	-	-	0.01	32.464	30.39	-	

Table 3.9: Results on infeasible instances in Tables 3.5 to 3.8 with increased resource limits.

			Original graph			Reduced graph					Time			
											Algorithm 3.6	CPLEX		Opt
	instance		$\lvert N\rvert$	$\lvert A\rvert$	$\lvert R\rvert - 1$	$\lvert N\rvert$	$\lvert A\rvert$	LB	UB	Gap%		OG	RG	
$\Gamma = 2$	Inverse	3	200	2040	1	4	3	292	316	8.22	0	0.099	0.004	316
		4	200	2040	1	4	3	282	329	16.67	0	0.099	0.005	329
		27	200	2080	10	11	14	234	303	29.49	0	0.797	0.013	303
		28	200	2080	10	68	166	202	536	165.35	0.001	14.663	0.289	536
		30	500	4847	10	476	2590	611	1895	210.15	0.009	18.45	10.013	1266
	Uniform	55	100	999	10	70	188	3	6	100.00	0.001	0.341	0.042	6
		56	100	999	10	45	95	3	5	66.67	0	0.237	0.027	5
		57	200	1960	10	183	1711	5	15	200.00	0.003	1.134	0.976	6
		58	200	1960	10	184	1795	5	-	-	0.002	1.631	1.368	8
		59	500	4868	10	69	93	3	4	33.33	0	2.013	0.036	4
		60	500	4868	10	58	79	3	4	33.33	0.001	2.325	0.027	4
$\Gamma = 3$	Inverse	3	200	2040	1	4	3	336	336	0.00	0	0.118	0.004	336
		4	200	2040	1	4	3	343	343	0.00	0	0.098	0.001	343
		24	500	4847	4	499	4040	611	2447	300.49	0.008	8.355	6.056	1322
		27	200	2080	10	26	43	202	390	93.07	0	6.905	0.022	390
		28	200	2080	10	68	166	203	537	164.53	0.001	10.866	0.165	537
		30	500	4847	10	499	4036	611	2447	300.49	0.014	82.068	15.519	1322
	Uniform	50	100	999	4	45	95	3	5	66.67	0	0.118	0.012	5
		52	200	1960	4	184	1795	5	35	600.00	0.001	0.494	0.586	8
		55	100	999	10	70	188	3	6	100.00	0	0.231	0.041	6
		56	100	999	10	45	95	3	5	66.67	0.001	0.246	0.027	5
		57	200	1960	10	184	1751	5	16	220.00	0.003	0.992	1.015	6
		58	200	1960	10	184	1795	5	-	-	0.002	1.598	1.598	8
		59	500	4868	10	69	93	3	4	33.33	0	1.294	0.035	4
		60	500	4868	10	138	243	3	5	66.67	0.003	2.055	0.157	4
$\Gamma = 4$	Inverse	3	200	2040	1	4	3	336	336	0.00	0	0.171	0.001	336
		4	200	2040	1	4	3	343	343	0.00	0	0.103	0.002	343
		24	500	4847	4	499	4042	611	2454	301.64	0.008	10.901	8.207	1371
		27	200	2080	10	26	43	202	390	93.07	0	8.454	0.022	390
		28	200	2080	10	68	166	203	537	164.53	0.001	16.482	0.19	537
		30	500	4847	10	499	4038	611	2454	301.64	0.014	21.344	16.645	1411
	Uniform	44	100	999	3	45	95	3	5	66.67	0	0.098	0.011	5
		50	100	999	4	45	95	3	5	66.67	0	0.126	0.014	5
		52	200	1960	4	184	1795	5	-	-	0.001	1.013	0.713	8
		54	500	4868	4	147	260	3	5	66.67	0.001	0.601	0.095	4
		55	100	999	10	70	188	3	6	100.00	0.001	0.291	0.041	6
		56	100	999	10	90	895	3	27	800.00	0.009	3.083	1.422	7
		57	200	1960	10	184	1795	5	26	420.00	0.004	0.877	0.818	6
		58	200	1960	10	184	1795	5	-	-	0.04	8.243	11.98	12
		59	500	4868	10	69	93	3	4	33.33	0.001	1.653	0.036	4
		60	500	4868	10	469	4572	3	28	833.33	0.022	7.416	3.957	4
$\Gamma = 5$	Inverse	3	200	2040	1	4	3	336	336	0.00	0	0.12	0.001	336
		4	200	2040	1	4	3	343	343	0.00	0	0.1	0.001	343
		24	500	4847	4	499	4042	611	2454	301.64	0.008	6.477	10.712	1411
		27	200	2080	10	26	43	202	390	93.07	0	8.56	0.024	390
		28	200	2080	10	68	166	203	537	164.53	0.001	16.379	0.17	537
		30	500	4847	10	499	4038	611	2454	301.64	0.014	20.297	16.473	1411
	Uniform	44	100	999	3	45	95	3	5	66.67	0	0.113	0.01	5
		50	100	999	4	45	95	3	5	66.67	0.001	0.125	0.012	5
		52	200	1960	4	184	1795	5	-	-	0.002	1.276	1.223	8
		54	500	4868	4	147	260	3	5	66.67	0.001	0.78	0.106	4
		55	100	999	10	70	188	3	6	100.00	0	0.251	0.044	6
		56	100	999	10	90	895	3	27	800.00	0.009	2.768	1.363	7
		57	200	1960	10	184	1795	5	26	420.00	0.004	0.931	0.876	6
		58	200	1960	10	184	1795	5	-	-	0.04	8.218	10.999	12
		59	500	4868	10	69	93	3	4	33.33	0.001	1.429	0.037	4
		60	500	4868	10	469	4572	3	28	833.33	0.022	7.435	3.859	4

Chapter 4

Robust vehicle routing problem with time windows

4.1. Introduction

The vehicle routing problem (VRP) is one of the most important problems faced by decision makers in transportation, distribution, and logistics. Since the early 1950s, the problem has led to several variants with different restrictions, such as vehicle capacity, service time windows, multiple tasks at customers, and service priorities. Among all these variants, the capacitated vehicle routing problem (CVRP) and the vehicle routing problem with time windows (VRPTW) are well studied in the literature. The CVRP concerns serving a set of customers with a number of vehicles starting and ending at a depot such that the demand of each customer is served by exactly one vehicle, the capacity of each vehicle is respected, and the total cost of routing the vehicles is minimized. Recent surveys on

CVRP can be found in Baldacci et al. [2010], Cordeau et al. [2007], Golden et al. [2008] and Laporte [2009] and Toth and Vigo [2002]. The VRPTW generalizes the CVRP by requiring the service at a customer to be within specified time windows. Thorough reviews on VRPTW are presented in Feillet [2010], Desaulniers et al. [2008], Kallehauge et al. [2005] and Cordeau et al. [2001a]. In real operations, variability of travel times and demand has a great impact on the reliability and cost of the operational plan [Mey]. Stochastic and robust optimizations are the two main frameworks to take uncertainty into account. A large volume of literature on the stochastic VRP is reviewed by Berhan et al. [2014], whereas VRP models using robust optimization are scant. In this chapter, we propose a robust VRPTW model that takes into account demand uncertainty and provides a solution whose robustness is controlled by the decision maker.

The remainder of the chapter is organized as follows. In Section 4.2, we review the literature on the VRPTW and its variants under uncertainty. Section 4.3 defines the robust VRPTW with demand uncertainty where the uncertainty support is based on cardinality constrained sets. Section 4.4 develops a branch-and-price-and-cut algorithm for the problem. We test our model and solution methods in Section 4.5 and conclude the chapter in Section 4.6.

4.2. Literature review

4.2.1 The deterministic VRPTW

A variety of solution methodologies have been proposed for the VRPTW. Most exact solution methods fall into the framework of branch-and-price. The branch-and-price scheme implements column generation (CG) to derive a lower bound at each node of the branch-and-bound tree. In the context of the VRPTW, each feasible route corresponds to a column covering a subset of customers in a set partitioning formulation. Since there is an exponential number of feasible routes, only a subset of all possible columns is explicitly included in the formulation. The linear programming (LP) relaxation defined over this subset of columns is called the restricted Dantzig-Wolfe (DW) master problem. New columns with negative reduced costs are generated by solving an elementary shortest path problem with resource constraints (ESPPRC) in the subproblem. The objective function of the subproblem is modified by the dual variables from the restricted DW master problem. When ESPPRC fails to find any columns with negative reduced cost, the best lower bound is found. Due to the NP-hard nature of ESPPRC, the elementary requirement can be relaxed to allow cycles. This accelerates the solution of the subproblem at the price of a weaker lower bound. Therefore, different methods have been proposed to improve the lower bound. One direction is to improve the quality of the generated routes. Popular techniques include forbidding short cycles [Desrochers et al., 1992; Irnich and Villeneuve, 2006], and imposing the elementary path requirement on a subset of nodes [Desaulniers et al., 2008; Baldacci et al., 2012; Righini and Salani, 2006; Boland et al., 2006]. Another direction is to add valid inequalities to strengthen the restricted DW master problem, resulting in a branch-and-price-and-cut approach [Kohl et al., 1999; Cook and Rich, 1999; Jepsen et al., 2008; Desaulniers et al., 2008]. Besides strengthening the lower bound, many strategies are devised to speed up the solution process. Heuristics can be used to obtain tight upper bounds and to warm start the CG process, or as an alternative approach to solve large-scale instances. Bräysy and Gendreau [2005a,b] survey various studies on route construction, local search and metaheuristics for the VRPTW. Furthermore, stabilization techniques that control the changes in dual space can lead to faster convergence and fewer negative cycles in the subproblem [Kallehauge et al., 2006].

4.2.2 The stochastic and robust VRPTW

The two main methods to incorporate uncertainty are stochastic and robust optimization.

Stochastic optimization and the stochastic VRPTW

Stochastic optimization assumes that the uncertain parameters follow a probability distribution that is known or that can be estimated. Stochastic optimization looks for a solution that is feasible under all possible parameter realizations and performs best on average. Two basic types of models in stochastic optimization are recourse models and chance-constrained models. The decision making process in recourse models is divided into two or more stages where the decision at each stage is determined based on the information available up to the current stage. These decisions jointly optimize an objective that evaluates the quality of the solution. For example, a two-stage recourse model first determines a here-and-now decision at the first stage before the uncertain parameters are realized. In the second stage, when parameters become known, it provides a recourse decision accordingly. The chance-constrained model generates a here-and-now decision that satisfies all constraints with a pre-specified probability. The interested readers are referred to Birge and Louveaux [2011], Shapiro et al. [2009] and Shapiro and Philpot.

The parameters that are assumed to be random in the stochastic VRPTW include stochastic demand [Mak and Guo, 2004], customer presence [Mak and Guo, 2004], travel time [Mak and Guo, 2004; Ando and Taniguchi, 2006; Russell and Urban, 2008; Li et al., 2010; Taş et al., 2013, 2014; Errico et al., 2013], and service time [Li et al., 2010; Errico et al., 2013]. The solution methodologies include branh-and-price-and-cut [Errico et al., 2013; Taş et al., 2014], genetic algorithms [Mak and Guo, 2004; Ando and Taniguchi, 2006], and tabu search [Russell and Urban, 2008; Li et al., 2010; Taş et al., 2013]. A comprehensive literature review is provided by Errico et al. [2013] and Taş et al. [2013].

Robust optimization and the robust VRPTW

Another framework for modeling uncertainty is robust optimization. It differs from stochastic optimizations in that a probabilistic description of the uncertainty is unnecessary. Instead, the uncertainty is described by a set of possible realizations, called the support of the uncertainty. The optimization model finds a minimum cost solution that stays feasible under all realizations. Ben-Tal et al. [2009] and Bertsimas et al. [2011] provide comprehensive reviews on robust optimization.

To the best of our knowledge, there is only one study of VRPTW under the framework of robust optimization. Agra et al. [2013] study a VRPTW with uncertain travel times. They develop two robust counterparts of the VRPTW formulations based on resource inequalities and path inequalities, respectively. They show that it suffices to consider only the scenarios corresponding to a subset of the extreme points of the support. Under cardinality constrained support, the number of necessary scenarios depends on the topology of the underlying network and the protection level. To solve the robust model based on the resource inequality formulation, the initial mixed integer programming (MIP) formulation explicitly formulates an arbitrary subset of the necessary scenarios. The solution of the MIP is optimal to the robust problem if it is feasible for the remaining scenarios. Otherwise, some infeasible scenarios are included in the model, expanding the sets of variables and constraints. For the robust model based on the path inequality formulation, the authors propose a dynamic recursive function to verify the feasibility of a solution. The optimal solution is obtained by gradually appending tournament inequalities. They apply the two robust models and their solution methodologies to a maritime transportation problem and report on the computational results of instances with a maximum of 50 nodes in the network.

Beside the work by Agra et al. [2013], there are only three studies of VRP under the framework of robust optimization. Sungur et al. [2008] is the first to propose three robust CVRP models with uncertain customer demands under ellipsoidal, hypercube and convex hull supports, respectively. They show that it suffices to consider an extreme point in each support that corresponds to the worst case scenario. As a result, the robust CVRP models reduce to the deterministic equivalents and can be solved by the solution methods for deterministic CVRP. The models are tested on instances with 49 customers. They briefly mention a possible extension to the VRPTW where the uncertainties of travel time and customer demand are defined using a cardinality constrained support. Sungur et al. [2010] apply the worst case based model to a courier delivery problem with random customer service times and soft time windows. The most recent work by Gounaris et al. [2013] focuses on CVRP with uncertain customer demands. They propose a generalized hypercube support that is encoded by the intersection of a hyper-rectangle with a halfspace that imposes an upper bound on total customer demand, and show that an equivalent deterministic CVRP can be derived. The robust counterparts of the two-index vehicle flow, the Miller-Tucker-Zemlin, the one-commodity flow, the two-commodity flow and the vehicle assignment formulations are provided and reformulated as MIP problems under the generalized hypercube support. Moreover, when the demand support is a set of disjoint generalized hypercube supports or a convex hull resulting from an affine transformation of a hypercube, the authors derive analytic solutions to evaluate the total demand of a subset of customers in the worst case. These analytic solutions are used to find rounded capacity inequalities. Using these models, they are able to optimally solve instances with no more than 50 customers and obtain an optimality gap less than 5% for instances with 53 to 80 customers.

4.2.3 Contributions

In this chapter, we present a robust VRPTW model where customer demands are random and the support of the uncertainty is based on cardinality constrained sets. To be specific, given a feasible route, the support is a cardinality constrained set where a specified protection level limits the maximum number of variations on the demands of the customers served on a route. The union of these supports defines the support of the entire model. Thus, given an arbitrary subset of customers, the total demand in the worst case is a function of the partitioning of all customers and can not be derived analytically. This characteristic makes the problem more challenging than those studied in Gounaris et al. [2013] where such maximum is independent of the partition. In summary, our contributions in this chapter are the following:

- We develop a robust counterpart for the VRPTW where the support of customer demand is based on cardinality constrained sets. A branch-and-price-and-cut algorithm is developed to solve the problem.
- We propose an exact approach to tackle the subproblem which is a robust counterpart of ESPPRC. This approach is based on the sequential algorithm proposed by Bertsimas and Sim [2003] and generalized by Lu and Gzara [2014a].
- To improve the lower bound at each node, we propose a strategy to separate valid inequalities based on the characteristics of the support for a given set of customers.
- We carry out computational experiments on a set of instances derived from the Solomon benchmark instances.

4.3. Problem definition

The robust VRPTW is described as follows. Let $G = (V, A)$ be a complete directed graph, where $V = \{v_0, ..., v_n\}$ is the set of nodes and A is the set of arcs represented by $(v_i, v_j) \in A$. The depot is denoted by v_0, and the customer set $\{v_1, .., v_n\}$ is denoted by V_C. A fleet of vehicles, denoted by set K, is available to serve the customers. Each vehicle $k \in K$ has a capacity Q. Each customer $v_i \in V_C$ has a time window $[a_i, b_i]$ and a positive service time s_i. The demand of customer v_i, denoted by d_i, randomly varies in the interval $[\underline{d}_i, \underline{d}_i + h_i]$, where $\underline{d}_i$ is the nominal demand of the customer under a deterministic setting and h_i is the maximum deviation from the nominal demand value. Let $\mathcal{D}$ be the support of demand uncertainty and $\boldsymbol{d} \in \mathcal{D}$ the vector composed of $\{d_1, ..., d_n\}$. A cost c_{ij} and a travel time t_{ij} are defined for every arc $(v_i, v_j) \in A$. Vehicles must begin and end their routes at v_0 within the time horizon $[a_0, b_0]$. Serving a customer must begin within its time window, but arriving earlier is allowed with no penalty. To protect against demand uncertainty, we aim to maintain the route feasibility in the following strategy: given a subset of customers $V_C^k \subseteq V_C$ assigned to vehicle $k \in K$, the capacity Q is enough to serve V_C^k as long as no more than Γ customers in V_C^k have the actual demands deviate from the nominal values. The robust VRPTW looks for a minimum cost set of at most $|K|$ routes visiting each customer exactly once with respect to the previous constraints. The problem is formulated as follows:

[R-VRPTW]

$$\min \sum_{k=1}^{|K|} \sum_{(v_i,v_j)\in A} c_{ij}x_{ij}^k \tag{4.1}$$

$$\text{s.t.} \sum_{k=1}^{|K|} \sum_{i:(v_i,v_j)\in A} x_{i,j}^k = 1 \quad j = 1, ..., n \tag{4.2}$$

$$\sum_{i:(v_i,v_j)\in A} x_{ij}^k - \sum_{i:(v_j,v_i)\in A} x_{ji}^k = 0 \quad j = 0, ..., n, k = 1, ..., |K| \tag{4.3}$$

$$\sum_{i:(v_0,v_i)\in A} x_{0i}^k \leq 1 \quad k = 1, ..., |K| \tag{4.4}$$

$$\sum_{(v_i,v_j)\in A, v_i\neq v_0} d_i x_{ij}^k \leq Q \quad k = 1, ..., |K| \quad \forall \boldsymbol{d} \in \mathcal{D} \tag{4.5}$$

$$u_i^k + s_i + t_{ij} - u_j^k + Mx_{ij}^k \leq M \quad \forall (v_i, v_j) \in A, v_i \neq v_0, k = 1, ..., |K| \tag{4.6}$$

$$u_i^k + s_i + t_{i0} - b_0 + Mx_{i0}^k \leq M \quad (v_i, v_0) \in A, k = 1, ..., |K| \tag{4.7}$$

$$a_i \leq u_i^k \leq b_i \quad i = 0, ..., n, k = 1, ..., |K| \tag{4.8}$$

$$x_{ij}^k \in \{0, 1\}, \quad \forall (v_i, v_j) \in A, k = 1, ..., |K| \tag{4.9}$$

where x_{ij}^k equals 1 if arc (v_i, v_j) is used by vehicle k and 0 otherwise. u_i^k is the starting time of service of vehicle k at customer v_i, if v_i is assigned to vehicle k. s_0^k is the departure time of vehicle k. The objective function (4.1) minimizes the total cost of all vehicle routes. Constraints (4.2) ensure that each customer is visited exactly once. Constraints (4.3) and (4.4) are flow balance constraints. Constraints (4.5) enforce that the total demand for each vehicle in any realization does not exceed the capacity. Constraints (4.6) and (4.7) require that service starts within the time window at each customer. For a set of customers $\bar{V}_C$, define $\boldsymbol{d}(\bar{V}_C)$ as the realization where $d_i = \underline{d}_i$ for $v_i \in V_C \setminus \bar{V}_C$ and $d_i = \underline{d}_i + h_i$ for $v_i \in \bar{V}_C$.

