# Routing and Scheduling Optimisation under Uncertainty for Engineering Applications

Mateusz Damian Polnik

Submitted in fulfilment of the requirements
for the degree of Doctor of Philosophy

Intelligent Computational Engineering Laboratory
Mechanical and Aerospace Engineering
University of Strathclyde, Glasgow

To my parents: Anna and Józef.

# Acknowledgements

During my stay in Glasgow, I was honoured to meet many outstanding people whom I am grateful for their help and support, guidance, words of wisdom, research collaborations and time we spent together. I take the opportunity to mention a few of them.

I want to thank Dr. Annalisa Riccardi, who was my supervisor. I grateful to you for the time you had for me, your endless patience, positive attitude, help in resolving problems I struggled with, our research projects and pastoral support. I thank you that trusted I am the person who can deliver this project.

I thank Prof. Massimiliano Vasile for his pastoral support and his tireless efforts to improve the satisfaction of postgraduate students in the department.

I thank Dr. Ashwin Arulselvan and Dr. Kerem Akartunalı from Strathclyde Business School for their active contributions to our research collaboration, many exciting conversations and going outs. Kerem, I am indebted to you for showing me the best practices of diplomacy and effective communication during the peer review process. I am also very grateful for the priceless lesson about the importance of hope and perseverance in achieving challenging goals.

I am grateful to Dr. Daniel KL Oi and Dr. Luca Mazzarella from the Department of Physics for introducing me to the topic of satellite quantum key distribution and their active contributions to the research collaboration. Due to their support and expertise, I had an excellent opportunity to contribute to a paper published in a Physics journal which was a rewarding experience.

I thank Kimberley Hose and Garry Martin from Glasgow City Council for the opportunity to work on scheduling home care workers in Glasgow. I am grateful for

their trust and the responsibility they were not afraid to share with me. The significant societal impact of the project made me very proud of my work.

I thank Dr. Edmondo Minisci, whose office was located near my desk, for his guidance and support. I was lucky to observe how well you deal with students. I was also amazed on numerous occasions by your passion for sharing knowledge and helping students to solve technical problems in their projects. If I ever have an opportunity to mentor other people, you will be the example I will follow.

I thank Dr. Marilena Di Carlo for her contributions to the research paper and many friendly conversations. I was inspired by your commitment to hard work every day and your resilience.

I am grateful to Cristian Greco and Dr. Christopher Lowe for their help in selecting a proper orbit for a satellite in one of my research projects. Cristian, it was a pleasure to work with such a talented person.

I am grateful to Dr. Viola Renato, Dr. Lorenzo Ricciardi, Dr. Alessandro Falchi, Dr. Andrew Wilson, Dr. Stephen Connolly, who were postgraduate students when I started my studentship. I could observe how you led your projects to a successful finish, which was an endless source of inspiration for me.

I thank Dr. Jimmy-John Hoste, who used to sit beside my desk in the final year of his PhD programme. I enjoyed our frequent going outs, conversations with coffee and your companionship.

I thank Dr. David Garcia Cava for his positive attitude, friendly conversations and dinners in a Polish restaurant at Glasgow West End.

I am grateful to Dr. Meisam Jalalvand for his friendship and sharing with me the culture of his home country. It was a truly unique experience.

I thank Dr. Stuart Grey and Dr. Olga Ganilova for their positive attitude and frequent words of encouragement.

I am grateful to Audrey Berquand for her active involvement in various initiatives in the department, the research group and outside the university. I was a big fan of your organisational and leadership skills.

I thank Deep Bandivadekar for many sincere conversations, stories about India, which I found fascinating, and frequent jogging activities in the beautiful Glasgow Green park.

I thank Robin Kamenický for the all the time we spent together, his friendship and support.

Finally, I would like to thank Robert Garner, Gaetano Pascarella, Francesco Marchetti, Francesco Torre, Gianluca Filippi, Callum Wilson, Carlos Ortega, Federico Toso, Liliane Kissita, and many others who were fellow students during my stay at Strathclyde.

Last but not least, I am indebted to Małgorzata Piesowicz for supporting me during difficult times and celebrating with me the moments of joy.

# Abstract

The thesis aims to develop a viable computational approach suitable for solving large vehicle routing and scheduling optimisation problems affected by uncertainty. The modelling framework is built upon recent advances in Stochastic Optimisation, Robust Optimisation and Distributionally Robust Optimization. The utility of the methodology is presented on two classes of discrete optimisation problems: scheduling satellite communication, which is a variant of Machine Scheduling, and the Vehicle Routing Problem with Time Windows and Synchronised Visits. For each problem class, a practical engineering application is formulated using data coming from the real world. The significant size of the problem instances reinforced the need to apply a different computational approach for each problem class. Satellite communication is scheduled using a Mixed-Integer Programming solver. In contrast, the vehicle routing problem with synchronised visits is solved using a hybrid method that combines Iterated Local Search, Constraint Programming and the Guided Local Search metaheuristic.

The featured application of scheduling satellite communication is the Satellite Quantum Key Distribution for a system that consists of one spacecraft placed in the Lower Earth Orbit and a network of optical ground stations located in the United Kingdom. The satellite generates cryptographic keys and transmits them to individual ground stations. Each ground station should receive the number of keys in proportion to the importance of the ground station in the network. As clouds containing water attenuate the signal, reliable scheduling needs to account for cloud cover predictions, which are naturally affected by uncertainty. A new uncertainty sets tailored for modelling uncertainty in predictions of atmospheric phenomena is the main contribution to the methodology. The uncertainty set models the evolution of uncertain parameters using

a Multivariate Vector Auto-Regressive Time Series, which preserves correlations over time and space. The problem formulation employing the new uncertainty set compares favourably to a suite of alternative models adapted from the literature considering both the computational time and the cost-effectiveness of the schedule evaluated in the cloud cover conditions observed in the real world. The other contribution of the thesis in the satellite scheduling domain is the formulation of the Satellite Quantum Key Distribution problem. The proof of computational complexity and thorough performance analysis of an example Satellite Quantum Key Distribution system accompany the formulation.

The Home Care Scheduling and Routing Problem, which instances are solved for the largest provider of such services in Scotland, is the application of the Vehicle Routing Problem with Time Windows and Synchronised Visits. The problem instances contain over 500 visits. Around 20% of them require two carers simultaneously. Such problem instances are well beyond the scalability limitations of the exact method and considerably larger than instances of similar problems considered in the literature. The optimisation approach proposed in the thesis found effective solutions in attractive computational time (i.e., less than 30 minutes) and the solutions reduced the total travel time threefold compared to alternative schedules computed by human planners. The Essential Riskiness Index Optimisation was incorporated into the Constraint Programming model to address uncertainty in visits' duration. Besides solving large problem instances from the real world, the solution method reproduced the majority of the best results reported in the literature and strictly improved the solutions for several instances of a well-known benchmark for the Vehicle Routing Problem with Time Windows and Synchronised Visits.

# Contents

# List of Acronyms

**AFSCN** Airforce Satellite Control Network. 101, 102

**ALNS** Adaptive Large Neighbourhood Search. 177, 178, 186, 187

**ARC** Adjustable Robust Counterpart. 48, 49

**ARO** Adjustable Robust Optimisation. 182

**BIC** Bayesian Information Criterion. xx, 86, 219, 220

**BOX** Box Uncertainty Set Model. 84, 118, 119, 122–126, 128

**BTS** Box Time Series Uncertainty Set Model. xxi, 87, 89, 118, 119, 121–126, 128–131

**CC** Chance Constraint. 22, 24, 27, 29, 36, 37, 43, 50–52, 95, 96, 103, 119, 180, 181

**CCE** Classical Certainty Equivalent. 26–28

**CP** Constraint Programming. xx, 12, 13, 60, 65, 67, 70, 82, 83, 131, 133, 134, 142–146, 149, 152, 155, 157, 163, 164, 166, 169, 172, 173, 177, 185, 188, 191–198, 201, 207–209, 211, 213, 214