Let $B = \{\bar{V}_C | \bar{V}_C \subseteq V_C, |\bar{V}_C| = \Gamma\}$ be the set of all combinations of Γ customers from V_C. According to Agra et al. [2013], under cardinality constrained support, defining $\mathcal{D}$ as $\bigcup_{\bar{V}_C \in B} \{\boldsymbol{d}(\bar{V}_C)\}$ guarantees the feasibility of a solution against all possible realizations. We propose an equivalent formulation under cardinality constrained uncertainty as follows:

[R1-VRPTW]

$$\min \sum_{k=1}^{|K|} \sum_{(v_i,v_j)\in A} c_{ij} x_{ij}^k \tag{4.10}$$

$$\text{s.t. } (4.2), (4.3), (4.4), (4.6), (4.7), (4.8), (4.9) \tag{4.11}$$

$$\sum_{(v_i,v_j)\in A} \underline{d}_i x_{ij}^k + \beta(\boldsymbol{x}^k, \Gamma) \leq Q \quad k = 1, ..., |K| \tag{4.12}$$

where $\boldsymbol{x}^k \in \{0,1\}^{|A|}$ in (4.12) is the vector composed of the decision variable set $\{x_{ij}^k | (v_i, v_j) \in A\}$. The cardinality constrained set support is formulated into the constraint by adding a function $\beta(\boldsymbol{x}^k, \Gamma) = \max_{\{\bar{V}_C | \bar{V}_C \subseteq V_C, |\bar{V}_C| \leq \Gamma\}} \sum_{v_i \in \bar{V}_C} h_i \sum_{j:(v_i,v_j)\in A} x_{ij}^k$, where $\bar{V}_C$ is a subset of customers with a cardinality of at most Γ such that the customers in $\bar{V}_C$ are visited by vehicle k and the cumulative sum of their demand deviations is maximized. Given the definition of $\beta(\boldsymbol{x}^k, \Gamma)$, the capacity of vehicle k is guaranteed to cover any demand realization as long as no more than Γ of the customers assigned to vehicle k change their demands to the maximum.

4.4. Solution methodology

4.4.1 Set partitioning formulation and branch-and-price

As the inner maximization $\beta(\boldsymbol{x}^k, \varGamma)$ in constraint (4.12) is imposed on each vehicle, [R1-VRPTW] can be formulated as a set covering problem. Let $\varOmega$ be the set of feasible routes that satisfy constraints (4.3), (4.4), (4.12), (4.6), (4.7), (4.8) and (4.9). Let $\lambda_{ijp} = 1$ if route $p \in \varOmega$ uses arc (v_i, v_j) and 0 otherwise. Let $\alpha_{ip} = \sum_{\{v_j \in V_C | (v_i, v_j) \in A\}} \lambda_{ijp}$ that equals 1 if route p visits customer v_i and 0 otherwise. Let $c_p = \sum_{(v_i, v_j) \in A} \lambda_{ijp} c_{ij}$ be the cost of route p. The robust VRPTW can be formulated as the following set covering model:

[P]

$$\min \sum_{p \in \varOmega} c_p \theta_p \tag{4.13}$$

$$\text{s.t.} \sum_{p \in \varOmega} \alpha_{ip} \theta_p \geq 1 \quad i = 1, ..., n \tag{4.14}$$

$$\sum_{p \in \varOmega} \theta_p \leq |K| \tag{4.15}$$

$$\theta_p \in \mathbb{N}, \quad \forall p \in \varOmega \tag{4.16}$$

where θ_p is a decision variable that equals 1 if route p is selected in the solution and 0 otherwise.

The LP relaxation of [P] is called the Dantzig-Wolfe (DW) master problem. When the DW master problem is defined over a subset $\bar{\varOmega} \subseteq \varOmega$, we refer to the resulting problem as the restricted DW master problem, denoted by $M(\bar{\varOmega})$. [P] and the DW master problem

$M(\Omega)$ require explicit enumeration of all feasible routes, which is almost impossible for a meaningful size problem. Therefore, $M(\Omega)$ is solved using column generation (CG) and the resulting lower bound is embedded within the branch-and-bound, resulting in a branch-and-price algorithm.

CG is a procedure that solves $M(\Omega)$ by implicitly considering all feasible routes. The principle can be described as follows. After solving $M(\bar{\Omega})$, if there does not exist any feasible route in $\Omega \backslash \bar{\Omega}$ with negative reduced cost, then the linear relaxation of [P] is implicitly solved. Otherwise, we solve a subproblem to find one or more routes with negative reduced costs. Let μ_i be the dual variable associated with constraint (4.14) and μ_0 be the dual variable of constraint (4.15). The subproblem is a robust ESPPRC as follows:

[R-ESPPRC]

$$\min \sum_{(v_i,v_j)\in A} c_{ij}x_{ij}^k - \sum_{i:v_i\in V} \mu_i \tag{4.17}$$

$$\text{s.t.} \sum_{i:(v_i,v_j)\in A} x_{ij}^k - \sum_{i:(v_i,v_j)\in A} x_{ji}^k = 0 \quad j = 0, ..., n \tag{4.18}$$

$$\sum_{j:(v_i,v_j)\in A} x_{ij}^k \leq 1 \quad i = 0, ..., n \tag{4.19}$$

$$\sum_{(v_i,v_j)\in A, v_i\neq v_0} \underline{d}_i x_{ij}^k + \beta(\boldsymbol{x}^k, \Gamma) \leq Q \tag{4.20}$$

$$u_i^k + s_i + t_{ij} - u_j^k + Mx_{ij}^k \leq M \quad \forall(v_i, v_j) \in A, v_i \neq v_0 \tag{4.21}$$

$$u_i^k + s_i + t_{i0} - b_0 + Mx_{i0}^k \leq M \quad (v_i, v_0) \in A \tag{4.22}$$

$$a_i \leq u_i^k \leq b_i \quad i = 0, ..., n \tag{4.23}$$

$$y_{ij}^k \in \{0, 1\}, \quad \forall(v_i, v_j) \in A \tag{4.24}$$

To avoid cycles, constraint (4.19) is needed in [R-ESPPRC] while it is unnecessary in [R1-VRPTW] due to constraint (4.2). The solution of [R-ESPPRC] corresponds to an elementary path with minimum reduced cost. When the elementary requirement is relaxed, α_{ip} becomes the number of times path p visits customer v_i. The resulting columns in the restricted DW master problem lead to a weaker lower bound. [R-ESPPRC] differs from the deterministic ESPPRC in that constraint (4.20) includes an inner maximization. We provide an algorithm that solves the robust ESPPRC as a series of deterministic ESPPRC's in Section 4.4.2.

4.4.2 Solution method for robust ESPPRC

The solution method for [R-ESPPRC] is not straightforward given the formulation with the inner maximization. Before presenting the solution approach, we first show that [R-ESPPRC] has an equivalent MIP reformulation. The reformulation uses similar techniques as those used in Chapters 2 and 3.

For ease of exposition, we omit the vehicle index k in [R-ESPPRC] and let X be the feasible set defined by flow balance constraints (4.18)-(4.19) and (4.21)-(4.24). Let $\bar{\boldsymbol{c}}$ be the vector of $\bar{c}_{ij} = c_{ij} - \mu_i$ and $\boldsymbol{x}$ be the vector of x_{ij}. Define $x_i = \sum_{j:(v_i,v_j)\in A, v_i \neq v_0} x_{ij}$ for $i = 1, ..., n$. Without loss of generality, we sort the customers using index $e = 1, ..., n$ such that $h^1 \geq h^2 \geq ... \geq h^n \geq h^{n+1}$, and $h^{n+1} = 0$. Let x^e, $\underline{d}^e$ and h^e be replicas of x_i, $\underline{d}_i$ and h_i when nominal customer demand $\underline{d}_i$ has the eth highest variation. $\boldsymbol{\chi}$ and $\underline{\boldsymbol{d}}$ are vectors of x^e and $\underline{d}^e$, respectively. Define w^e as a binary variable that takes value 1 if $h^e x^e$ is selected to maximize $\beta(\boldsymbol{x}, \Gamma)$ and 0 otherwise. Then [R-ESPPRC] is rewritten as follows:

[R2-ESPPRC]

$$
\begin{aligned}
Z = \ & \min \ \bar{\boldsymbol{c}}^\top \boldsymbol{x} \\
& \text{s.t. } \boldsymbol{x} \in X \\
& \qquad \underline{\boldsymbol{d}}^\top \boldsymbol{\chi} + \max_{\substack{0 \leq w^e \leq 1 \\ \sum_{e=1}^n w^e \leq \Gamma}} \sum_{e=1}^n h^e x^e w^e \leq Q
\end{aligned}
$$

In the robust term $\max_{0 \leq w^e \leq 1, \sum_{e=1}^n w^e \leq \Gamma} \sum_{e=1}^n h^e x^e w^e$, let q_0 be the dual variable of constraint $\sum_{e=1}^n w^e \leq \Gamma_r$, and q^e be the dual variable of constraint $0 \leq w^e \leq 1$. An equivalent MIP formulation is as follows:

[R3-ESPPRC]

$$Z = \min \bar{\boldsymbol{c}}^\top \boldsymbol{x} \tag{4.25}$$

$$\text{s.t. } \boldsymbol{x} \in X$$

$$\underline{\boldsymbol{d}}^\top \boldsymbol{\chi} + \Gamma q_0 + \sum_{e=1}^{n} q^e \leq Q \tag{4.26}$$

$$q_0 + q^e \geq h^e x^e \qquad e = 1, ..., n \tag{4.27}$$

$$q^e, q_0 \geq 0 \qquad e = 1, ..., n. \tag{4.28}$$

The robust terms in [R2-ESPPRC] are transformed into a minimization problem by duality. The objective function of the resulting dual problem is captured by (4.25) and constraints (4.26). The constraints of the resulting dual minimization problems are added as constraints (4.27) and (4.28). The latter may be solved directly using any deterministic MIP solver. However, Theorem 3.1 provides an alternative approach that solves [R-ESPPRC] by solving a series of $n+1$ problems:

$$Z = \min_{f=1,...,n+1} Z^f \tag{4.29}$$

where Z^f is the minimum objective value of problem $[\mathrm{I}^f]$ defined as

[I^f]

$$Z^f = \min \bar{\boldsymbol{c}}\boldsymbol{x} \tag{4.30}$$

$$\text{s.t. } \boldsymbol{x} \in X \tag{4.31}$$

$$\boldsymbol{d}^\top \boldsymbol{\chi} + U^f(\boldsymbol{x}) \leq Q, \tag{4.32}$$

and

$$U^f(\boldsymbol{x}) = \begin{cases} \Gamma h^1 & \text{if } f = 1, \\ \Gamma h^f + \sum_{e=1}^{f} (h^e - h^f) x^e & \text{if } f = 2, \ldots, n, \\ \sum_{e=1}^{n} h^e x^e & \text{if } f = n+1. \end{cases}$$

The sequential approach to solve [R-ESPPRC] is presented in Algorithm 4.1.

Algorithm 4.1 The sequential algorithm for problem [R-ESPPRC].

for $h^f = h^1, ..., h^{n+1}$ and $h^f \neq h^{f+1}, f = 1, ..., n$ **do**
  Set $d_i = \underline{d}_i + h_i - h^f$ for all $v_i \in V$ such that $h_i > h^f$.
  Solve the resulting ESPPRC and find the optimal path $\overline{\boldsymbol{x}}$.
  if the cost of $\overline{\boldsymbol{x}}$ is less than current best solution **then**
    Update current best solution.
  end if
end for

Algorithm 4.1 creates a sequence of deterministic ESPPRC and solves the resulting deterministic problems using a label-setting algorithm. Since solving ESPPRC is not the focus of this chapter, we briefly review the concepts and techniques involved in the label-setting algorithm. A label is a data container that records the information of a feasible partial path from source node o in the graph. The information includes the accumulated consumption of resource, visited nodes, and cost of the partial path. A partial path p_1 dominates p_2 if p_1 and p_2 end at the same node, the cost of p_1 is no more than the cost of p_2, and any feasible extension from p_2 is also a feasible extension from p_1. Given such dominance relation, ignoring all the paths extended from p_2 does not exclude the minimum cost path from source to sink. The algorithm starts extending the initial label from the source node and stops when all the labels have been either extended or verified as dominated. Since the number of labels created can scale exponentially for large instances, several techniques are used to accelerate the algorithm. One technique is to simultaneously carry out a forward extension from the source node and a backward extension from the sink node, and combining partial paths at a splitting point [Righini and Salani, 2006]. Feillet et al. [2004] suggest rigorously marking the unreachable node in a label, which helps trigger more dominance between labels. The interested readers are referred to Righini and Salani [2006], Dumitrescu and Boland [2003], Feillet et al. [2004] and Chabrier [2006] for a detailed discussions on the label-setting algorithm.

4.4.3 Valid inequalities for the robust VRPTW

Cutting planes have contributed greatly to the success of solving difficult CVPR and VRPTW instances. A commonly used inequality for CVRP is the rounded capacity inequality, while the successful inequalities for the VRPTW are k-path inequalities [Kohl et al., 1999] and subset row inequalities [Jepsen et al., 2008]. In this section, we propose a new separation strategy to generate valid inequalities that enhance the rounded capacity inequality for the deterministic CVRP, and extend the separation of k-path inequalities with $k = 2$ to the robust VRPTW.

Rounded capacity inequalities

One of the inequalities proven effective in CVRP is the rounded capacity inequality (RCI). The idea of the RCI can be described as follows. Given a subset of nodes $S \subseteq V_C$, let $\delta(S)^- \subset A$ be the subset of arcs (v_i, v_j) entering into S, i.e. $v_i \notin S$ and $v_j \in S$. Let $x(S) = \sum\limits_{(v_i,v_j)\in\delta(S)^-} x_{ij}$ be the total flow entering S and $r(S) = \lceil \sum_{v_i \in S} \underline{d}_i / Q \rceil$ be a lower bound on the minimum number of vehicles needed to serve the customers in S satisfying the capacity constraints. The RCI is defined as:

$$x(S) \geq r(S) \tag{4.33}$$

The RCI is rarely used in the VRPTW since the time window constraints lead to the feasible routes with relatively low utilization of vehicle capacity, which, as a result, reduces the situation where a solution of $M(\bar{\Omega})$ violates the RCI. However, in the robust VRPTW, the highest Γ demand variations along a feasible route tend to increase the utilization of vehicle capacity. Therefore, the rounded capacity inequality can be promising in this case. To consider the variations, we propose a new separation strategy developed for the robust VRPTW based on the RCI. The resulting inequalities are named as enhanced RCI. The idea is as follows. Given a fractional solution $\bar{\boldsymbol{x}}$ and a target number of L vehicles to use, we want to construct a candidate S such that S can be partitioned into L subsets $S_l, l = 1, .., L$, where $\sum\limits_{v_j \in S_l} \underline{d}_j + \max\limits_{\{\tilde{S} | \tilde{S} \subseteq S_l, |\tilde{S}| \leq \Gamma\}} \sum\limits_{v_j \in \tilde{S}} h_j > Q$ for each S_l, and $\bar{x}(S)$ is minimized. This is achieved by solving the following MIP:

[Candidate Separation]

$$\bar{x}(S) = \min \sum_{(v_i,v_j)\in A} y_{ij} \tag{4.34}$$

$$\text{s.t.} \sum_{v_j\in V_C} \underline{d}_j \vartheta_{jl} + \sum_{v_j\in V_C} h_j \tau_{jl} > Q \quad l = 1, ..., L \tag{4.35}$$

$$\tau_{jl} \leq \vartheta_{jl} \quad j = 1, ..., n, l = 1, ..., L \tag{4.36}$$

$$\sum_{v_j\in V_C}^{n} \tau_{jl} \leq \Gamma \quad l = 1, ..., L \tag{4.37}$$

$$\sum_{l=1}^{L} \vartheta_{jl} \leq 1 \quad j = 1, ..., n \tag{4.38}$$

$$\bar{x}_{ij} \sum_{l=1}^{L} (\vartheta_{jl} - \vartheta_{il}) \leq y_{ij} \quad \forall (v_i, v_j) \in A, v_i \neq v_0 \tag{4.39}$$

$$\bar{x}_{0j} \sum_{l=1}^{L} \vartheta_{jl} \leq y_{0j} \quad \forall (v_0, v_j) \in A \tag{4.40}$$

$$\vartheta_{jl} \in \{0, 1\}, \tau_{jl} \in \{0, 1\} \quad j = 1, ..., n, l = 1, ..., L \tag{4.41}$$

where ϑ_{jl} equals 1 if customer v_j is assigned to vehicle l and 0 otherwise. τ_{jl} equals 1 if the demand variation h_j is included in the robust total demand of the customers served by vehicle l. y_{ij} is a continuous variable indicating the flow on arc (v_i, v_j). Constraints (4.35) ensure that the constructed S can be partitioned into L subsets such that the total robust demand of customers in each subset is more than vehicle capacity. Constraints (4.36) ensure the demand deviation of customer v_j is only considered by vehicle l if $\vartheta_{jl} = 1$. Constraints (4.37) limit the number of deviations considered to be no more than Γ. Constraints (4.38) enforce each customer is included in S at most once. Constraints (4.39) and (4.40) ensure that $\bar{y}_{ij}$ is accounted in the total flow entering S if $v_i \notin S$ and $v_j \in S$. The objective is to minimize the total flow into S.

The feasible solution of [Candidate Separation] identifies a set S and a partition of S such that the total robust demand of the customers in each subset can not be covered by a single vehicle. However, it is possible that a different partition on S can satisfy each vehicle capacity. Therefore, to guarantee such a feasible partition does not exist, the following problem is solved:

[Guarantee]

$$NV(S) = \min \sum_{l=1}^{L} z_l \tag{4.42}$$

$$\text{s.t.} \sum_{v_j \in S} \underline{d}_j \sigma_{jl} + \max_{\{\tilde{S} | \tilde{S} \subseteq S, |\tilde{S}| \leq \Gamma\}} \sum_{v_j \in \tilde{S}} h_j \sigma_{jl} \leq Q z_l \quad l = 1, ..., L \tag{4.43}$$

$$\sum_{l=1}^{L} \sigma_{jl} = 1 \quad \forall v_j \in S \tag{4.44}$$

$$\sigma_{jl} \leq z_l \quad l = 1, .., L, \forall v_j \in S \tag{4.45}$$

$$\sigma_{jl} \in \{0, 1\} \quad \forall v_j \in S, l = 1, ..., L. \tag{4.46}$$

[Guarantee] is a robust binpacking problem based on cardinality constrained support, where the item size varies in an interval and the number of items allowed to vary is limited by Γ. By applying the MIP reformulation proposed by Bertsimas and Sim [2003], the equivalent MIP is:

[Guarantee MIP]

$$
\begin{aligned}
NV(S) = \ & \min \sum_{l=1}^{L} z_l && && (4.47)\\
& \text{s.t.} \sum_{v_j \in S} \underline{d}_j \sigma_{jl} + \Gamma \eta_l + \sum_{v_j \in S} \varrho_{jl} \le Q z_l && l = 1, ..., L && (4.48)\\
& \sum_{l=1}^{L} \sigma_{jl} = 1 && \forall v_j \in S && (4.49)\\
& \sigma_{jl} \le z_l && l = 1, .., L, \forall v_j \in S && (4.50)\\
& \eta_l + \varrho_{jl} \ge d_j \sigma_{jl} && l = 1, .., L, \forall v_j \in S && (4.51)\\
& \sigma_{jl} \in \{0, 1\} && \forall v_j \in S, l = 1, ..., L && (4.52)
\end{aligned}
$$

If [Candidate Separation] finds an S such that $\bar{x}(S) < L + 1$ and [Guarantee MIP] is infeasible, then a violated inequality $x(S) \ge L + 1$ is obtained. We can add this cut to strengthen the restricted DW master problem.

In fact, we can iteratively generate more sets S by appending the constraints $\sum_{v_j \in S} \vartheta_{jl} \le |S|, l = 1, ..., L$ in [Candidate Separation] to cut off the sets S generated so far until $\bar{x}(S) \ge L+1$ for any $S \subseteq V_C$. The resulting separation strategy is illustrated in Algorithm 4.2.

Since any RCI found using the various heuristics designed for CVRP in the literature is valid for the robust VRPTW, these RCI's can be strengthened by solving [Guarantee] for some fixed $L > r(S)$ to obtain the enhanced RCI. The process is summarized in Algorithm 4.3.

Algorithm 4.2 Cutting plane strategy

```
Inputs: x̄_ij, (v_i, v_j) ∈ A, L
Initiation: Continue = true;
while Continue do
    Solve [Candidate Separation].
    if [Candidate Separation] has a solution then
        retrieve S corresponding to the feasible solution.
        if x̄(S) < L + 1 then
            Solve [Guarantee].
            if [Guarantee] is infeasible then
                Add x(S) ≥ L + 1 into RMP.
            end if
            Add Σ_{v_j ∈ S} ϑ_jl ≤ |S|, l = 1, ..., L into [Candidate Separation].
        else
            Continue = false.
        end if
    end if
end while
```

k-path inequalities

The k-path inequality is similar to RCIs except that the separation of k-path inequalities considers the solution feasibility against time windows besides capacity constraints. To distinguish this, let $k(S)$ be the minimum number of vehicles needed to serve the customers in S satisfying the time window and capacity constraints. The k-path inequality is defined as follows:

$$x(S) \geq k(S) \tag{4.53}$$

Separation of these cuts is achieved by generating S and solving the VRPTW problem defined on S with the objective of minimizing the number of vehicles needed. To limit

Algorithm 4.3 Cutting plane strategy

Inputs: S.
Initiation: Continue = true, $L = r(S) + 1$.
while Continue **do**
  Solve [Guarantee] with L.
  if [Guarantee] is infeasible **then**
    Add $x(S) \geq L$ into RMP.
    $L = L + 1$.
  else
    Continue = false.
  end if
end while

the complexity of the separation, most researchers generate inequalities for $k \leq 2$. When $k = 2$, verifying $k(S) \geq 2$ reduces to answering two questions. The first one is to check if

$$\left\lceil \sum_{v_i \in S} \underline{d}_i / Q \right\rceil \geq 2. \tag{4.54}$$

The second is to check the feasibility of a traveling salesman problem with time windows (TSPTW) defined on S. If $x(S) < 2$ and (4.54) holds or the TSPTW is infeasible, then $x(S) \geq 2$ is a valid cut to the problem. Kohl et al. [1999] provide a heuristic that first determines a set S with $x(S) < 2$ and then verifies if $k(S) \geq 2$.

Desaulniers et al. [2008] generalize the k-path inequalities to

$$\sum_{l=1}^{k-1} \frac{1}{l} \sum_{(v_i,v_j) \in \delta(S)_l^-} x_{ij} + \frac{1}{k} \sum_{(v_i,v_j) \in \bigcup \delta(S)_l^-} x_{ij} \geq 1. \tag{4.55}$$

The generalization is based on partitioning $\delta(S)^-$ into several disjoint sets $\delta(S)_l^-, l = 1, .., |S|$, where $(v_i, v_j) \in \delta(S)_l^-$ if a minimum of l vehicles are needed to serve S when arc (v_i, v_j) is used. Let $\mathscr{C}$ be the set of k-path inequalities in the restricted DW master problem, S_ϵ be the set of customers involved for inequality $\epsilon \in \mathscr{C}$, and $\gamma_\epsilon \geq 0$ be the dual variable of inequality ϵ. This type of cuts does not affect the complexity of the subproblem. When solving the subproblem, the reduced cost associated with arc (v_i, v_j) is modified as

$$\bar{c}_{ij} = c_{ij} - \mu_i - \sum_{\epsilon \in \mathscr{C}: v_i \notin S_\epsilon, v_j \in S_\epsilon} \gamma_\tau \tag{4.56}$$

We modify the separation of 2-path inequalities to fit in the robust case by changing the left hand side of (4.54) as follows:

$$\sum_{v_i \in S} \underline{d_i} + \max_{\{\tilde{S} | \tilde{S} \subseteq S, |\tilde{S}| \leq \varGamma\}} \sum_{v_j \in \tilde{S}} h_j \geq Q. \tag{4.57}$$

For a given solution $\bar{x}$ and S, if $\bar{x}(S) < 2$ and inequality (4.57) holds, then a violated inequality $x(S) \geq 2$ is obtained. This is because inequality (4.57) indicates that the total demand in S including the highest $\varGamma$ variations needs at least 2 vehicles to satisfy, which contradicts with $\bar{x}(S) < 2$.

Subset row inequalities

Jepsen et al. [2008] develop inequalities based on the clique and odd hole inequalities for the set-packing problem, called subset row inequalities. To be precise, for a subset of customers $S \subseteq V_C$, any feasible solution must satisfy

$$\sum_{p \in \varOmega} \left\lfloor \frac{1}{k} \sum_{v_i \in S} \alpha_{ip} \right\rfloor \theta_p \leq \left\lfloor \frac{|S|}{k} \right\rfloor, \quad 2 \leq k \leq S \tag{4.58}$$

A tractable separation focuses on a subset of customers with $|S| = 3$, resulting in the inequality

$$\sum_{p \in \Omega_S} \theta_p \leq 1, \tag{4.59}$$

where Ω_S consists of the paths visiting at least two customers in S. Let $\mathscr{S}$ be the subset row inequalities involving three customers in the restricted DW master problem, $\pi_r \leq 0$ be the dual variable for $r \in \mathscr{S}$, and S_r be the three customers relating to inequality $r \in \mathscr{S}$. If route p covers more than one customer in S_r, $-\pi_r$ is added to the reduced cost, i.e. $\bar{c}_p = c_p - \sum_{v_i \in V^p} \mu_i - \sum_{r \in \mathscr{S}: |V^p \bigcap S_r| \geq 2} \pi_r$, where V^p is the set of customers visited by path p. In the label-setting algorithm, this can be captured by introducing a resource for each subset row inequality, and recording the number of customers in S_r visited along the path. However, the efficiency of the label-setting algorithm deteriorates as the number of resources increases. To tackle this, Jepsen et al. [2008] suggest modifying the dominance rule when comparing two partial paths at node v_i. To be specific, given that any feasible extension of p_2 is also feasible to p_1, a sufficient condition for p_1 to dominate p_2 is

$$\bar{c}_1 - \sum_{\substack{r \in \mathscr{S}: \\ |V^1 \bigcap S_r| = 1 \\ |V^2 \bigcap S_r| = 0}} \pi_r \leq \bar{c}_2. \tag{4.60}$$

The left hand side of (4.60) adds all the potential $-\pi_r$'s that may be picked up by p_1 but not by p_2 when extending to one more node. This ensures that for any node that p_2 can reach, p_1 can reach with a reduced cost the same at least.

As the subset row inequalities are based on the characteristic of the set partitioning formulation, and the robust VRPTW can be modeled as a set partitioning problem, the subset row inequalities are valid in the robust case. Also, since the robust ESPPRC can be equivalently solved by solving a series of deterministic ESPPRC, the dominance rule proposed by Jepsen et al. [2008] can be used when solving these deterministic ESPPRC.

4.5. Computational experiments

This section reports on the computational results on a set of modified benchmark instances. First, we describe the instances and the settings used in the branch-and-price-and-cut algorithm. Then we analyze the solution results in terms of the effect of each type of valid inequality and the ease of solution for each type of instances. Finally, we compare the robust solutions with the deterministic solutions in terms of cost, number of vehicles used, average percent of infeasible scenarios (referred as infeasibility rate) and average percent of capacity utilization (referred as utilization rate) under simulated demand variations. We report the complete results for all instances. All algorithms are coded in C++, and MIP and LP models are solved using CPLEX 12.5. Testing is carried out on a workstation with Xeon processor with 3.00GHz and 8GB RAM. If the solution process does not terminate within 3 hours of clock time, we stop the programm.

4.5.1 Test instances

The instances used are based on Solomon benchmark instances for VRPTW with 25 and 50 customers. The benchmark instances are divided into three problem classes with different geographical distributions of the customers. C instances locate customers in clusters. R instances randomly generate customer locations. RC instances involve the properties of both randomness and cluster. Moreover, for each geographical type, there are two sets of instances with different vehicle capacity. Type 1 instances have a vehicle capacity of 200 for each vehicle, while type 2 instances have a capacity of 1000. In type 2 instances, vehicle capacity is relatively high compared to assigned demand. In this case, solutions are not very sensitive to variations in demand. In other words, ample capacity naturally protects against uncertainty. As a result, we use type 1 instances only. For each customer i in a benchmark instance, we uniformly generate h_i in $[0, \lceil 0.5\underline{d}_i \rceil]$ for high level of variation, and in $[0, \lceil 0.3\underline{d}_i \rceil]$ for low level of variation. According to the solutions of the deterministic instances, the average number of customers per vehicle ranges between 6 and 10. Therefore, we set the protection level to 3. This allows 30% to 50% of customers on average to change their demands.

4.5.2 Implementation issues for the Branch-and-price-and-cut algorithm

Before starting the branch-and-price-and-cut algorithm, we first determine a lower bound on the minimum number of vehicles needed, instead of using the lower bound $r(V_C)$. To be specific, the minimum number of vehicles needed to cover all customer demands without considering time window constraints equals the optimal objective value of [Guarantee MIP] with $L = |K|$ and $S = V_C$. We derive a lower bound $\underline{NV}(V_C)$ on $NV(V_C)$ by relaxing constraints (4.49) and using CG to solve the resulting DW master problem. Define $\rho_i, \forall v_i \in V_C$

as the dual variables for constraints (4.49), the resulting subproblem is a robust knapsack problem as follows:

[Robust Knapsack]

$$\min \; -\sum_{v_i \in V_C} \rho_i + 1 \tag{4.61}$$

$$\text{s.t.} \sum_{v_j \in V_C} d_j \sigma_{jl} + \mathit{\Gamma} \eta_l + \sum_{v_j \in V_C} \varrho_{jl} \leq Q \tag{4.62}$$

$$\sigma_{jl} \leq z_l \qquad \forall v_j \in V_C \tag{4.63}$$

$$\eta_l + \varrho_{jl} \geq d_j \sigma_{jl} \qquad \forall v_j \in V_C \tag{4.64}$$

$$\sigma_{jl} \in \{0, 1\} \qquad \forall v_j \in V_C \tag{4.65}$$

The corresponding DW master problem replaces c_p by 1 in the LP relaxation of [P]. When the CG terminates, we use $\lceil \underline{NV}(V_C) \rceil$ as the lower bound for constraint (4.15) in [P]. We solve a TSPTW defined on the set of customers covered by each feasible packing. In this process, a feasible traveling salesman route is a feasible route for the robust VRPTW. The corresponding column is added to the restricted DW master problem.

Since Algorithm 4.1 requires solving a series of ESPPRC problems, we relax the elementary path requirement to accelerate the solution time of the subproblem. However, we enforce the 2-cycle free requirement using the dominance rule proposed by Desrochers et al. [1992]. The bi-directional search is also implemented. To be specific, we define the split point as $b_0/2$ and extend partial paths both forward from the source node and backward from the sink node. Any partial path that consumes more than $b_0/2$ units of time will not be extended. The forward and backward searches are carried out in parallel on two cores using Boost 1.55.0 library. In each iteration of CG, a maximum of 400 columns are added until no more routes with negative reduced cost can be found.

Once the master problem is solved, we generate the three types of valid inequalities sequentially and solve the master problem using CG again. The process repeats until no more valid inequalities are generated. To be specific, at the root node, we separate all three types of inequalities, i.e. 2-path, enhanced RCI and subset row inequalities. Based on our preliminary experiments, the 2-path inequalities are the easiest to separate and improve the lower bound most significantly, and the enhanced RCIs improve the lower bound more than the subset row inequalities. Therefore, the separation process is done as follows. First, 2-path inequalities are separated using the heuristic proposed by Kohl et al. [1999]. In the heuristic, the feasibility check with respect to capacity is based on inequality (4.57). Then the master problem is strengthened with the separated 2-path inequalities and reoptimized using CG. This repeats until no more 2-path inequalities can be found. Second, the enhanced RCIs are separated using Algorithm 4.2. Since solving [Candidate Separation] with $L \geq 4$ is time consuming, we only focus on $L = 2$ and 3. Again, the separation of the enhanced RCI and the reoptimization of the strengthened master problem repeat iteratively until no more enhanced RCIs can be obtained. The last type of inequalities generated are subset row inequalities. As suggested in the literature, we only generate those corresponding to $|S| = 3$ and enumerate all the sets S. Moreover, unlike the 2-path inequalities and the enhanced RCIs, a large number of subset row inequalities create great burden for the label-setting algorithm. Thus, we generate no more than 50 inequalities in each iteration. The separation of subset row inequalities and reoptimization of the strengthened master problem repeat until no more subset row inequalities can be found or a maximum of 150 subset row inequalities is obtained. The separation of the three types of inequalities repeats until none of them can be obtained. For other nodes in the branching tree, we only separate enhanced RCIs for $L = 2$.

The branching scheme involves two types of branching rules. The first branching is on the number of vehicles used, denoted by T. If T is fractional, we add $\sum_p \alpha_p \leq \lfloor T \rfloor$ on one branch and $\sum_p \alpha_p \geq \lceil T \rceil$ on the other. If T is integer, we branch on arc (v_i, v_j) such that $\sum_{k=1}^{|K|} \bar{x}_{ij}^k$ is the closest to 1 for solution $\bar{\boldsymbol{x}}$. On one branch, we enforce the solution to serve v_j right after v_i by the same vehicle, i.e. $\sum_{k=1}^{|K|} x_{ij}^k = 1$. This requires removing all the columns that do not satisfy the constraint, and removing all the arcs from v_i to nodes other than v_j and all the arcs to v_j from nodes other than v_i as well as arc (v_j, v_i). The other branch requires that v_i is followed by any node but v_j, i.e. $\sum_{k=1}^{|K|} x_{ij}^k = 0$. The corresponding modifications are to remove arc (v_i, v_j) and all the columns with v_j being the immediate successor of v_i. The branching scheme employs a depth-first search that explores the branch with $\sum_p \alpha_p \geq \lceil T \rceil$ first or the branch with $\sum_{k=1}^{|K|} x_{ij}^k = 1$ first. Note that the k-path inequalities generated at one branch may not be valid on the other branches when the branching rule forces an arc to be used in a solution on one branch but not on the other. This is not an issue for enhanced RCI.

4.5.3 Numerical results

In this section, we analyze the lower bound on the minimum number of vehicles needed, the effect of each type of valid inequalities on the lower bound and the ease of solution for each type of instances. We compare the robust and deterministic solutions in terms of cost and the number of vehicles used. First, we define the notation used in Tables 4.4 to 4.7 as follows:

- $\underline{L}$ is the lower bound $\lceil \underline{NV}(V_C) \rceil$ on the minimum number of vehicles needed found.
- L_r is the the lower bound on the minimum number of vehicles needed given by $r(V_C) = \lceil \sum_{v_i \in V_C} \underline{d_i}/Q \rceil$.
- LB_0 is the lower bound at the root node without adding any cut.
- I_p is the set of 2-path inequalities added at the root node before any inequalities of other types are added.
- LB_1 is the lower bound at the root node after adding the inequalities in I_p.
- I_e^2 is the set of enhanced RCI inequalities with $L = 2$ added at the root node before any subset row inequalities are added.
- I_e^3 is the set of enhanced RCI inequalities with $L = 3$ added at the root node before any subset row inequalities are added.
- LB_2 is the lower bound at the root node after adding the inequalities in $I_e = I_e^2 \bigcup I_e^3$.
- I_{sr} is the set of subset row inequalities added at the root node in the first round of separation of all three types of inequalities.
- LB_3 is the lower bound at the root node after adding the inequalities in I_{sr}.
- LB_e is the best lower bound at the root node when no more valid cuts can be found.

- $Time$ is the total clock time in seconds used when the branch-and-price-and-cut algorithm terminates.
- $Node$ is the number of nodes explored when the algorithm terminates.
- TV_D is the number of vehicles used in the deterministic solution.
- TV_R is the number of vehicles used in the robust solution.
- UB_R is the best upper bound found when the algorithm terminates. The value is marked with asterisk if the corresponding solution is optimal.
- UB_D is the optimal value of the deterministic solution.
- Inc is the percentage increase of the objective value of the robust solution compared to the corresponding deterministic solution.
- Gap_0 gives the optimality gaps in percentage calculated as $(UB_R/LB_0 - 1) * 100$.
- Gap_1 gives the optimality gaps in percentage calculated as $(UB_R/LB_1 - 1) * 100$.
- Gap_2 gives the optimality gaps in percentage calculated as $(UB_R/LB_2 - 1) * 100$.
- Gap_3 gives the optimality gaps in percentage calculated as $(UB_R/LB_3 - 1) * 100$.
- Gap_e gives the optimality gaps in percentage calculated as $(UB_R/LB_e - 1) * 100$ if the best incumbent is not proved to be optimal.

The values of $\lceil \underline{NV}(V_C) \rceil$ and $r(S)$ in Tables 4.4 to 4.7 show that all the C instances and most R instances have the same value of $\lceil \underline{NV}(V_C) \rceil$ and $r(S)$, whereas most RC instances use an extra vehicle in the robust solution. To facilitate the explanation, Table 4.1 summarizes the average $\lceil \underline{NV}(V_C) \rceil$, $r(S)$, the total nominal demand $\sum_{v_i \in V_C} \underline{d}_i$, and $\sum_{v_i \in V_C} \underline{d}_i / Q$ for each type of instances under low and high variations. Note that the instances of the same type have the same nominal demand data. Since $\sum_{v_i \in V_C} \underline{d}_i / Q$ is closest to $r(V_C)$ for RC instances, it is easier to obtain $\underline{NV}(V_C) > r(S)$ by considering some demand variations. This indicates that it is only worth the efforts to solve the relaxed robust binpacking problem when $\sum_{v_i \in V_C} \underline{d}_i / Q$ is close to $r(V_C)$.

Table 4.1: Relation between total nominal demand and the lower bound on the minimum number of vehicles needed.

	High variation							
	$\lvert V_C \rvert = 25$				$\lvert V_C \rvert = 50$			
avg	$\underline{L}$	L_r	$\sum_{v_i \in V_C} \underline{d}_i$	$\sum_{v_i \in V_C} \underline{d}_i / Q$	$\underline{L}$	L_r	$\sum_{v_i \in V_C} \underline{d}_i$	$\sum_{v_i \in V_C} \underline{d}_i / Q$
C	3	3	460	2.30	5	5	860	4.30
R	2	2	332	1.66	4.17	4	721	3.61
RC	4	3	540	2.70	6	5	970	4.85
	Low variation							
	$\lvert V_C \rvert = 25$				$\lvert V_C \rvert = 50$			
avg	$\underline{L}$	L_r	$\sum_{v_i \in V_C} \underline{d}_i$	$\sum_{v_i \in V_C} \underline{d}_i / Q$	$\underline{L}$	L_r	$\sum_{v_i \in V_C} \underline{d}_i$	$\sum_{v_i \in V_C} \underline{d}_i / Q$
C	3	3	460	2.30	5	5	860	4.30
R	2	2	332	1.66	4	4	721	3.61
RC	3.13	3	540	2.70	6	5	970	4.85

To compare the effect of each type of valid inequalities on the lower bound, we summarize the average gap closed by adding the inequalities in the sets I_p, I_e and I_{sr} in Table 4.2. If a set is empty for all the instances of a type, the corresponding cell is blank in Table 4.2. For both C and RC instances, the 2-path inequalities are most effective for improving the lower bound, closing more than 10% of the gap under all cases. For the R instances with $|V_C| = 25$, the 2-path inequalities close an average of 0.55% gap while the SR inequalities close an average of 0.42%. When $|V_C| = 50$, the subset row inequalities are more effective , with an average of 0.46%, compared to 0.34% for 2-path inequalities. The enhanced RCIs show more impact on the C instances, where the improvements are larger than those by the subset row inequalities in all cases except for $|V_C| = 25$. For the RC instances, the enhanced RCIs are less effective than SR inequalities except for $|V_C| = 25$. There are no enhanced RCIs found for any of the R instances. However, given the fact that the size of I_{sr} is far larger than I_p and I_e as shown in Tables 4.4 to 4.7, and the subset row inequalities complicate the label-setting algorithm, the 2-path and enhanced RCIs are more helpful.

Table 4.2: The average gap closed by the inequalities in sets I_p, I_e and I_{sr}.