**CV@R** Conditional Value-at-Risk. 24, 28, 30, 36, 56, 95, 118, 119, 122–126, 128, 129, 183, 206

**CVRP** Capacitated Vehicle Routing Problem. 132, 180, 184

**DO** Discrete Optimization. 2, 6, 12, 17, 46, 64, 65, 67, 209, 213

**DRCC** Distributionally Robust Chance Constraint. 23, 50, 52

**DRH** Deterministic Multistage Model with Folding Horizon. 80, 118, 123, 124, 126, 128

**DRO** Distributionally Robust Optimisation. 5, 6, 9, 12, 15, 22–24, 36, 37, 41, 50, 51, 53, 54, 59, 60, 69–71, 76, 84, 92, 97, 104, 119, 121, 128, 206, 210, 213

**DSF** Deterministic Single Forecast Model. 78, 85, 107, 118, 123, 124, 128

**EIR** Economic Index of Riskiness. 28

**EO** Earth Observations. 101

**EOS** Earth Observation Satellite. 68, 99, 100, 103

**ERI** Essential Riskiness Index. xix, 8, 11, 13, 28, 29, 96, 118, 119, 122–126, 128, 129, 131, 146, 147, 149–152, 172, 173, 198, 200, 202, 206, 208

**FPTAS** Fully Polynomial Time Approximation Scheme. 65, 79, 205

**GA** Genetic Algorithm. 177, 179

**GLS** Guided Local Search. 156, 169–171, 177, 179, 185, 192

**GRASP** Greedy Randomized Adaptive Search Procedure. 184

**HCSRP** Home Health Care Scheduling and Routing Problem. xxi, xxii, 11, 14, 179, 190, 207–209

**HHC** Home Health Care. 13, 14, 133, 175–179, 198, 202, 211

**HSBB** High-Speed Broadband. 106

**ILS** Iterated Local Search. 184

**LEO** Lower Earth Orbit. 12, 68, 97, 99, 110, 204

**LNS** Large Neighbourhood Search. 155, 163, 168, 179

**LP** Linear Programming. 2, 4, 8, 16, 17, 19, 20, 37–39, 41–43, 48, 49, 54, 57, 60, 61, 65, 100, 141, 176, 205, 210

**LS** Local Search. xi, 12, 60, 65–67, 100, 102, 131, 155–157, 159–164, 166–171, 177, 191, 193, 207

**MAP** Manpower Allocation Problem. 133, 178

**MAPE** Mean Absolute Percentage Error. 122–124, 126, 127

**MaxPDT** Maximum Percentage Data Transferred. 68, 70, 71, 102, 211

**MCC** Mean Cloud Cover. 121–124, 126, 127

**MILP** Mixed-Integer Linear Programming. 17, 96, 97, 131

**MIP** Mixed-Integer Programming. 12, 13, 21, 59–62, 65, 70, 79, 87, 95–97, 100, 106, 131, 133, 134, 136, 137, 141, 143, 144, 173, 176, 177, 179, 184, 205–207, 209–211, 213