		Variation			
		High		Low	
	$\|V_C\|$=	25	50	25	50
C	$2-path$	11.66	10.72	11.40	10.64
	$EhancedRCI$	2.27	0.79		0.41
	$Subsetrow$	0.53	0.39	0.51	0.33
R	$2-path$	0.55	0.33	0.55	0.35
	$EhancedRCI$				
	$Subsetrow$	0.42	0.47	0.42	0.45
RC	$2-path$	12.46	17.89	12.34	17.63
	$EhancedRCI$	0.05	0.19	0.44	0.21
	$Subsetrow$	0.15	0.30	0.35	0.27

To compare the robust solutions and deterministic solutions, we summarize the average Inc and Gap_e in Table 4.3. First, we look at the instances with $|V_C| = 25$. The average cost increases for the R instances are 0 for both low and high variations, indicating that all the R instances are solved to optimality and the optimal objective values of the robust solutions are the same as the deterministic solutions. This is because in these instances, time window constraints dominate. Therefore, more vehicles are used in the optimal solutions of R in the deterministic case, leading to low utilization of vehicle capacity. The C and RC instances are also solved to optimality in both high variation and low variation cases as indicated by the 0 gaps in both cases. However, the optimal values are much higher than the deterministic solutions. The C instances have an average increase of 16.23% for low variation case and 21.41% for high variation case, whereas the values for RC instances are 16.87% and 26.45%, respectively.

Comparing the results for the instances with $|V_C| = 50$ in Tables 4.5 and 4.7, similar patterns are observed. The R instances show a small average cost increase of 0.22% in the low variation case, and a 2.32% increase in the high variation case. Since the R instances that solved to optimality have the same objective values for the robust and deterministic solutions, the increase may be caused by the gaps in the instances not solved to optimality. For the C and RC instances, the average cost increases are far larger than the R instances. The C instances result in 18.08% and 22.99% in the low and high variation cases, whereas the increases are 13.79% and 16.79% for the RC instances. More instances are not solved to optimality for the C and RC instances. However, the gaps are not necessarily much higher than the R instances. Specifically, the average gaps for C and RC instances are 2.80% and 3.08% in the high variation case, compared to 2.72% for the R instances, while the values for the low variation cases are 0.81% and 1.84%, which are smaller than the 3.96% for the R instances. This indicates that suboptimality is not the main reason for the huge increases on cost for C and RC instances. The possible reason is that when clustered customers can not be covered in a single vehicle, the model may need to explore combining customers from different clusters which leads to higher cost.

Table 4.3: Summary of the average Inc and Gap_e.

		Variation			
		High		Low	
	$\lvert V_C\rvert =$	25	50	25	50
Gap_e	C	0	2.80	0	0.81
	R	0	2.72	0	3.96
	RC	0	3.08	0	1.84
Inc	C	21.41	22.99	16.23	18.08
	R	0	0.24	0	0.22
	RC	26.45	16.37	16.87	13.79

4.5.4 Simulation

In this section, we compare the robust and deterministic solutions under simulated scenarios. First, we vary the percentage of customers experiencing demand change, denoted by $\tilde{p}$, and set it to 20%, 40%, 60% or 80% of $|V_C|$. For a given $\tilde{p}$, and for each instance of the data sets, we create 1000 scenarios. In a scenario, a random set of customers $S_{\tilde{p}}$ with $|S_{\tilde{p}}| = \tilde{p} * |V_C|$ are selected from the $|V_C|$ customers. For each customer v_i in $S_{\tilde{p}}$, the real demand is uniformly generated in $[\underline{d}_i, \underline{d}_i + h_i]$.

Tables 4.9 to 4.12 report the statistics on the robust and deterministic solutions. The information reported is the number of vehicles used TV_R and TV_D in the robust and deterministic solutions, the average vehicle utilization in percentage, and infeasibility rates. Average vehicle utilization is calculated in percentage as the ratio of total simulated demand and the number of vehicles used. Infeasibility rate in percentage is the ratio of the number of infeasible scenarios and the total number of scenarios (i.e. 1000). A summary of Tables 4.9 to 4.12 is provided in Table 4.8. The average utilizations of the robust and deterministic solutions are the same when $TV_D = TV_R$. When $TV_R > TV_D$, e.g. the RC

Table 4.4: Results of instances with 25 customers and high demand variation.

	$\underline{L}$	L_r	$\|I_p\|$	$\|I_e^2\|$	$\|I_e^3\|$	$\|I_{sr}\|$	LB_0	LB_e	$Time$	$Node$	TV_D	TV_R	UB_R	UB_D	Inc	Gap_0	Gap_1	Gap_2	Gap_3	Gap_e
C101	3	3	1	0	0	102	205.08	234.08	34.8	11	3	3	238.7*	191.3	24.78	16.39	2.50	2.50	1.98	
C102	3	3	4	0	1	0	210.43	237.7	13.7	1	3	3	237.7*	190.3	24.91	12.96	2.69	0		
C103	3	3	1	0	0	0	197.81	220.4	8.0	1	3	3	220.4*	190.3	15.82	11.42	0			
C104	3	3	3	0	0	112	195.89	218.22	970.5	25	3	3	218.9*	186.9	17.12	11.75	0.74	0.74	0.32	
C105	3	3	9	0	1	0	209.01	237.2	3.6	1	3	3	237.2*	191.3	23.99	13.49	1.90	0		
C106	3	3	4	0	0	114	205.08	234.08	61.0	19	3	3	238.7*	191.3	24.78	16.39	2.44	2.44	1.98	
C107	3	3	6	0	0	102	209.01	232.21	213.8	23	3	3	235.1*	191.3	22.90	12.48	2.00	2.00	1.24	
C108	3	3	2	0	0	51	202.26	226.6	8.6	1	3	3	226.6*	191.3	18.45	12.03	0.48	0.48	0	
C109	3	3	6	2	3	0	203.14	229.5	9.4	1	3	3	229.5*	191.3	19.97	12.98	2.23	0		
														avg	21.41	13.32	1.67	1.02	1.10	
R101	2	2	0	0		0	617.1	617.1	0.1	1	8	8	617.1*	617.1	0	0				
R102	2	2	1	0		0	546.33	547.1	0.3	1	7	7	547.1*	547.1	0	0.14	0			
R103	2	2	0	0		0	454.6	454.6	0.6	1	5	5	454.6*	454.6	0	0				
R104	2	2	0	0		0	416.9	416.9	1.1	1	4	4	416.9*	416.9	0	0				
R105	2	2	0	0		0	530.5	530.5	0.2	1	6	6	530.5*	530.5	0	0				
R106	2	2	6	0		0	457.3	465.4	1.3	1	5	5	465.4*	465.4	0	1.77	0			
R107	2	2	2	0		0	422.93	424.3	1.2	1	4	4	424.3*	424.3	0	0.33	0			
R108	2	2	2	0		51	396.14	397.3	3.4	1	4	4	397.3*	397.3	0	0.29	0.15	0.15	0	
R109	2	2	0	0		0	441.3	441.3	0.6	1	5	5	441.3*	441.3	0	0				
R110	2	2	4	0		66	437.3	442.93	6.9	3	5	5	444.1*	444.1	0	1.55	1.41	1.41	0.26	
R111	2	2	5	0		29	423.79	428.8	1.9	1	4	4	428.8*	428.8	0	1.18	0.18	0.18	0	
R112	2	2	14	0		92	384.2	386.31	108.6	17	4	4	393.0*	393	0	2.29	1.97	1.97	1.76	
														avg	0	0.63	0.53	0.93	0.51	
RC101	4	3	9	0	0	29	424.4	507.9	1.9	1	4	5	507.9*	461.1	10.15	19.67	0.53	0.53	0	
RC102	4	3	5	0	0	0	400.95	434.2	3.4	1	3	4	434.2*	351.8	23.42	8.29	0			
RC103	4	3	7	0	2	8	386.01	445.76	11.5	5	3	4	446.2*	332.8	34.07	15.59	0.25	0.15	0.10	
RC104	4	3	10	0	0	44	363.32	426.03	16.1	3	3	4	426.1*	306.6	38.98	17.28	0.13	0.13	0.03	
RC105	4	3	2	0	0	23	424.52	481.5	3.6	5	4	4	482.9*	411.3	17.41	13.75	0.54	0.54	0.29	
RC106	4	3	2	0	0	7	388.18	420.2	2.4	1	3	4	420.2*	345.5	21.62	8.25	0.19	0.19	0	
RC107	4	3	5	0	1	16	363.61	396.64	26.0	9	3	4	397.6*	298.3	33.29	9.35	0.40	0.40	0.24	
RC108	4	3	10	0	0	0	356.79	390.8	13.2	1	3	4	390.8*	294.5	32.70	9.53	0			
														avg	26.45	12.71	0.25	0.32	0.11	
														all avg	13.94	7.90	0.86	0.77	0.55	

Table 4.5: Results of instances with 50 customers and high demand variation.

	$\underline{L}$	L_r	$\|I_p\|$	$\|I_e^2\|$	$\|I_e^3\|$	$\|I_{sr}\|$	LB_0	LB_e	$Time$	$Node$	TV_D	TV_R	UB_R	UB_D	Inc	Gap_0	Gap_1	Gap_2	Gap_3	Gap_e
C101	5	5	4	0	5	51	395.63	445.4	69.8	9	5	6	455.1*	362.4	25.58	15.03	2.47	2.30	2.18	
C102	5	5	8	0	3	91	392.54	441.65	17043.2	3	5	6	449.7*	361.4	24.43	14.56	4.76	2.23	1.82	
C103	5	5	5	0	0	0	378.41	417.6	103.7	1	5	5	417.6*	361.4	15.55	10.36	0			
C104	5	5	19	0	7	102	377.4	413.2	10856.3	418	5	5	427	358	19.27	13.14	3.81	3.61	3.34	3.34
C105	5	5	26	2	26	102	399.12	448.68	3050.7	3	5	6	453.7*	362.4	25.19	13.67	2.41	1.83	1.15	
C106	5	5	7	0	4	102	397.72	447.75	366.8	3	5	6	454*	362.4	25.28	14.15	1.73	1.61	1.40	
C107	5	5	12	0	4	102	395.63	443.37	2972.6	5	5	6	453.6*	362.4	25.17	14.65	3.50	2.97	2.31	
C108	5	5	9	0	14	102	392.69	435.71	10801.9	384	5	5	444	362.4	22.52	13.07	3.01	2.32	1.92	1.90
C109	5	5	29	6	4	102	392.08	435.43	10803.6	288	5	6	449.2	362.4	23.95	14.57	5.05	3.51	3.17	3.16
														avg	22.99	13.69	2.97	2.55	2.16	2.80
R101	5	4	2	0	0	0	1043.4	1044	0.7	1	12	12	1044*	1044	0	0.06	0			
R102	4	4	0	0	0	0	909		2.7	1	11	11	909*	909	0	0				
R103	4	4	4	0	0	43	765.95	768.63	766.6	203	9	9	772.9*	772.9	0	0.91	0.73	0.73	0.57	
R104	4	4	14	0	0	102	618.92	623.86	20907.4	375	6	6	629.4	625.4	0.64	1.69	1.18	1.18	0.89	0.89
R105	4	4	4	0	0	1	892.12	893.65	27.2	65	9	9	899.3*	899.3	0	0.80	0.63	0.63	0.63	
R106	5	4	6	0	0	0	791.37	793	10.3	1	8	8	793*	793	0	0.21	0			
R107	4	4	3	0	0	102	704.57	707.55	1825.9	115	7	7	711.1*	711.1	0	0.93	0.74	0.74	0.50	
R108	4	4	42	0	0	102	590.77	601.81	10802.1	2697	6	6	625.9	617.7	1.33	5.95	4.92	4.92	4.00	4.00
R109	4	4	6	0	0	142	775.1	781.43	776.5	171	8	8	786.8	786.8	0	1.51	1.34	1.34	0.69	
R110	4	4	3	0	0	0	692.58	694.15	44.1	5	7	7	697*	697	0	0.64	0.41	0.41	0.41	
R111	4	4	6	0	0	102	691.81	699.58	3281.6	527	7	7	707.2*	707.2	0	2.22	2.10	2.10	1.09	
R112	4	4	22	0	0	102	607.92	615.57	10806.1	2963	6	6	635.6	630.2	0.86	4.55	3.75	3.75	3.26	3.25
														avg	0.24	1.62	1.44	1.76	1.34	2.72
RC101	6	5	27	0	0	0	854.07	944	6.6	1	8	8	944*	944	0	10.53	0			
RC102	6	5	22	0	2	102	744.01	882.19	3555.6	635	7	7	887.9*	822.5	7.95	19.34	1.88	1.87	0.65	
RC103	6	5	14	4	24	102	687.18	812.66	10741.5	3	6	7	817.3*	710.9	14.97	18.94	1.37	1.02	0.59	
RC104	6	5	73	3	16	102	609.89	736.8	10912.8	94	5	6	744	545.8	36.31	21.99	1.40	1.18	0.98	0.98
RC105	6	5	17	0	3	48	770.57	885.3	29.8	1	8	8	885.3*	855.3	3.51	14.89	0.03	0.03	0	
RC106	6	5	38	0	4	102	690	834.03	10801.1	16036	6	7	883.9	723.2	22.22	28.10	6.55	6.48	5.98	5.98
RC107	6	5	71	1	12	102	634	761.43	10893.4	94	6	7	774.5	642.7	20.51	22.16	2.55	2.09	1.72	1.72
RC108	6	5	46	7	29	71	599.66	724.28	10802.1	195	6	7	750.8	598.1	25.53	25.20	4.20	3.96	3.71	3.66
														avg	16.37	20.14	2.25	2.38	1.95	3.08
														all avg	11.75	10.48	2.16	2.20	1.79	2.79

Table 4.6: Results of instances with 25 customers and low demand variation.

	$\underline{L}$	L_r	$\|I_p\|$	$\|I_e^2\|$	$\|I_e^3\|$	$\|I_{sr}\|$	LB_0	LB_e	$Time$	$Node$	TV_D	TV_R	UB_R	UB_D	Inc	Gap_0	Gap_1	Gap_2	Gap_3	Gap_e
C101	3	3	1	0	0	148	205.08	234.08	128.6	31	3	3	238.6*	191.3	24.73	16.34	2.61	2.61	1.93	
C102	3	3	1	0	0	0	201.37	220.4	7.5	1	3	3	220.4*	190.3	15.82	9.45	0			
C103	3	3	1	0	0	0	197.17	220.4	8.2	1	3	3	220.4*	190.3	15.82	11.78	0			
C104	3	3	4	0	0	102	195.49	218.22	266.3	19	3	3	218.9*	186.9	17.12	11.98	0.75	0.75	0.32	
C105	3	3	1	0	0	0	205.08	227.6	2.1	1	3	3	227.6*	191.3	18.98	10.98	0			
C106	3	3	0	0	0	0	191.3	191.3	1.5	1	3	3	191.3*	191.3	0	0				
C107	3	3	3	0	0	0	203.11	226.6	3.8	1	3	3	226.6*	191.3	18.45	11.56	0			
C108	3	3	4	0	0	51	203.75	226.6	7.3	1	3	3	226.6*	191.3	18.45	11.21	0.48	0.48	0	
C109	3	3	9	0	0	115	198.69	222.9	280.1	19	3	3	223.2*	191.3	16.68	12.34	0.60	0.60	0.14	
														avg	16.23	10.63	0.55	1.11	0.60	
R101	2	2	0	0		0	617.1	617.1	0.1	1	8	8	617.1*	617.1	0	0				
R102	2	2	1	0		0	546.33	547.1	0.3	1	7	7	547.1*	547.1	0	0.14	0			
R103	2	2	0	0		0	454.6	454.6	0.7	1	5	5	454.6*	454.6	0	0				
R104	2	2	0	0		0	416.9	416.9	0.8	1	4	4	416.9*	416.9	0	0				
R105	2	2	0	0		0	530.5	530.5	0.2	1	6	6	530.5*	530.5	0	0				
R106	2	2	6	0		0	457.3	465.4	1.2	1	5	5	465.4*	465.4	0	1.77	0			
R107	2	2	2	0		0	422.93	424.3	1.1	1	4	4	424.3*	424.3	0	0.33	0			
R108	2	2	2	0		51	396.14	397.3	3.4	1	4	4	397.3*	397.3	0	0.29	0.15	0.15	0	
R109	2	2	0	0		0	441.3	441.3	0.4	1	5	5	441.3*	441.3	0	0				
R110	2	2	4	0		58	437.3	442.93	4.2	3	5	5	444.1*	444.1	0	1.55	1.41	1.41	0.26	
R111	2	2	5	0		24	423.79	428.8	1.8	1	4	4	428.8*	428.8	0	1.18	0.18	0.18	0	
R112	2	2	14	0		81	384.2	386.31	72.2	17	4	4	393*	393	0	2.29	1.97	1.97	1.76	
														avg	0	0.63	0.53	0.93	0.51	
RC101	3	3	8	0	0	0	406.63	461.1	0.6	1	4	4	461.1*	461.1	0	13.40	0			
RC102	4	3	5	0	0	0	400.95	434.2	2.5	1	3	4	434.2*	351.8	23.42	8.29	0			
RC103	3	3	3	0	0	102	336.51	371.96	199.5	7	3	4	387.1*	332.8	16.32	15.04	4.39	4.39	4.07	
RC104	3	3	5	0	0	67	308.88	345.33	2164.5	83	3	4	360.9*	306.6	17.71	16.84	4.91	4.91	4.51	
RC105	3	3	1	0	0	11	416.31	448.95	3.8	3	4	4	449.3*	411.3	9.24	7.93	0.46	0.46	0.08	
RC106	3	3	16	0	1	36	351.8	389.26	28.3	5	3	4	396.9*	345.5	14.88	12.82	2.23	2.17	1.96	
RC107	3	3	20	3	6	53	306.46	349.27	511.2	3	3	4	360*	298.3	20.68	17.47	3.42	3.33	3.07	
RC108	3	3	54	1	4	102	306.4	377.44	2158.0	3	3	4	390.8*	294.5	32.70	27.54	5.23	4.06	3.56	
														avg	16.87	14.92	2.58	3.22	2.88	
														all avg	9.69	7.67	1.25	1.96	1.55	

Table 4.7: Results of instances with 50 customers and low demand variation.

	$\underline{L}$	L_r	$\|I_p\|$	$\|I_e^2\|$	$\|I_e^3\|$	$\|I_{sr}\|$	LB_0	LB_e	$Time$	$Node$	TV_D	TV_R	UB_R	UB_D	Inc	Gap_0	Gap_1	Gap_2	Gap_3	Gap_e
C101	5	5	2	0	2	102	388	442.25	73.1	15	5	6	450.8*	362.4	24.39	16.19	2.27	2.15	1.93	
C102	5	5	2	0	0	0	383.25	417.6	28.3	1	5	5	417.6*	361.4	15.55	8.96	0			
C103	5	5	5	0	0	0	377.33	417.51	92.6	1	5	5	417.6*	361.4	15.55	10.67	0.02			
C104	5	5	19	0	4	102	374.23	411.23	10816.0	143	5	5	413	358	15.36	10.36	0.82	0.74	0.43	0.43
C105	5	5	9	0	3	102	387.93	438.98	548.7	35	5	5	442.1*	362.4	21.99	13.97	1.59	1.07	0.71	
C106	5	5	5	0	1	0	377.98	407.1	15.3	1	5	5	407.1*	362.4	12.33	7.70	0.95	0		
C107	5	5	9	0	1	102	385.96	436.83	651.1	69	5	5	437.8*	362.4	20.81	13.43	1.17	0.58	0.22	
C108	5	5	6	10	29	51	386.54	430.4	3774.4	1	5	5	430.4*	362.4	18.76	11.35	0.91	0.31	0	
C109	5	5	35	2	7	102	379.57	422.34	10922.0	1020	5	5	427.4	362.4	17.94	12.60	1.70	1.68	1.22	1.20
														avg	18.08	11.69	1.05	0.93	0.75	0.81
R101	4	4	2	0	0	0	1043.4	1044	0.6	1	12	12	1044*	1044	0	0.06	0			
R102	4	4	0	0	0	0	909	909	2.0	1	11	11	909*	909	0	0				
R103	4	4	4	0	0	59	765.95	768.63	383.8	115	9	9	772.9*	772.9	0	0.91	0.73	0.73	0.57	
R104	4	4	19	0	0	102	617.29	622.83	8020.1	259	6	6	625.4*	625.4	0	1.31	0.68	0.68	0.41	
R105	4	4	4	0	0	3	892.12	893.65	30.5	41	9	9	899.3*	899.3	0	0.80	0.63	0.63	0.63	
R106	4	4	6	0	0	0	791.37	793	15.8	1	8	8	793*	793	0	0.21	0			
R107	4	4	3	0	0	102	704.44	707.55	1960.8	227	7	7	711.1*	711.1	0	0.95	0.74	0.74	0.50	
R108	4	4	43	0	0	102	590.27	600.93	10818.9	4842	6	6	619.8	617.7	0.34	5.00	3.96	3.96	3.14	3.14
R109	4	4	8	0	0	116	775.1	781.21	629.9	265	8	8	786.8*	786.8	0	1.51	1.36	1.36	0.72	
R110	4	4	3	0	0	0	692.58	694.15	934.2	603	7	7	697*	697	0	0.64	0.41	0.41	0.41	
R111	4	4	6	0	0	102	691.81	699.48	4907.1	537	7	7	707.2*	707.2	0	2.22	2.10	2.10	1.10	
R112	4	4	37	0	0	102	607.23	615.45	10800.4	3129	6	6	644.8	630.2	2.32	6.19	5.29	5.29	4.80	4.77
														avg	0.22	1.65	1.45	1.77	1.37	3.96
RC101	6	5	31	0	0	42	850.02	948.6	18.3	1	8	8	948.6*	944	0.49	11.60	0.24	0.24	0	
RC102	6	5	18	0	3	80	735.42	876.9	164.6	1	7	7	876.9*	822.5	6.61	19.24	0.90	0.90	0	
RC103	6	5	11	4	12	102	676.7	809.44	11282.5	225	6	7	817.3	710.9	14.97	20.78	1.77	1.29	0.97	0.97
RC104	6	5	14	0	5	0	599.44	687.2	256.5	1	5	6	687.2*	545.8	25.91	14.64	0.19	0		
RC105	6	5	18	0	0	64	762.57	884.94	63.8	5	8	8	885.3*	855.3	3.51	16.09	0.18	0.18	0.04	
RC106	6	5	37	0	3	102	685.54	833.62	10836.1	4850	6	7	855.4	723.2	18.28	24.78	3.14	3.09	2.62	2.61
RC107	6	5	35	2	14	102	627.92	756.05	11032.0	178	6	7	770.8	642.7	19.93	22.75	2.65	2.26	1.97	1.95
RC108	6	5	41	5	16	102	596.98	719.11	10272.9	97	6	6	721.3*	598.1	20.60	20.83	0.60	0.46	0.31	
														avg	13.79	18.84	1.21	1.05	0.84	1.84
														all avg	9.50	9.51	1.25	1.28	1.03	2.15

instances and the C instances with $|V_C| = 50$, average vehicle utilization in the robust solutions is lower than the deterministic solutions under all demand variations.

However, even for the same number of vehicles, $TV_D = TV_R$, the infeasibility rate of robust solutions is much lower than the deterministic solutions. Over all 464,000 scenarios, the average infeasibility rate for the robust solutions is 0.57% while that for the deterministic solutions is 41.62%. For each type of instances, as $\tilde{p}$ increases, the average infeasibility rate for deterministic solutions increases significantly. For C instances with 50 customers and $\tilde{p} = 80\%$, all simulated scenarios turn out to be infeasible. While at most 7.52% of scenarios turn out to be infeasible for the robust solutions under the same setting. This demonstrates that robust solutions indeed protect against uncertainties. On average over all 464,000 simulated scenarios, only 2389 are infeasible for the robust case, compared to 170,969 for the deterministic case.

Robustness, however, comes at the expense of an average increase of 11.22% in routing cost over all 116 tested instances. It is worth noting that our model allows controlling the level of robustness of the solutions by adjusting the protection level Γ.

4.6. Conclusion

In this chapter, we presented a robust VRPTW under demand uncertainty. The uncertainty support is based on a set of cardinality constrained sets. We first presented a set of equivalent formulations for the problem. We, then, proposed a branch-and-price-and-cut algorithm to solve the problem, in which the subproblem is a robust ESPPRC that was solved by solving a series of ESPPRC. Several types of inequalities were discussed

Table 4.8: Summary of Tables 4.9 to 4.12.

			TV_R	TV_D	Avg vehicle utilization								Infeasibility rate							
					20%		40%		60%		80%		20%		40%		60%		80%	
					Det	Rob	Det	Rob	Det	Rob	Det	Rob	Det	Rob	Det	Rob	Det	Rob	Det	Rob
Low	25	C	3	3	77.75	77.75	78.83	78.83	79.90	79.90	81.00	81.00	10.53	0	20.61	0.01	34.92	0.12	53.54	0.96
		R	5.08	5.08	34.92	34.92	35.48	35.48	36.04	36.04	36.58	36.58	0	0	0	0	0	0	0	0
		RC	3.25	4	85.73	68.57	87.10	69.66	88.43	70.72	89.79	71.80	2.39	0	12.55	0.01	30.85	0.1	50.1	0.55
	50	C	5	5.11	87.22	85.61	88.44	86.80	89.66	88.00	90.85	89.17	79.31	0	96.97	0.06	99.86	0.83	100	3.77
		R	8	8	48.08	48.08	48.85	48.85	49.62	49.62	50.39	50.39	0	0	0	0	0.02	0.02	0.02	0.02
		RC	6.5	7	77.55	71.11	78.76	72.22	79.94	73.30	81.13	74.40	71.28	0	74.53	0.04	75.43	0.35	76.74	1.33
High	25	C	3	3	78.56	78.56	80.44	80.44	82.29	82.29	84.19	84.19	17.78	0	37.36	0.04	58.44	0.34	78.33	1.62
		R	5.08	5.08	35.29	35.29	36.21	36.21	37.11	37.11	38.05	38.05	0	0	0	0	0	0	0	0
		RC	3.25	4.13	86.64	67.59	88.90	69.36	91.16	71.13	93.43	72.91	11.71	0	39.10	0.06	64.98	0.48	80.49	1.83
	50	C	5	5.67	87.98	78.19	89.99	79.96	91.97	81.72	93.97	83.48	80.87	0.01	97.22	0.14	99.82	1.83	100	7.52
		R	8	8	48.57	48.57	49.81	49.81	51.06	51.06	52.30	52.30	0.12	0	1.33	0	3.94	0.01	6.63	0.28
		RC	6.50	7.13	78.35	70.35	80.37	72.16	82.33	73.92	84.31	75.69	75.61	0.03	81.34	0.13	85.49	1.01	87.71	4.05

Table 4.9: Infeasibility rate for instances with 25 customers and high variation.

			Avg vehicle utilization								Infeasibility rate							
	TV_R	TV_D	20%		40%		60%		80%		20%		40%		60%		80%	
			Det	Rob	Det	Rob	Det	Rob	Det	Rob	Det	Rob	Det	Rob	Det	Rob	Det	Rob
C101	3	3	78.90	78.90	81.19	81.19	83.36	83.36	85.66	85.66	14.3	0	48.6	0	79.7	0	96.3	0
C102	3	3	78.54	78.54	80.37	80.37	82.25	82.25	84.10	84.10	17.2	0	40.6	0	65.9	0	88	0
C103	3	3	78.23	78.23	79.75	79.75	81.31	81.31	82.88	82.88	2.8	0	12.3	0	37.6	1.5	68.1	7.4
C104	3	3	78.23	78.23	79.76	79.76	81.37	81.37	82.81	82.81	81.8	0	98.2	0	99.9	0.1	100	0.2
C105	3	3	78.65	78.65	80.60	80.60	82.44	82.44	84.44	84.44	15	0	35.5	0	56.5	0	81.4	0
C106	3	3	78.69	78.69	80.85	80.85	82.83	82.83	85.00	85.00	3.6	0	20.8	0	43.8	0	71.8	0
C107	3	3	78.90	78.90	81.15	81.15	83.43	83.43	85.58	85.58	14.1	0	42.2	0	73.4	0	91.4	0.3
C108	3	3	78.43	78.43	80.12	80.12	81.76	81.76	83.51	83.51	1.2	0	7.6	0.4	16.6	1.5	28.3	6.7
C109	3	3	78.44	78.44	80.19	80.19	81.90	81.90	83.71	83.71	10	0	30.4	0	52.6	0	79.7	0
Avg	3	3	78.56	78.56	80.44	80.44	82.29	82.29	84.19	84.19	17.78	0	37.36	0.04	58.44	0.34	78.33	1.62
R101	8	8	21.31	21.31	21.87	21.87	22.43	22.43	22.97	22.97	0	0	0	0	0	0	0	0
R102	7	7	24.25	24.25	24.74	24.74	25.26	25.26	25.80	25.80	0	0	0	0	0	0	0	0
R103	5	5	33.92	33.92	34.65	34.65	35.36	35.36	36.08	36.08	0	0	0	0	0	0	0	0
R104	4	4	42.89	42.89	44.32	44.32	45.74	45.74	47.20	47.20	0	0	0	0	0	0	0	0
R105	6	6	28.32	28.32	28.96	28.96	29.61	29.61	30.26	30.26	0	0	0	0	0	0	0	0
R106	5	5	34.18	34.18	35.10	35.10	36.06	36.06	37.04	37.04	0	0	0	0	0	0	0	0
R107	4	4	42.71	42.71	44.04	44.04	45.25	45.25	46.50	46.50	0	0	0	0	0	0	0	0
R108	4	4	42.53	42.53	43.56	43.56	44.56	44.56	45.64	45.64	0	0	0	0	0	0	0	0
R109	5	5	34.11	34.11	35.01	35.01	35.89	35.89	36.81	36.81	0	0	0	0	0	0	0	0
R110	5	5	33.92	33.92	34.64	34.64	35.34	35.34	36.09	36.09	0	0	0	0	0	0	0	0
R111	4	4	42.61	42.61	43.73	43.73	44.77	44.77	45.87	45.87	0	0	0	0	0	0	0	0
R112	4	4	42.70	42.70	43.89	43.89	45.05	45.05	46.33	46.33	0	0	0	0	0	0	0	0
Avg	5.08	5.08	35.29	35.29	36.21	36.21	37.11	37.11	38.05	38.05	0	0	0	0	0	0	0	0
RC101	4	5	69.39	55.52	71.18	56.95	73.13	58.51	75.03	60.03	0	0	0.3	0	1	0	3.2	0
RC102	3	4	92.10	69.08	94.27	70.70	96.38	72.29	98.48	73.86	20.9	0	60.7	0.4	86	3.1	98.2	10.6
RC103	3	4	92.49	69.37	95.16	71.37	97.58	73.18	100.16	75.12	9.1	0	40.2	0	73.6	0	94.8	0
RC104	3	4	92.53	69.40	95.11	71.34	97.72	73.29	100.30	75.23	13.3	0	46.5	0.1	82.9	0.3	97.5	1.8
RC105	4	4	69.58	69.58	71.69	71.69	73.80	73.80	75.82	75.82	1.6	0	8.4	0	29.9	0	59.5	0
RC106	3	4	92.28	69.21	94.45	70.84	96.75	72.56	98.97	74.23	10.5	0	44.1	0	79.3	0.1	96.6	0.9
RC107	3	4	92.40	69.30	94.78	71.08	97.11	72.83	99.57	74.68	15.4	0	50.5	0	80.2	0	96.7	0.5
RC108	3	4	92.35	69.26	94.56	70.92	96.81	72.61	99.10	74.32	22.9	0	62.1	0	86.9	0.3	97.4	0.8
Avg	3.25	4.13	86.64	67.59	88.90	69.36	91.16	71.13	93.43	72.91	11.71	0	39.10	0.06	64.98	0.48	80.49	1.83
Total Avg	3.93	4.17	62.88	57.63	64.47	59.08	66.04	60.52	67.65	61.99	8.75	0	22.38	0.03	36.06	0.24	46.51	1.01

Table 4.10: Infeasibility rate for instances with 50 customers and high variation.

			Avg vehicle utilization								Infeasibility rate							
	TV_R	TV_D	20%		40%		60%		80%		20%		40%		60%		80%	
			Det	Rob	Det	Rob	Det	Rob	Det	Rob	Det	Rob	Det	Rob	Det	Rob	Det	Rob
C101	5	6	88.22	73.51	90.44	75.37	92.69	77.24	94.84	79.03	74.4	0	96.5	0	99.8	0	100	0
C102	5	6	87.79	73.16	89.66	74.71	91.47	76.23	93.31	77.76	84.5	0.1	99.3	0.2	99.9	1.6	100	6
C103	5	5	87.72	87.72	89.41	89.41	91.13	91.13	92.85	92.85	80.1	0	97.6	0.2	99.9	2.2	100	7.4
C104	5	5	87.90	87.90	89.82	89.82	91.69	91.69	93.58	93.58	97.2	0	99.7	0.5	100	7.6	100	35.5
C105	5	6	88.27	73.56	90.59	75.49	92.89	77.41	95.22	79.35	82.7	0	98.1	0	100	0.1	100	0.5
C106	5	6	88.17	73.48	90.39	75.32	92.58	77.15	94.76	78.96	76.7	0	96.9	0	99.8	0	100	0
C107	5	6	88.06	73.38	90.13	75.11	92.22	76.85	94.35	78.63	83	0	98	0.1	100	1.3	100	4.3
C108	5	5	87.66	87.66	89.41	89.41	91.11	91.11	92.81	92.81	64.2	0	89.9	0.3	99.1	3.2	100	11.7
C109	5	6	88.01	73.34	90.02	75.01	91.98	76.65	94.01	78.34	85	0	99	0	99.9	0.5	100	2.3
Avg	5	5.67	87.98	78.19	89.99	79.96	91.97	81.72	93.97	83.48	80.87	0.01	97.22	0.14	99.82	1.83	100	7.52
R101	12	12	30.92	30.92	31.82	31.82	32.71	32.71	33.56	33.56	0	0	0	0	0	0	0	0
R102	11	11	33.55	33.55	34.37	34.37	35.14	35.14	35.92	35.92	0	0	0	0	0	0	0	0
R103	9	9	41.09	41.09	42.09	42.09	43.13	43.13	44.14	44.14	0	0	0	0	0	0	0	0
R104	6	6	62.07	62.07	64.11	64.11	66.09	66.09	67.97	67.97	1.4	0	15.9	0	47.3	0.1	79.6	3.4
R105	9	9	41.06	41.06	42.05	42.05	43.03	43.03	44.04	44.04	0	0	0	0	0	0	0	0
R106	8	8	46.28	46.28	47.48	47.48	48.71	48.71	49.87	49.87	0	0	0	0	0	0	0	0
R107	7	7	52.83	52.83	54.08	54.08	55.37	55.37	56.66	56.66	0	0	0	0	0	0	0	0
R108	6	6	61.66	61.66	63.27	63.27	64.87	64.87	66.48	66.48	0	0	0	0	0	0	0	0
R109	8	8	46.27	46.27	47.48	47.48	48.68	48.68	49.87	49.87	0	0	0	0	0	0	0	0
R110	7	7	52.77	52.77	54.00	54.00	55.25	55.25	56.52	56.52	0	0	0	0	0	0	0	0
R111	7	7	52.74	52.74	53.95	53.95	55.22	55.22	56.45	56.45	0	0	0	0	0	0	0	0
R112	6	6	61.57	61.57	63.07	63.07	64.55	64.55	66.08	66.08	0	0	0	0	0	0	0	0
Avg	8	8	48.57	48.57	49.81	49.81	51.06	51.06	52.30	52.30	0.12	0	1.33	0	3.94	0.01	6.63	0.28
RC101	8	8	62.16	62.16	63.73	63.73	65.29	65.29	66.86	66.86	0	0	0	0	1.7	1.7	5.2	5.2
RC102	7	7	70.88	70.88	72.53	72.53	74.04	74.04	75.64	75.64	88.6	0	99.1	0.3	100	0.9	100	5.4
RC103	6	7	83.21	71.32	85.64	73.41	88.03	75.46	90.40	77.48	96.4	0	100	0	100	0.2	100	0.8
RC104	5	6	99.45	82.88	101.95	84.96	104.33	86.94	106.75	88.96	100	0	100	0.3	100	1.6	100	6.3
RC105	8	8	62.28	62.28	63.97	63.97	65.58	65.58	67.23	67.23	20.1	0.1	51.6	0.1	82.2	0.9	96.5	4.7
RC106	6	7	82.82	70.99	84.84	72.72	86.87	74.46	88.78	76.10	100	0	100	0.1	100	0.6	100	1.5
RC107	6	7	83.00	71.14	85.16	73.00	87.26	74.80	89.36	76.60	100	0.1	100	0.2	100	2.1	100	6.5
RC108	6	7	82.97	71.12	85.11	72.95	87.23	74.77	89.46	76.68	99.8	0	100	0	100	0.1	100	2
Avg	6.50	7.13	78.35	70.35	80.37	72.16	82.33	73.92	84.31	75.69	75.61	0.03	81.34	0.13	85.49	1.01	87.71	4.05
Total Avg	6.66	7.03	69.01	63.77	70.71	65.33	72.38	66.88	74.06	68.43	46.00	0.01	53.16	0.08	56.19	0.85	57.98	3.57

Table 4.11: Infeasibility rate for instances with 25 customers and low variation.

			Avg vehicle utilization								Infeasibility rate							
	TV_R	TV_D	20%		40%		60%		80%		20%		40%		60%		80%	
			Det	Rob	Det	Rob	Det	Rob	Det	Rob	Det	Rob	Det	Rob	Det	Rob	Det	Rob
C101	3	3	77.76	77.76	78.90	78.90	80.01	80.01	81.13	81.13	2.2	0	15.4	0	36.8	0	65.4	0
C102	3	3	77.60	77.60	78.55	78.55	79.44	79.44	80.46	80.46	1.4	0	10.6	0	25.3	0	53	0
C103	3	3	77.63	77.63	78.61	78.61	79.60	79.60	80.60	80.60	0.9	0	7	0	19.4	0	39.6	0
C104	3	3	77.75	77.75	78.78	78.78	79.90	79.90	81.00	81.00	76.9	0	95	0	99.6	0	100	0.3
C105	3	3	78.04	78.04	79.51	79.51	80.81	80.81	82.18	82.18	4.5	0	22.6	0	48.3	0.7	72.3	4.4
C106	3	3	77.46	77.46	78.28	78.28	79.07	79.07	79.90	79.90	0	0	0	0	0	0	0	0
C107	3	3	77.98	77.98	79.28	79.28	80.57	80.57	81.84	81.84	1	0	7.1	0	19.6	0.2	43.2	1.1
C108	3	3	77.94	77.94	79.12	79.12	80.38	80.38	81.56	81.56	3.2	0	16.3	0.1	42	0.2	68.9	2.6
C109	3	3	77.56	77.56	78.48	78.48	79.33	79.33	80.28	80.28	4.7	0	11.5	0	23.3	0	39.5	0.2
avg	3	3	77.75	77.75	78.83	78.83	79.90	79.90	81.00	81.00	10.53	0	20.61	0.01	34.92	0.12	53.54	0.96
R101	8	8	21.09	21.09	21.43	21.43	21.77	21.77	22.11	22.11	0	0	0	0	0	0	0	0
R102	7	7	24.16	24.16	24.65	24.65	25.09	25.09	25.55	25.55	0	0	0	0	0	0	0	0
R103	5	5	33.69	33.69	34.20	34.20	34.70	34.70	35.19	35.19	0	0	0	0	0	0	0	0
R104	4	4	42.15	42.15	42.80	42.80	43.49	43.49	44.16	44.16	0	0	0	0	0	0	0	0
R105	6	6	27.95	27.95	28.24	28.24	28.54	28.54	28.82	28.82	0	0	0	0	0	0	0	0
R106	5	5	33.76	33.76	34.34	34.34	34.93	34.93	35.47	35.47	0	0	0	0	0	0	0	0
R107	4	4	42.12	42.12	42.79	42.79	43.43	43.43	44.06	44.06	0	0	0	0	0	0	0	0
R108	4	4	42.24	42.24	43.00	43.00	43.70	43.70	44.44	44.44	0	0	0	0	0	0	0	0
R109	5	5	33.81	33.81	34.41	34.41	34.98	34.98	35.59	35.59	0	0	0	0	0	0	0	0
R110	5	5	33.62	33.62	34.05	34.05	34.46	34.46	34.87	34.87	0	0	0	0	0	0	0	0
R111	4	4	42.29	42.29	43.05	43.05	43.84	43.84	44.60	44.60	0	0	0	0	0	0	0	0
R112	4	4	42.16	42.16	42.81	42.81	43.51	43.51	44.14	44.14	0	0	0	0	0	0	0	0
avg	5.08	5.08	34.92	34.92	35.48	35.48	36.04	36.04	36.58	36.58	0	0	0	0	0	0	0	0
RC101	4	4	68.51	68.51	69.58	69.58	70.53	70.53	71.56	71.56	0	0	0	0	0	0	0	0
RC102	3	4	91.51	68.64	93.06	69.79	94.62	70.96	96.12	72.09	3	0	11.3	0	30.1	0	50.7	0
RC103	3	4	91.60	68.70	93.25	69.94	94.74	71.05	96.45	72.34	2	0	16.8	0.1	47.6	0.8	79.9	4.4
RC104	3	4	91.16	68.37	92.33	69.25	93.55	70.16	94.68	71.01	0.1	0	4.1	0	12.1	0	32	0
RC105	4	4	68.39	68.39	69.31	69.31	70.17	70.17	71.01	71.01	0	0	0.5	0	2.7	0	7.3	0
RC106	3	4	91.34	68.50	92.71	69.53	94.03	70.52	95.27	71.45	4	0	15.1	0	38.8	0	59.5	0
RC107	3	4	91.58	68.68	93.11	69.83	94.69	71.02	96.26	72.20	4.2	0	26.4	0	58.5	0	83.7	0
RC108	3	4	91.73	68.80	93.45	70.09	95.12	71.34	96.97	72.73	5.8	0	26.2	0	57	0	87.7	0
avg	3.25	4	85.73	68.57	87.10	69.66	88.43	70.72	89.79	71.80	2.39	0	12.55	0.01	30.85	0.1	50.1	0.55
all avg	3.93	4.14	62.23	57.50	63.17	58.36	64.10	59.22	65.04	60.08	3.93	0	9.86	0.01	19.35	0.07	30.44	0.45

Table 4.12: Infeasibility rate for instances with 50 customers and low variation.

	TV_R	TV_D	Avg vehicle utilization								Infeasibility rate							
			20%		40%		60%		80%		20%		40%		60%		80%	
			Det	Rob	Det	Rob	Det	Rob	Det	Rob	Det	Rob	Det	Rob	Det	Rob	Det	Rob
C101	5	6	87.23	72.69	88.47	73.73	89.68	74.73	90.88	75.74	81.1	0	98.2	0	100	0.3	100	0.8
C102	5	5	87.06	87.06	88.12	88.12	89.19	89.19	90.23	90.23	77.1	0	95.2	0	99.8	0	100	0
C103	5	5	87.13	87.13	88.23	88.23	89.30	89.30	90.39	90.39	74.1	0	94.5	0	99.9	0	100	0
C104	5	5	87.19	87.19	88.45	88.45	89.66	89.66	90.84	90.84	94.9	0	100	0	100	0.3	100	1.9
C105	5	5	87.38	87.38	88.67	88.67	90.05	90.05	91.38	91.38	77.3	0	96.8	0.1	99.9	0.4	100	4.3
C106	5	5	87.18	87.18	88.38	88.38	89.53	89.53	90.75	90.75	77	0	95.6	0	99.6	0.1	100	0.8
C107	5	5	87.42	87.42	88.82	88.82	90.20	90.20	91.61	91.61	74.9	0	96	0.2	99.9	4.4	100	17
C108	5	5	87.35	87.35	88.68	88.68	90.06	90.06	91.31	91.31	79.2	0	98.7	0.2	99.9	1.8	100	8
C109	5	5	87.08	87.08	88.14	88.14	89.26	89.26	90.29	90.29	78.2	0	97.7	0	99.7	0.2	100	1.1
avg	5	5.11	87.22	85.61	88.44	86.80	89.66	88.00	90.85	89.17	79.31	0	96.97	0.06	99.86	0.83	100	3.77
R101	12	12	30.54	30.54	31.03	31.03	31.55	31.55	32.05	32.05	0	0	0	0	0	0	0	0
R102	11	11	33.37	33.37	33.98	33.98	34.56	34.56	35.19	35.19	0	0	0	0	0	0	0	0
R103	9	9	40.65	40.65	41.23	41.23	41.81	41.81	42.38	42.38	0	0	0	0	0	0	0	0
R104	6	6	60.93	60.93	61.76	61.76	62.59	62.59	63.41	63.41	0	0	0	0	0.2	0.2	0.2	0.2
R105	9	9	40.61	40.61	41.17	41.17	41.73	41.73	42.28	42.28	0	0	0	0	0	0	0	0
R106	8	8	45.81	45.81	46.56	46.56	47.30	47.30	48.04	48.04	0	0	0	0	0	0	0	0
R107	7	7	52.31	52.31	53.12	53.12	53.91	53.91	54.73	54.73	0	0	0	0	0	0	0	0
R108	6	6	61.01	61.01	61.95	61.95	62.94	62.94	63.88	63.88	0	0	0	0	0	0	0	0
R109	8	8	45.95	45.95	46.83	46.83	47.71	47.71	48.59	48.59	0	0	0	0	0	0	0	0
R110	7	7	52.31	52.31	53.10	53.10	53.96	53.96	54.78	54.78	0	0	0	0	0	0	0	0
R111	7	7	52.38	52.38	53.27	53.27	54.19	54.19	55.04	55.04	0	0	0	0	0	0	0	0
R112	6	6	61.13	61.13	62.16	62.16	63.23	63.23	64.27	64.27	0	0	0	0	0	0	0	0
avg	8	8	48.08	48.08	48.85	48.85	49.62	49.62	50.39	50.39	0	0	0	0	0.02	0.02	0.02	0.02
RC101	8	8	61.55	61.55	62.50	62.50	63.43	63.43	64.34	64.34	0	0	0	0	1.3	0	4.3	0
RC102	7	7	70.43	70.43	71.62	71.62	72.81	72.81	73.99	73.99	75.9	0	95.7	0	99.2	0.1	100	0.1
RC103	6	7	82.13	70.40	83.46	71.54	84.78	72.67	86.03	73.74	98.4	0	100	0	100	0	100	0
RC104	5	6	98.23	81.86	99.52	82.94	100.70	83.92	101.97	84.97	98	0	100	0.1	100	0	100	1.4
RC105	8	8	61.45	61.45	62.26	62.26	63.04	63.04	63.89	63.89	0	0	0.5	0	2.9	0	9.6	0
RC106	6	7	82.08	70.36	83.33	71.43	84.54	72.46	85.80	73.55	98.4	0	100	0	100	0	100	0.2
RC107	6	7	82.24	70.49	83.64	71.69	84.99	72.85	86.39	74.05	99.6	0	100	0.2	100	2.5	100	8.3
RC108	6	6	82.31	82.31	83.76	83.76	85.25	85.25	86.66	86.66	99.9	0	100	0	100	0.2	100	0.6
avg	6.5	7	77.55	71.11	78.76	72.22	79.94	73.30	81.13	74.40	71.28	0	74.53	0.04	75.43	0.35	76.74	1.33
all avg	6.66	6.83	68.36	66.08	69.39	67.07	70.41	68.07	71.43	69.05	44.28	0	50.65	0.03	51.80	0.36	52.21	1.54

and extended to the robust VRPTW based on cardinality constrained uncertainty. We applied the branch-and-price-and-cut algorithm to 112 instances with up to 50 customers. Numerical testing showed that the branch-and-price-and-cut algorithm was successful in solving most instances within reasonable time. Simulation of the robust and deterministic solutions revealed that the former are significantly more robust and protect against uncertainties at the expense of an increase in routing cost.

Chapter 5

Conclusions

5.1. Summary of the thesis

To deal with the challenges resulting from executing decisions made under the deterministic assumption when operational parameters are likely random, the thesis studies three important problems in routing and scheduling under uncertainty. Particularly, we propose the robust crew pairing, the robust resource constrained shortest path, and the robust vehicle routing problem with time windows, based on cardinality constrained uncertainty.

Chapter 2 presents a robust crew pairing problem that considers disruptions between flights. A protection level is specified for each pairing that allows a decision maker to control the level of robustness of the solutions. A major challenge is to solve a robust shortest path problem in the subproblem with uncertainty both in the objective function and in the resource constraint. A sequential algorithm transforms and solves the problem by solving a series of deterministic counterparts. The subproblem algorithm is used within column generation to obtain optimal or close to optimal integer solutions. The robust solutions are tested against the deterministic solutions under simulated disruptions. The results show that the robust solutions lead to more stable crew schedules on average and to significantly less delays.

Chapter 3 extends the robust shortest path problem with single resource constraint to the case with uncertainty in multiple resource constraints. The sequential algorithm, while still valid, fails to solve even small size instances as the number of nominal problems increases exponentially. To address this challenge, we develop a series of graph reduction techniques and propose a new dominance rule for the robust case. Computational tests are carried out on a set of instances generated from benchmark instances. The proposed label-setting algorithm outperforms the sequential approach in terms of solution time and size of problems solved.

Chapter 4 focuses on the robust VRPTW with demand uncertainty. Customer demand is considered to randomly vary in an interval, and a vehicle is able to guarantee service for en-route customers as long as the number of demand changes is limited to a specified level. The model differs from other robust vehicle routing problems in the literature in the uncertainty support assumption and in the solution methodology. The proposed model uses cardinality constrained uncertainty. The robust model is solved using a branch-and-price-and-cut algorithm that uses a novel separation method to determine valid inequalities. The computational results show that the proposed solution method is able to solve instances with up to 50 customers to optimality in a reasonable time. Under simulated demand variations, the robust solutions are significantly less vulnerable to demand uncertainty than the deterministic solutions, and remain feasible most of the time.

5.2. Future research directions

The models proposed in this thesis make assumptions and limitations that can be extended in the future research.

For the robust crew pairing problem, a possible extension is to control the position of the protection in a pairing. This is based on the fact that in real operations, disruptions show both complex correlation and propagation. As a result, protection at the early stage of a pairing is more effective than that at a later stage. To model this, weighed disruptions can be incorporated so that a disruption near the beginning of a pairing is given a larger weight.

For the robust shortest path problem with resource constraints, the performance of the modified label-setting algorithm is affected by factors such as the density of the graph, the number of variations with high values, and the number of resources. It is possible that an exact robust solution may not be attainable for difficult instances. Therefore, heuristics provide an alternative to obtain good solution. Although many successful heuristics have been proposed in the deterministic case, there are no heuristics developed for robust models. Moreover, the extension from the deterministic case to the robust case is not straight forward and a good heuristic requires research efforts to take advantage of the special structure of the uncertainty support.

Future research on the robust VRPTW should focus on improving the separation strategy. The separation strategy proposed in Chapter 4 finds a subset of inequalities that violate the capacity constraints for a specified number of vehicles. Specifically, the subset of customers determined by the separation strategy can be partitioned into a number of subsets where the total demand in each subset violates the capacity. A stronger inequality is defined by a subset of customers where any partition has at least one subset that violates the capacity. Moreover, as the model assumes that the service time and the traveling time are certain, a natural question is whether it is possible to consider uncertainty in these times. Since such extension is not trivial, we need to first note the difference between capacity constraints and the time window constraints. The main difference is that the capacity constraints have an identical upper bound and a lower bound of 0 at each customer while the time window constraints have different upper and lower bounds at each customer. A non-zero lower bound at each customer complicates the problem because the accumulated variation can be absorbed along the path when the vehicle reaches a customer before its service time window. Therefore, considering the first Γ highest variations does not necessarily lead to the worst scenario under the specified protection level. As the time window constraint in the two index flow formulation of VRPTW use auxiliary continuous variables to represent the arrival time at each customer, it is not straight forward to extend this constraint to a robust case with cardinality constrained support. The variations considered along the path have to be explicitly formulated in the constraint. Apart from the need of a new formulation, the solution method, especially the label-setting algorithm for the resulting subproblem, requires novel ideas to address the challenges.

References

Annual and per-minute cost of delays to U.S. airlines. http://www.airlines.org/Pages/Annual-and-Per-Minute-Cost-of-Delays-to-U.S.-Airlines.aspx. Accessed: 2012-05-14.

Air Canada's on-time performance. http://www.aircanada.com/en/about/otp/. Accessed: 2012-05-04.

Why airlines make such meagre profits? http://www.economist.com/blogs/economist-explains/2014/02/economist-explains-5. Accessed: 2014-10-01.

Outlook for air transport to the year 2015. http://mir.gov.kz/images/stories/contents/Cir%20304-AT127_en.pdf. Accessed: 2012-04-13.

WestJet sets sights on Air Canada's title. http://www.theglobeandmail.com/report-on-business/westjet-sets-sights-on-air-canadas-title/article2288493/. Accessed: 2012-04-30.

Gateway cities trucking study. http://www.freightworks.org/Documents/Gateway%20Cities%20Trucking%20Study.pdf. Accessed: 2012-11-10.

Airline operations control. www.agifors.org/document.go?documentId=1207&action=download. Accessed: 2012-04-30.

Adelman, D. A price-directed approach to stochastic inventory/routing. *Operations Research*, 52(4):499–514, 2004.

Agra, A., Christiansen, M., Figueiredo, R., Hvattum, L., Poss, M., and Requejo, C. The robust vehicle routing problem with time windows. *Computers & Operations Research*, 40(3):856–866, 2013.

Ak, A. and Erera, A. A paired-vehicle recourse strategy for the vehicle-routing problem with stochastic demands. *Transportation Science*, 41(2):222–237, 2007.

Ando, N. and Taniguchi, E. Travel time reliability in vehicle routing and scheduling with time windows. *Networks and Spatial Economics*, 6(3-4):293–311, 2006.

Augerat, P., Belenguer, J. M., Benavent, E., Corberán, A., and Naddef, D. Separating capacity constraints in the CVRP using tabu search. *European Journal of Operational Research*, 106(23):546–557, 1998.

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

Baldacci, R., Toth, P., and Vigo, D. Exact algorithms for routing problems under vehicle capacity constraints. *Annals of Operations Research*, 175(1):213–245, 2010.

Baldacci, R., Mingozzi, A., and Roberti, R. Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints (invited review). *European Journal of Operational Research*, 218(1):1–6, 2012.

Ball, M., Barnhart, C., Dresner, M., Hansen, M., Neels, K., Odoni, A., Peterson, E., Sherry, L., Trani, A., and Zou, B. Total delay impact study: A comprehensive assessment of the costs and impacts of flight delay in the united states. http://www.nextor.org/pubs/TDI_Report_Final_11_03_10.pdf, 2010. Accessed: 2010-04-10.

Ball, M. *Network routing*. Handbooks in Operations Research and Management Science. Elsevier, 1st edition, 1995.

Barnhart, C., Hatay, L., and Johnson, E. Deadhead selection for the long-haul crew pairing problem. *Operations Research*, 43(3):491–499, 1995.

Barnhart, C., Johnson, E., Nemhauser, G., Savelsbergh, M., and Vance, P. Branch-and-price: column generation for solving huge integer programs. *Operations Research*, 46(3): 316–329, 1998.

Barnhart, C., Cohn, A., Johnson, E., Klabjan, D., Nemhauser, G., and Vance, P. Airline crew scheduling. In *Handbook of Transportation Sciences*, pages 517–560. Kluwer, 2nd edition, 2003.

Beasley, J. E. and Christofides, N. An algorithm for the resource constrained shortest path problem. *Networks*, 19(4):379–394, 1989.

Ben-Tal, A. and Nemirovski, A. Robust convex optimization. *Mathematics of Operations Research*, 23(4):769–805, 1998.

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

Ben-Tal, A. and Nemirovski, A. Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical Programming*, 88(3):414–424, 2000.

Ben-Tal, A., Goryashko, A., Guslitzer, E., and Nemirovski, A. Adjustable robust solutions of uncertain linear programs. *Mathematical Programming Series A*, 99(2):351–376, 2004.

Ben-Tal, A., Ghaoui, L. E., and Nemirovski, A. *Robust Optimization*. Princeton Series in Applied Mathematics. Princeton University Press, 2009.

Ben-Tal, A., Chung, B. D., Mandala, S., and Yao, T. Robust optimization for emergency logistics planning: Risk mitigation of humanitarian relief supply chains. *Transportation Science B*, 45(8):1177–1189, 2011.

Bent, R. W. and Van Hentenryck, P. Scenario-based planning for partially dynamic vehicle routing with stochastic customers. *Operations Research*, 52(6):977–987, 2004.

Berhan, E., Beshah, B., Kitaw, D., and Abraham, A. Stochastic vehicle routing problem: A literature survey. *Journal of Information & Knowledge Management*, 13(3), 2014.

Bertsimas, D. and Sim, M. Robust discrete optimization and network flows. *Mathematical Programming*, 98(1-3):49–71, 2003.

Bertsimas, D. and Sim, M. The price of robustness. *Operations Research*, 52(1):35–53, 2004.

Bertsimas, D., Iancu, D. A., and Parrilo, P. A. Optimality of affine policies in multistage robust optimization. *Mathematics of Operations Research*, 35(2):363–394, 2010.

Bertsimas, D., Brown, D. B., and Caramanis, C. Theory and applications of robust optimization. *SIAM Review*, 53(3):464–501, 2011.

Bertsimas, D. J. and Simchi-Levi, D. A new generation of vehicle routing research: Robust algorithms, addressing uncertainty. *Operations Research*, 44(2), 1996.

Bertsimas, D., Pachamanova, D., and Sim, M. Robust linear optimization under general norms. *Operations Research Letters*, 32(6):510–516, 2004.

Birge, J. R. and Louveaux, F. *Introduction to Stochastic Programming*. Springer Series in Operations Research and Financial Engineering. Springer, 2011.

Boland, N., Dethridge, J., and Dumitrescu, I. Accelerated label setting algorithms for the elementary resource constrained shortest path problem. *Operations Research Letters*, 34 (1):58–68, 2006.

Bond, D. Commercial aviation on the ropes. *Aviation Week and Space Technology*, 153 (12):46, 2000.

Borndörfer, R., Grötschel, M., and Löbel, A. Scheduling duties by adaptive column generation, 2001.

Bräysy, O. and Gendreau, M. Vehicle routing problem with time windows, part i: Route construction and local search algorithms. *Transportation Science*, 39(1):104–118, 2005a.

Bräysy, O. and Gendreau, M. Vehicle routing problem with time windows, part ii: Metaheuristics. *Transportation Science*, 39(1):119–139, 2005b.

Cardoen, B., Demeulemeester, E., and Beliën, J. Operating room planning and scheduling: A literature review. *European Journal of Operational Research*, 201(3):921–932, 2010.

Chabrier, A. Vehicle routing problem with elementary shortest path based column generation. *Computers & Operations Research*, 33(10):2972–2990, 2006.

Chen, W., Sim, M., Sun, J., and Teo, C.-P. From CVaR to uncertainty set: Implications in joint chance-constrained optimization. *Operations Research*, 58(2):470–485, 2010.

Chen, X., Sim, M., Sun, P., and Zhang, J. A linear-decision based approximation approach to stochastic programming. *Operations Research*, 56(2):344–357, 2008.

Christiansen, C., Lysgaard, J., and Wøhlk, S. A branch-and-price algorithm for the capacitated arc routing problem with stochastic demands. *Operations Research Letter*, 37 (6):392–398, 2009.

Clarke, M. and Ryan, D. *Airline Industry Operations Research*. Springer, 1st edition, 2001.

Cohn, A. and Barnhart, C. Improving crew scheduling by incorporating key maintenance routing decisions. *Operations Research*, 51(3):387–396, 2003.

Cohn, A. and Lapp, M. Airline resource scheduling. In *Wiley Encyclopedia of Operations Research and Management Science*. John Wiley & Sons, Inc., 2010.

Cook, W. and Rich, J. L. A parallel cutting-plane algorithm for the vehicle routing problem with time windows. Technical Report TR99-04, Department of Computational and Applied Mathematics, Rice University, 1999.

Cordeau, J.-F., Desaulniers, G., Desrosiers, J., Solomon, M., and Soumis, F. VRP with time windows. In Toth, P. and Vigo, D., editors, *The Vehicle Routing Problem*, SIAM e-books. Society for Industrial and Applied Mathematics, 2001a.

Cordeau, J.-F., Stojkovic, G., Soumis, F., and Desrosiers, J. Benders decomposition for simultaneous aircraft routing and crew scheduling. *Transportation Science*, 35(4):375–388, 2001b.

Cordeau, J.-F., Laporte, G., Savelsbergh, M. W., and Vigo, D. Vehicle Routing. In *Transportation*, volume 14 of *Handbooks in Operations Research and Management Science*, chapter 6, pages 367–428. Elsevier, 2007.

Daniels, R. and Kouvelis, P. Robust scheduling to hedge against processing time uncertainty in single-stage production. *Management Science*, 41(2):363–376, 1995.

Delage, A. J., Nori, V. S., and Savelsbergh, M. W. P. Dynamic programming approximations for a stochastic inventory routing problem. *Transportation Science*, 38(1):42–70, 2004.

Delage, E. and Ye, Y. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations Research*, 58(3):596–612, 2010.

Desaulniers, G., Desrosiers, J., Dumas, Y., Marc, S., Rioux, B., Solomon, M., and Soumis, F. Crew pairing at air france. *European Journal of Operational Research*, 97(2):245–259, 1997.

Desaulniers, G., Desrosiers, J., Ioachim, I., Solomon, M., and Soumis, F. A unified framework for deterministic time constrained vehicle routing and crew scheduling problems. In Crainic, T. and Laporte, G., editors, *Fleet Management and Logistics*, pages 57–93. Kluwer, 1998.

Desaulniers, G., Desrosiers, J., and Solomon, M. Shortest path problems with resource constraints. In *Column Generation*, pages 33–65. Springer, New York, 2005.

Desaulniers, G., Lessard, F., and Hadjar, A. Tabu search, partial elementarity, and generalized k-path inequalities for the vehicle routing problem with time windows. *Transportation Science*, 42(3):387–404, 2008.

Deshpande, V. and Arikan, M. The impact of airline flight schedules on flight delays. *Manufacturing and Service Operations Management*, 14(3):423–440, 2012.

Desrochers, M. An algorithm for the shortest path problem with resource constraints. Technical Report G-88-27, GEARD, Ecole des H.E.C., Montreal, Canada, 1988.

Desrochers, M. and Laporte, G. Improvements and extensions to the miller-tucker-zemlin subtour elimination constraints. *Operations Research Letters*, 10(1), 1991.

Desrochers, M., Desrosiers, J., and Solomon, M. A new optimization algorithm for the vehicle routing problem with time windows. *Operations Research*, 40(2):342–354, 1992.

Desrosiers, J., Dumas, Y., Desrochers, M., Soumis, F., Sanso, B., and Trudeau, P. A breakthrough in airline crew scheduling. Technical Report G-91-11, Cahiers du GERAD, 1991.

Desrosiers, J., Dumas, Y., Solomon, M. M., and Soumis, F. Time constrained routing and scheduling. In *Network Routing*, volume 8 of *Handbooks in Operations Research and Management Science*, chapter 2, pages 35–139. Elsevier, 1995.

Dror, M. and Trudeau, P. Stochastic vehicle routing with modified savings algorithm. *European Journal of Operational Research*, 23(2), 1986.

Dror, M., Laporte, G., and Trudeau, P. Vehicle routing with stochastic demands: Properties and solution frameworks. *Transportation Science*, 23(3):166–176, 1989.

Dror, M., Laporte, G., and Louveaux, F. Vehicle routing with stochastic demands and restricted failures. *Mathematical Methods of Operations Research*, 37(3):273–283, 1993.

Dumitrescu, I. and Boland, N. Improved preprocessing, labeling and scaling algorithms for the weight-constrained shortest path problem. *Networks*, 42(3):135–153, 2003.

Dunbar, M., Froyland, G., and Wu, C. Robust airline schedule planning: Minimizing propagated delay in an integrated routing and crewing framework. *Transportation Science*, 46(2):204–216, 2012.

Eggenberg, N. *Combining Robustness and Recovery for Airline Schedules.* PhD thesis, École Polytechnique Fédérale de Lausanne, Suisse, Swiss, 2009.

Eggenberg, N., Salani, M., and Bierlaire, M. Robust optimization with recovery: application to shortest paths and airline scheduling. In *7th Swiss Transport Research Conference*, Sep 2007.

Ehrgott, M. and Ryan, D. Constructing robust crew scheduleswith bicriteria optimization. *Journal of Multi-Criteria Decision Analysis*, 11(3):139–150, 2002.

El Ghaoui, L., Oks, M., and Oustry, F. Worst-case value-at-risk and robust portfolio optimization: A conic programming approach. *Operations Research*, 51(4):543–556, 2003.

Erera, A., Morales, J., and Savelsbergh, M. Robust optimization for empty repositioning problems. *Operations Research*, 57(2):468–483, 2009.

Erera, A., Morales, J., and Savelsbergh, M. The vehicle routing problem with stochastic demand and duration constraints. *Transportation Science*, 44(4):474–492, 2010.

Errico, F., Desaulniers, G., Gendreau, M., Rei, W., and Rousseau, L.-M. The vehicle routing problem with hard time windows and stochastic service times. Accessed: 2014-11-30, 2013.

Feillet, D. A tutorial on column generation and branch-and-price for vehicle routing problems. *4OR*, 8(4):407–424, 2010.

Feillet, D., Dejax, P., Gendreau, M., and Gueguen, C. An exact algorithm for the elementary shortest path problem with resource constraints: Application to some vehicle routing problems. *Networks*, 44(3):216–229, 2004.

Ferguson, J., Kara, A. Q., Hoffman, K., and Sherry, L. Estimating domestic U.S. airline cost of delay based on european model. *Transportation Research Part C: Emerging Technologies*, 33(0):311–323, 2013.

Gao, C., Johnson, E., and Smith, B. Integrated airline fleet and crew robust planning. *Transportation Science*, 43(1):2–16, 2009.

Gendreau, M., Laporte, G., and Seguin, R. An exact algorithm for the vehicle routing problem with stochastic demands and customers. *Transportation Science*, 29(2), 1995.

Gendreau, M., Laporte, G., and Seguin, R. Stochastic vehicle routing. *European Journal of Operational Research*, 88(1):3–12, 1996.

Golden, B. L. and Yee, J. R. A framework for probabilistic vehicle routing. *IIE Transactions*, 11:109–112, 1979.

Golden, B. L., Magnanti, T. L., and Nguyen, H. Q. Implementing vehicle routing algorithms. *Networks*, 7:113–148, 1977.

Golden, B. L., Raghavan, S., and Wasil, E. A., editors. *The Vehicle Routing Problem: Latest Advances and New Challenges*. Springer, New York, 2008.

Gounaris, C. E., Repoussis, P. P., Tarantilis, C. D., and Floudas, C. A. A hybrid branch-and-cut approach for the capacitated vehicle routing problem. In *21st European Symposium on Computer Aided Process Engineering*, volume 29 of *Computer Aided Chemical Engineering*, pages 507–511. Elsevier, 2011.

Gounaris, C. E., Wiesemann, W., and Floudas, C. A. The robust capacitated vehicle routing problem under demand uncertainty. *Operations Research*, 61(3):677–693, 2013.

Gouveia, L. A result on projection for the vehicle routing problem. *Journal of Operations Research Society*, 85(3):610–624, 1995.

Groër, C., Golden, B., and Wasil, E. The consistent vehicle routing problem. *Manufacturing Service Operations Management*, 11(4):630–643, 2009.

Gupta, S. and Rosenhead, J. Robustness in sequential investment decisions. *Management Science*, 15(2):18–29, 1972.

Handler, G. and Zang, I. A dual algorithm for the constrained shortest path problem. *Networks*, 10(4):293–309, 1980.

Hassin, R. Approximation schemes for the restricted shortest path problem. *Mathematics of Operations Research*, 17(1):36–42, 1992.

Herroelen, W. and Leus, R. Project scheduling under uncertainty: Survey and research potentials. *European Journal of Operational Research*, 165(2):289–306, 2005.

Irnich, S. and Desaulniers, G. Shortest path problems with resource constraints. In *Column Generation*, pages 33–65. Springer U.S., 2005.

Irnich, S. and Villeneuve, D. The shortest-path problem with resource constraints and k-cycle elimination for k $\geq$ 3. *INFORMS Journal on Computing*, 18(3):391–406, 2006.

Jaillet, P. A priori solution of a traveling salesman problem in which a random subset of the customers are visited. *Operations Research*, 36(6), 1988.

Janak, S., Lin, X., and Floudas, C. A new robust optimization approach for scheduling under uncertainty: II. uncertainty with known probability distribution. *Computers & Chemical Engineering*, 31(3):171–195, 2007.

Jepsen, M., Petersen, B., Spoorendonk, S., and Pisinger, D. Subset-row inequalities applied to the vehicle-routing problem with time windows. *Operations Research*, 56(2):497–511, 2008.

Kallehauge, B., Larsen, J., Madsen, O. B., and Solomon, M. M. Vehicle routing problem with time windows. In *Column Generation*, pages 67–98. Springer U.S., 2005.

Kallehauge, B., Larsen, J., and Madsen, O. B. Lagrangian duality applied to the vehicle routing problem with time windows. *Computers & Operations Research*, 33(5):1464–1487, 2006.

Kara, I., Laporte, G., and Bektas, T. A note on the lifted Miller-Tucker-Zemlin subtour elimination constraints for the capacitated vehicle routing problem. *European Journal of Operational Research*, 158(3):793–795, 2004.

Karasan, O., Pinar, M., and Yaman, H. The robust shortest path problem with interval data. http://www.optimization-online.org/DB_HTML/2001/08/361.html, 2004.

Kenyon, A. and Morton, D. Stochastic vehicle routing with random travel times. *Transportation Science*, 37(1):69–82, 2003.

Klabjan, D. *Topics in airline crew scheduling and large scale optimization*. PhD thesis, Georgia Institute of Technology, 1999.

Klabjan, D., Johnson, E., Nemhauser, G., Gelman, E., and Ramaswamy, S. Airline crew scheduling with regularity. *Transportaion Science*, 35:359–374, 2001a.

Klabjan, D., Johnson, E., Nemhauser, G., Gelman, E., and Ramaswamy, S. Solving large airline crew scheduling problems: Random pairing generation and strong branching. *Computational Optimization and Applications*, 20(1):73–91, 2001b.

Klabjan, D., Johnson, E., Nemhauser, G., Gleman, E., and Ramaswamy, S. Airline crew scheduling with time windows and plane count constraints. *Transportation Science*, 36 (3):337–348, 2002.

Kohl, N., Larsen, A., Larsen, J., Ross, A., and Tiourine, S. Airline disruption management - perspectives, experiences and outlook. *Journal of Air Transport Management*, 13(3): 149–162, 2007.

Kohl, N., Desrosiers, J., Madsen, O. B. G., Solomon, M. M., and Soumis, F. 2-path cuts for the vehicle routing problem with time windows. *Transportation Science*, 33(1):101–116, 1999.

Kouvelis, P. and Yu, G. *Robust Discrete Optimization and Its Applications.* Kluwer Academic Publishers, Netherlands, 1st edition, 1997.

Kouvelis, P., Karawarwala, A., and Guitérrez, G. Algorithms for robust single and multiple period layout planning for manufacturing systems. *European Journal of Operational Research*, 63(2):287–303, 1992.

Kuhn, D., Wiesemann, W., and Georghiou, A. Primal and dual linear decision rules in stochastic and robust optimization. *Mathematical Programming Series A*, 130(1):177–209, 2011.

Kulkarni, R. V. and Bhave, P. R. Integer programming formulation of vehicle routing problems. *European Journal of Operational Research*, 20(1):58–67, 1985.

Lan, S., Clarke, J., and Barnhart, C. Planning for robust airline operations: Optimizing aircraft routings and flight departure times to minimize passenger disruptions. *Transportation Science*, 40(1):15–28, 2006.

Laporte, G. Fifty years of vehicle routing. *Transportation Science*, 43(4):408–416, 2009.

Laporte, G., Nobert, Y., and Desrochers, M. Optimal routing under capacity and distance restrictions. *Operations Research*, 33(5):1050–1073, 1985.

Laporte, G., Louveaux, F. V., and Mercure, H. The vehicle routing problem with stochastic travel times. *Transportation Science*, 26(3):161–170, 1992.

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

Laporte, G., Louveaux, F., and Mercure, H. Models and exact solutions for a class of stochastic location-routing problems. *European Journal of Operational Research*, 39(1): 71–78, 1989.

Lavoie, S., Minoux, M., and Odier, E. A new approach for crew pairing problems by column generation with an application to air transportation. *European Journal of Operational Research*, 35(1):45–58, 1988.

Lettovský, L. *Airline Operations Recovery: An Optimization Approach.* PhD thesis, Georgia Institute of Technology, 1997.

Lettovský, L., Johnson, E., and Nemhauser, G. Airline crew recovery. *Transportation Science*, 34(4):337–348, 2000.

Li, X., Tian, P., and Leung, S. C. Vehicle routing problems with time windows and stochastic travel and service times: Models and algorithm. *International Journal of Production Economics*, 125(1):137–145, 2010.

Li, Z., Ding, R., and Floudas, C. A. A comparative theoretical and computational study on robust counterpart optimization: I. robust linear optimization and robust mixed integer linear optimization. *Industrial Engeering Chemical Research*, 50(18):10567–10603, 2011.

Lin, X., Janak, S., and Floudas, C. A new robust optimization approach for scheduling under uncertainty: I. bounded uncertainty. *Computers & Chemical Engineering*, 28(6-7): 1069–1085, 2004.

Listes, O. and Dekker, R. A scenario aggregation based approach for determining a robust airline fleet composition. *Transportation Science*, 39(3):367–382, 2005.

Lorenz, D. H. and Raz, D. A simple efficient approximation scheme for the restricted shortest path problem. *Operations Research Letters*, 28(5):213–219, 2001.

Lozano, L. and Medaglia, A. L. On an exact method for the constrained shortest path problem. *Computers & Operations Research*, 40(1):378–384, 2013.

Lu, D. and Gzara, F. Models and algorithms for the robust resource constrained shortest path problem. *submitted to Mathematical Programming*, 2014a.

Lu, D. and Gzara, F. The robust crew pairing problem: model and solution methodology. *Journal of Global Optimization, published online*, pages 1–26, 2014b.

Luong, C. OR Journalism Series: Dr Michael W. Carter. https://www.informs.org/About-INFORMS/News-Room/O.R.-Analytics-at-Work-Blog/OR-Journalism-Series-Dr-Michael-W.-Carter, Jan 2013.

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

Mak, K. and Guo, Z. A genetic algorithm for vehicle routing problems with stochastic demand and soft time windows. In *Systems and Information Engineering Design Symposium, 2004. Proceedings of the 2004 IEEE*, pages 183–190, April 2004.

Makri, A. and Klabjan, D. A new pricing scheme for airline crew scheduling. *INFORMS Journal on Computing*, 16:56–67, 2004.

Mehlhorn, K. and Ziegelmann, M. Resource constrained shortest paths. In *Algorithms - ESA 2000*, volume 1879 of *Lecture Notes in Computer Science*, pages 326–337. Springer Berlin Heidelberg, 2000.

Mercier, A. and Soumis, F. An integrated aircraft routing, crew scheduling and flight retiming model. *Computers & Operations Research*, 34(8):2251–2265, 2007.

Mercier, A., Cordeau, J.-F., and Soumis, F. A computational study of benders decomposition for the integrated aircraft routing and crew scheduling problem. *Computers & Operations Research*, 32(6):1451–1476, 2005.

Minoux, M. Column generation techniques in combinatorial optimization: a new application to crew pairing problems. In *XXIVth AGIFORS Symposium*, 1984.

Montemanni, R. and Gambardella, L. An exact algorithm for the robust shortest path problem with interval data. *Computers & Operations Research*, 31(10):1667–1680, 2004.

Muter, I., Birbil, S., Bulbul, K., Sahin, G., Yenigun, H., Tas, D., and Tuzun, D. Solving a robust airline crew pairing problem with column generation. *Computers & Operations Research*, 40(3):815–830, 2013.

Ordóñez, F. *Robust Vehicle Routing*, chapter 8, pages 153–178. 2010.

Pillac, V., Gendreau, M., Guéret, C., and Medaglia, A. L. A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225(1):1–11, 2013.

Ralphs, T. K., Kopman, L., Pulleyblank, W. R., and Trotter, L. E. J. On the capacitated vehicle routing problem. *Mathematical Programming Series B*, 94(2-3):343–359, 2003.

Ribeiro, C. C. and Minoux, M. A heuristic approach to hard constrained shortest path problems. *Discrete Applied Mathematics*, 10(2):125–137, 1985.

Righini, G. and Salani, M. Symmetry helps: Bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints. *Discrete Optimization*, 3(3):255–273, 2006.

Rosenberger, J., Schaefer, A., Goldsman, D., Johnson, E., Kleywegt, A., and Nemhauser, G. A stochastic model of airline operations. *Transportation Science*, 36(4):357–377, 2003.

Rosenberger, J., Johnson, E., and Nemhauser, G. A robust fleet assignment model with hub isolation and short cycles. *Transportation Science*, 38(3):357–368, 2004.

Rosenhead, M., Elton, M., and Gupta, S. Robustness and optimality as criteria for strategic decisions. *Operational Research Quarterly*, 23(4):413–430, 1972.

Russell, R. A. and Urban, T. L. Vehicle routing with soft time windows and erlang travel times. *The Journal of the Operational Research Society*, 59(9):1220–1228, 2008.

Ryan, D. and Foster, B. An integer programming approach to shceduling. In *Computer Scheduling of Public Transport Urban Passenger Vehicle and Crew Scheduling*, pages 269–280. Elsevier, 1 edition, 1981.

Sahinidis, N. Optimization under uncertainty: state-of-the-art and opportunities. *Computers and Chemical Engineering*, 28(6-7):971–983, 2004.

Santos, L., Coutinho-Rodrigues, J., and Current, J. R. An improved solution algorithm for the constrained shortest path problem. *Transportation Research Part B: Methodological*, 41(7):756–771, 2007.

Savage, L. The theory of statistical decision. *Journal of the American Statistical Associatio*, 46(253):55–67, 1951.

Schaefer, A., Johnson, E., Kleywegt, A., and Nemhauser, G. Airline crew scheduling under uncertainty. Technical Report TLI-01-01, Georgia Institute of Technology, 2000.

Secomandi, N. and Margot, F. Reoptimization approaches for the vehicle-routing problem with stochastic demands. *Operations Research*, 57(1):214–230, 2008.

Sengupta, J. Robust decisions in economic models. *Computers & Operations Research*, 18 (2):221–232, 1991.

Shapiro, A. and Philpot, A. A tutorial on stochastic programming. http://www.cse.iitd.ernet.in/~naveen/courses/CSL866/TutorialSP.pdf. Accessed: 2013-10-07.

Shapiro, A., Dentcheva, D., and Ruszczyński, A. *Lectures on Stochastic Programming: Modeling and Theory*. MOS-SIAM series on optimization. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104), 2009.

Shebalov, S. and Klabjan, D. Robust airline crew pairing: Move-up crews. *Transportation Science*, 40(3):300–312, 2006.

Shen, Z., Dessouky, M., and Ordóñez, F. A two-stage vehicle routing model for large-scale bioterrorism emergencies. *Networks*, 54(2):255–269, 2009.

Silver, N. Fly early, arrive on-time. http://www.airlines.org/Pages/Annual-and-Per-Minute-Cost-of-Delays-to-U.S.-Airlines.aspx. Accessed: 2014-06-21.

Smith, B. and Johnson, E. Robust airline fleet assignment: Imposing station purity using station decomposition. *Transportation Science*, 40(4):497–516, 2006.

Smith, S. L., Pavone, M., Bullo, F., and Frazzoli, E. Dynamic vehicle routing with priority classes of stochastic demands. *SIAM Journal of Control and Optimization*, 48(5):3224–3245, 2010.

Sohoni, M., Lee, Y., and Klabjan, D. Robust airline scheduling under block time uncertainty. *Transportation Science*, 45(4):451–464, 2011.

Soyster, A. Convex programming with set-inclusive constraints and application to inexact linear programming. *Operations Research*, 21(5):1154–115, 1973.

Stewart, W. and B., G. Stochastic vehicle routing: A comprehensive approach. *European Journal of Operational Research*, 14(4), 1983.

Sungur, I., Ordóñez, F., and Dessouky, M. A robust optimization approach for the capacitated vehicle routing problem with demand uncertainty. *IIE Transactions*, 40(5): 509–523, 2008.

Sungur, I., Ren, Y., Ordóñez, F., Dessouky, M., and Zhong, H. A model and algorithm for the courier delivery problem with uncertainty. *Transportation Science*, 44(2):193–205, 2010.

Taş, D., Gendreau, M., Dellaert, N., van Woensel, T., and de Kok, A. Vehicle routing with soft time windows and stochastic travel times: A column generation and branch-and-price solution approach. *European Journal of Operational Research*, 236(3):789–799, 2014.

Taş, D., Dellaert, N., van Woensel, T., and de Kok, T. Vehicle routing problem with stochastic travel times including soft time windows and service costs. *Computers & Operations Research*, 40(1):214–224, 2013.

Tam, B., Ehrgott, M., Ryan, D., and Zakeri, G. A comparison of stochastic programming and bi-objective optimisation approaches to robust airline crew scheduling. *OR Spectrum*, 33(1):49–75, 2011.

Tarantilis, C., Stavropoulou, F., and Repoussis, P. A template-based tabu search algorithm for the consistent vehicle routing problem. *Expert Systems with Applications*, 39(4):4233–4239, 2012.

Tekiner, H., Birbil, S., and Bulbul., K. Robust crew pairing for managing extra flights. *Computers & Operations Research*, 36(6):2031–2048, 2009.

Toth, P. and Vigo, D. Models, relaxations and exact approaches for the capacitated vehicle routing problem. *Discrete Applied Mathematics*, 123(13):487–512, 2002.

Verderame, P. M. and Floudas, C. A. Operational planning of large-scale industrial batch plants under demand due date and amount uncertainty. I. Robust optimization framework. *Industrial & Engineering Chemistry Research*, 48(15):7214–7231, 2009.

Verderame, P. M. and Floudas, C. A. Multisite planning under demand and transportation time uncertainty: Robust optimization and conditional value-at-risk frameworks. *Industrial & Engineering Chemistry Research*, 50(9):4959–4982, 2011.

Verderame, P. M., Elia, J. A., Li, J., and Floudas, C. A. Planning and scheduling under uncertainty: a review across multiple sectors. *Industrial & Engineering Chemistry Research*, 49(9):3993–4017, 2010.

Weide, O. *Robust and Integrated Airline Scheduling*. PhD thesis, The University of Auckland, New Zealand, 2009.

Weide, O., Ryan, D., and Ehrgott, M. An iterative approach to robust and integrated aircraft routing and crew scheduling. *Computers & Operations Research*, 37(5):833–844, 2010.

Yang, W., Mathur, K., and Ballou, R. Stochastic vehicle routing problem with restocking. *Transportation Science*, 34(1):99–112, 2000.

Yen, J. and Birge, J. A stochastic programming approach to the airline crew scheduling problem. Technical report, University of Washington, WA., 2000.

Yen, J. Y. Finding the k shortest loopless paths in a network. *Management Science*, 17 (11):712–716, 1971.

Zhu, X. and Wilhelm, W. E. A three-stage approach for the resource-constrained shortest path as a sub-problem in column generation. *Computers & Operations Research*, 39(2): 164–178, 2012.

Zieliński, P. The computational complexity of the relative robust shortest path problem with interval data. *European Journal of Operational Research*, 158(3):570–576, 2004.

Appendices

Appendix A

Label setting algorithm for resource constrained shortest path

A label residing at node i is denoted by vector $C_i = (c_{i1}, ..., c_{i|R|})$ where c_{ir} is the consumption of resource $r = 1, ..., |R|$ for a path from origin to node i. If the path from origin to node i is indexed by p, then the corresponding label is denoted by C_i^p. For two labels at node i, C_i^1 dominates C_i^2, represented by $C_i^1 < C_i^2$, if there exists resource $\overline{r} \in R$ such that $c_{i\overline{r}}^1 < c_{i\overline{r}}^2$ and $c_{ir}^1 \leq c_{ir}^2$ for $r, \overline{r} \in R$. L_i is the set of labels created at node i and P_i is the set of non-dominated labels at node i. $f_{ij}(C_i^p)$ represents the resource extension function that calculates the resource consumption at node j when C_i^p is extended through arc (i, j) .

The algorithm in this appendix is applicable if there exists $r \in R$ such that $t_{ijr} \geq 0$. *Unprocessed* keeps the labels that are created but not processed yet. $W(i)$ is the set of nodes that connects node i with an arc. A pseudo-code of label-setting algorithm is presented by Algorithm 1.1.

Algorithm 1.1 Label setting algorithm

Initialization:
$p = 0$
$C_o^p = (0, ..., 0)$, where $C_o^p \in \mathbb{R}^{|}R|$
mark C_o^p as not dominated
$L_o = \{C_o^p\}$
$L_i = \{(\infty, B_2, ..., B_{|R|})\}, \forall i \in N \backslash \{o\};$
$Unprocessed = \{C_o^p\}$
while $Unprocessed \neq \varnothing$ **do**
    Extract a label $C_i^{\tilde{p}}$ from $Unprocessed$
    if $C_i^{\tilde{p}}$ is not dominated **then**
        for all $j \in W(i)$ **do**
            $p = p + 1$
            $C_j^p = f_{ij}(C_i^{\tilde{p}})$
            if C_j is feasible **then**
                for all $C_j^p \in L_j$ **do**
                    Perform dominance check for C_j^p and $C_j^{\bar{p}}$
                    if C_j^p is dominated **then**
                        mark C_j^p as dominated
                        break
                    else if $C_j^{\bar{p}}$ is dominated **then**
                        mark $C_j^{\bar{p}}$ as dominated
                        if $C_j^{\bar{p}}$ is processed **then**
                            $L_j = L_j \backslash \{C_j^{\bar{p}}\}$
                        end if
                    end if
                end for
                if C_j^p is not dominated **then**
                    $L_j = L_j \cup \{C_j^p\}$
                else
                    delete C_j^p
                end if
            end if
        end for
        mark $C_i^{\tilde{p}}$ as processed.
    else
        delete $C_i^{\tilde{p}}$.
    end if
end while

Appendix B

MIP reformulation

Since the inner maximizations in the objective function and in resource constraints are in the same format, we first deal with the inner maximization in (3.1). Given $\overline{\mathbf{y}} \in Y$, the inner maximization in (3.1) is a bounded knapsack problem as follows:

[KN-OBJ]

$$U(\overline{\mathbf{y}}) = \max \sum_{a=1}^{|A|} h_1^a \overline{y}_1^a u_1^a \tag{B.1}$$

$$\text{s.t.} \sum_{a=1}^{|A|} u_1^a \leq \mathit{\Gamma}_1 \tag{B.2}$$

$$u_1^a \in \{0,1\} \qquad a = 1, ..., |A| \tag{B.3}$$

Since $\mathit{\Gamma}_1$ is an integer, the binary requirement for u_1^a can be relaxed, and u_1^a becomes a continuous variable in $[0,1]$. [KN-OBJ] becomes:

[KN-OBJ]

$$U(\overline{\mathbf{y}}) = \max \sum_{a=1}^{|A|} h_1^a \overline{y}_1^a u_1^a \tag{B.4}$$

$$\text{s.t.} \sum_{a=1}^{|A|} u_1^a \leq \mathit{\Gamma}_1 \tag{B.5}$$

$$0 \leq u_1^a \leq 1 \qquad a = 1, ..., |A| \tag{B.6}$$

Define v_1 as the dual variable for constraint (B.5), $q_1^a, \forall a \in A$ as the dual variable for constraints (B.6). The dual problem of [KN-OBJ] is:

[DKN-OBJ]

$$U(\overline{\mathbf{y}}) = \min \mathit{\Gamma}_1 v_1 + \sum_{a=1}^{|A|} q_1^a \tag{B.7}$$

$$\text{s.t. } v_1 + q_1^a \geq h_1^a \overline{y}_1^a \quad a = 1, ..., |A| \tag{B.8}$$

$$q_1^a, v_1 \geq 0 \qquad a = 1, ..., |A| \tag{B.9}$$

The same primal-dual transformation can be carried out on the inner maximization in constraints (3.5) and result in the following knapsack problems:

[DKN-CONS]

$$Q(\overline{\mathbf{y}}) = \min \mathit{\Gamma}_r v_r + \sum_{a=1}^{|A|} q_r^a \tag{B.10}$$

$$\text{s.t. } v_r + q_r^a \geq h_r^a \overline{y}_r^a \quad a = 1, ..., |A| \tag{B.11}$$

$$q_r^a, v_r \geq 0 \qquad a = 1, ..., |A| \tag{B.12}$$

Replacing the inner maximizations in [robust SPPRC] with $U(\mathbf{y})$ and $Q(\mathbf{y})$, and adding constraints (B.8), (B.9), (B.11), (B.12) result in [MIP-robust SPPRC].

Appendix C

The sequential algorithm

Observing that constraints (B.8) are equivalent to $q_1^a \geq h_1^a \overline{y}_1^a - v_1$ and $q_1^a \geq 0$ for $a = 1, ..., |A|$, we can write q_1^a as:

$$q_1^a = \max\{h_1^a \overline{y}_1^a - v_1, 0\}, \quad a = 1, ..., |A| \tag{C.1}$$

Since $\overline{y}_1^a$ is binary, equality (C.1) can be written as:

$$q_1^a = \max\{h_1^a - v_1, 0\}\overline{y}_1^a, \quad a = 1, ..., |A| \tag{C.2}$$

By replacing q_1^a with equality (C.2), [DKN-OBJ] is reformulated as:

[DKN-OBJ]

$$U(\overline{y}) = \min \ \mathit{\Gamma}_1 v_1 + \sum_{a=1}^{|A|}(\max\{h_1^a - v_1, 0\})\overline{y}_1^a \tag{C.3}$$

$$\text{s.t. } v_1 \geq 0 \tag{C.4}$$

Recall that h_1^a is ordered such that $h_1^1 \geq h_1^2 \geq \cdots \geq h_1^{|A|} \geq h_1^{|A|+1} = 0$. If $v_1 \in [h_1^1, \infty)$, $\max\{h_1^a - v_1, 0\}\overline{y}_1^a = 0, \forall a = |A|+1, \ldots, 1$. If $v_1 \in [h_1^{\overline{a}}, h_1^{\overline{a}-1}]$, $\overline{a} = 2, \ldots, |A|+1$, then $\max\{h_1^a - v_1, 0\}\overline{y}_1^a = (h_1^a - v_1)\overline{y}_1^a, \forall a = 1, \ldots, \overline{a}-1$, while $\max\{h_1^a - v_1, 0\}\overline{y}_1^a = 0, \forall a = \overline{a}, \ldots, |A|+1$. Based on this, the following equation holds:

$$\sum_{a=1}^{|A|}(\max\{h_1^a - v_1, 0\})\overline{y}_1^a = \begin{cases} \sum_{a=1}^{\overline{a}-1}(h_1^a - v_1)\overline{y}_1^a & \text{if } v_1 \in [h_1^{\overline{a}}, h_1^{\overline{a}-1}], \forall \overline{a} = |A|+1, \ldots, 2, \\ 0 & \text{if } v_1 \in [h_1^1, \infty). \end{cases} \tag{C.5}$$

Based on equation (C.5), [DKN-OBJ] can be decomposed into $|A|+1$ subproblems, where each subproblem has v_1 being within a certain interval. Let us use index $e = 1, \ldots, |A|+1$ to denote these subproblems. For $e = 1$,

[SubDKN-OBJ^1]

$$U^1(\mathbf{\overline{y}}) = \min \mathit{\Gamma}_1 v_1 \tag{C.6}$$

$$\text{s.t. } v_1 \in [h_1^1, \infty) \tag{C.7}$$

where $\sum_{a=1}^{|A|}(\max\{h_1^a - v_1, 0\})\overline{y}_1^a$ equals 0 in this case and is removed from the objective. The optimal solution of [SubDKN-OBJ^1] is obtained at $v_1 = h_1^1$. For $e = 2, \ldots, |A|+1$,

[SubDKN-OBJ^e]

$$U^e(\overline{\mathbf{y}}) = \min \ \mathit{\Gamma}_1 v_1 + \sum_{a=1}^{e-1} (h_1^a - v_1)\overline{y}_1^a \tag{C.8}$$

$$\text{s.t. } v_1 \in [h_1^e, h_1^{e-1}] \tag{C.9}$$

Rearranging the objective (C.8), [SubDKN-OBJ^e] becomes:

[SubDKN-OBJ^e]

$$U^e(\overline{\mathbf{y}}) = \min \ (\mathit{\Gamma}_1 - \sum_{a=1}^{e-1} \overline{y}_1^a) v_1 + \sum_{a=1}^{e-1} h_1^a \overline{y}_1^a \tag{C.10}$$

$$\text{s.t. } v_1 \in [h_1^e, h_1^{e-1}] \tag{C.11}$$

Ignoring the constant term $\sum_{a=1}^{e-1} h_1^a \overline{y}_1^a$, the linear objective (C.10) optimizes v_1 over interval $[h_1^e, h_1^{e-1}]$. The optimal solution is obtained at $v_1 = h_1^e$ or $v_1 = h_1^{e-1}$. So,

[SubDKN-OBJ^e]

$$\begin{aligned} U^e(\overline{\mathbf{y}}) &= \min \ \left\{ \mathit{\Gamma}_1 h_1^{e-1} + \sum_{a=1}^{e-1} (h_1^a - h_1^{e-1})\overline{y}_1^a, \mathit{\Gamma}_1 h_1^e + \sum_{a=1}^{e-1} (h_1^a - h_1^e)\overline{y}_1^a \right\} \\ &= \min \ \left\{ \mathit{\Gamma}_1 h_1^{e-1} + \sum_{a=1}^{e-1} (h_1^a - h_1^{e-1})\overline{y}_1^a, \mathit{\Gamma}_1 h_1^e + \sum_{a=1}^{e} (h_1^a - h_1^e)\overline{y}_1^a \right\} \end{aligned}$$

The optimal solution of [DKN-OBJ] is the minimum solution over all [SubDKN-OBJ^e], that is:

[DNK-OBJ]

$$U(\overline{\mathbf{y}}) = \min \left\{ \mathit{\Gamma}_1 h_1^1, \ldots, \mathit{\Gamma}_1 h_1^{e-1} + \sum_{a=1}^{e-1} (h_1^a - h_1^{e-1}) \overline{y}_1^a, \mathit{\Gamma}_1 h_1^e + \sum_{a=1}^{e} (h_1^a - h_1^e) \overline{y}_1^a, \ldots, \right.$$
$$\left. \mathit{\Gamma}_1 h_1^{|A|} + \sum_{a=1}^{|A|} (h_1^a - h_1^{|A|}) \overline{y}_1^a, \sum_{a=1}^{|A|} h_1^a \overline{y}_1^a \right\} \tag{C.12}$$

For ease of exposition, let us redefine $U^e(\overline{y})$ as follows:

$$U^e(\overline{\mathbf{y}}) = \begin{cases} \mathit{\Gamma}_1 h_1^1 & \text{if } e = 1, \\ \mathit{\Gamma}_1 h_1^e + \sum\limits_{a=1}^{e} (h_1^a - h_1^e) \overline{y}_r^a & \text{if } e = 2, \ldots, |A|, \\ \sum\limits_{a=1}^{e} h_1^a \overline{y}_1^a & \text{if } e = |A| + 1. \end{cases} \tag{C.13}$$

Then,

[DNK-OBJ]

$$U(\overline{\mathbf{y}}) = \min \left\{ U^1(\overline{\mathbf{y}}), \ldots, U^{e-1}(\overline{\mathbf{y}}), \ldots, U^{|A|+1}(\overline{\mathbf{y}}) \right\} \tag{C.14}$$

Replacing the inner maximization in (3.1) with $U(\overline{\mathbf{y}})$ defined as in (C.13), [P1] becomes:

[P-OBJ]

$$Z = \min \mathbf{t}_1\mathbf{y}_1 + \min\left\{U^1(\mathbf{y}), \ldots, U^{e-1}(\mathbf{y}), \ldots, U^{|A|+1}(\mathbf{y})\right\} \tag{C.15}$$

$$\text{s.t. } \mathbf{y} \in Y \tag{C.16}$$

$$\mathbf{t}_r\mathbf{y}_r + \max_{\substack{0 \le u_r^a \le 1 \\ \sum_{a=1}^{|A|} u_r^a \le \Gamma_r}} \sum_{a=1}^{|A|} h_r^a y_r^a u_r^a \le B_r \qquad r = 2, ..., |R| \tag{C.17}$$

We show that we can solve [P-OBJ] by solving at most $|A| + 1$ subproblems. First, let us define the following problem:

[I^e]

$$Z^e = \min \mathbf{t}_1\mathbf{y}_1 + U^e(\mathbf{y}) \tag{C.18}$$

$$\text{s.t. } \mathbf{y} \in Y \tag{C.19}$$

$$\mathbf{t}_r\mathbf{y}_r + \max_{\substack{0 \le u_r^a \le 1 \\ \sum_{a=1}^{|A|} u_r^a \le \Gamma_r}} \sum_{a=1}^{|A|} h_r^a y_r^a u_r^a \le B_r \quad r = 2, ..., |R| \tag{C.20}$$

Solving [robust SPPRC-OBJ] is equivalent to solving:

[P-OBJ]

$$Z = \min_{e=1,...,|A|+1} \{Z^e\} \tag{C.21}$$

Theoretically, any objective function can be formulated as a constraint. The previous way of dealing with the inner maximization in the objective function can be applied to the robust resource constraints since their inner maximizations are in the same format. It creates a series of subproblems that have the same feature as [I^e] but with one less robust resource constraint. This constraint is transformed to a nominal resource constraint in these subproblems. Iteratively applying this transformation on the subproblems results in a sequence of SPPRC.

Proof of Theorem 3.1. To show the equivalence, we first prove that the feasible regions of [I^e] and [I^e-Seq] are identical, i.e. any feasible solution of [I^e] is feasible to at least one problem [I^e_w], and any infeasible solution of [I^e] is infeasible to all nominal problems [I^e_w] for $w = 1, \ldots, |A| + 1$.

- Suppose $\mathbf{y}'$ is feasible to [I^e], then $\mathbf{y}' \in Y \cup \mathit{\Omega}$ and there exists $Q^w(\mathbf{y}'_{\overline{r}})$ such that $\mathbf{t}_{\overline{r}}\mathbf{y}'_{\overline{r}} + Q^w(\mathbf{y}'_{\overline{r}}) \leq B_{\overline{r}}$. This means $\mathbf{y}'$ is feasible to problem [I^w].

- Now, suppose $\mathbf{y}'$ is infeasible to [I^e], then

$$\mathbf{y}' \notin Y \cup \mathit{\Omega} \text{ or } \ \mathbf{t}_{\overline{r}}\mathbf{y}'_{\overline{r}} + \min\{Q^1(\mathbf{y}'_{\overline{r}}), \ldots, U^w(\mathbf{y}'_{\overline{r}}), \ldots, U^{n+1}(\mathbf{y}'_{\overline{r}}) > B_{\overline{r}}.$$

  If $\mathbf{y}' \notin Y \cup \mathit{\Omega}$ then $\mathbf{y}'$ is infeasible to all [I^e_w]. If $\mathbf{t}_{\overline{r}}\mathbf{y}'_{\overline{r}} + \min\{Q^1(\mathbf{y}'_{\overline{r}}), \ldots, U^w(\mathbf{y}'_{\overline{r}}), \ldots, U^{n+1}(\mathbf{y}'_{\overline{r}}) > B_{\overline{r}}$, then

$$\mathbf{t}_{\overline{r}}\mathbf{y}'_{\overline{r}} + Q^w(\mathbf{y}'_{\overline{r}}) \geq \mathbf{t}_{\overline{r}}\mathbf{y}'_{\overline{r}} + \min\{Q^1(\mathbf{y}'_{\overline{r}}), \ldots, U^w(\mathbf{y}'_{\overline{r}}), \ldots, U^{n+1}(\mathbf{y}'_{\overline{r}})\} > B_{\overline{r}}, \forall w = 1, \ldots, |A|+1.$$

  This means $\mathbf{y}'$ is infeasible to all [I^e_w].

Therefore, the feasible regions of [I^e] and [I^e-Seq] are identical. Obviously, the objective function of [I^e] is same to the objective function of each [I^e_w], so the objective value of of any feasible solution of [I^e] is the same as the objective value of [I^e-Seq]. The equivalence between [I^e] and [I^e-Seq] is proved. □