**MP** Mathematical Programming. 4, 5, 12, 15–17

**MSTD** Mean Standard Deviation Model. 91–93, 118, 119, 122–126, 128, 129, 131

**OCE** Optimised Certainty Equivalent. 25, 27–29

**OR** Operations Research. 2, 4, 99, 100, 175

**PERT** Program Evaluation Review Technique. xix, 147–149

**PSD** Positive Semidefinite. 45, 46, 55, 56

**PSDP** Positive Semidefinite Programming. 16

**PSO** Particle Swarm Optimisation. 177, 179

**QP** Quadratic Programming. 16

**RAAN** Right Ascension of the Ascending Node. xxi, 108–110, 112, 205

**RC** Robust Counterpart. 21, 23, 38, 41–49, 51, 52, 57, 60, 62, 64, 216, 219

**RO** Robust Optimisation. 5, 6, 8, 9, 12, 15, 20, 21, 23, 24, 36–39, 41, 42, 47–51, 58–60, 67, 69–71, 76, 84, 90, 92, 93, 97, 119, 180, 181, 185, 206, 210, 213, 215

**RVI** Requirements Violation Index. 28, 29, 146

**SA** Simulated Annealing. 177, 179, 186

**SAA** Sample Average Approximation. 30–36, 103

**SatQKD** Satellite Quantum Key Distribution. xix, 7, 9, 11, 12, 68–71, 74, 77–79, 82, 87, 91, 94–97, 99, 105, 106, 112, 115, 117, 130, 204–207, 209–212, 214

**SDP** Semidefinite Programming. 57

**SFI** Service Fulfilment Index. 28, 29, 146

**SkillVRP** Skill Vehicle Routing Problem. 176

**SL** Service Level. xix, xxi, 109, 110, 116, 117

**SO** Stochastic Optimisation. 5, 6, 9, 12, 15, 17, 22, 23, 30, 34, 36, 37, 69–71, 84, 94, 97, 103, 118, 128, 180, 181, 185, 206, 210, 213

**SOC** Second-Order Conic. 16, 45, 46, 56–58, 60, 61, 90

**SOCP** Second-Order Conic Programming. 16, 43

**SP** Stochastic Programming. 17, 18, 30, 37, 53

**SSO** Sun-Synchronous Orbit. 105, 108

**TRS** Technician Routing and Scheduling. 133

**TS** Tabu Search. 66, 155, 177, 179, 182

**TSP** Traveling Salesman Problem. 95, 132, 157, 160, 184

**UK** United Kingdom. xx, 13, 104–106, 108, 110, 205, 206, 219, 220

**uMCE** u-Mean Certainty Equivalent. 27, 28

**UTC** Coordinated Universal Time. 108, 112

**V@R** Value-at-Risk. 24, 27, 52

**VAR** Vector Autoregression. xx, 6, 8, 9, 12, 85–87, 97, 106, 206, 219, 220

**VNS** Variable Neighbourhood Search. 155, 177, 179

**VRP** Vehicle Routing Problem. 11, 66, 71, 132, 156, 173, 175, 176, 180, 181, 184, 185

**VRPPD** Vehicle Routing Problem with Pickup and Delivery. 132, 133

**VRPRC** Vehicle Routing Problem with Resource Constraints. 176

**VRPSC** Vehicle Routing Problem with Side Constraints. 13, 133, 134, 145, 174, 175, 188, 198, 202

**VRPTW** Vehicle Routing Problem with Time Windows. 29, 132, 133, 136, 175, 180–182, 207

**VRPTWSync** Vehicle Routing Problem with Time Windows and Synchronized Visits. xxi, 6, 8, 11, 14, 133, 174, 175, 179, 185, 188, 191, 202, 207–209, 211, 213, 214

**VRPTWTD** Vehicle Routing Problem with Time Windows and Temporal Dependencies. 133

**WSRP** Workforce Scheduling and Routing Problems. 176

# List of Figures

# List of Tables

List of Tables

# Preface

## Suggested Background

The author of the thesis assumes the reader is familiar with the basics of Linear Programming (LP), Discrete Optimization (DO), Statistics and Computational Complexity Theory. The recommended background for the reader is equivalent to having an undergraduate degree either in Computer Science or Operations Research (OR) and substantial exposure to another discipline.

## Notation and Conventions

The vector notation is used to formulate mathematical equations. The symbols representing vectors are denoted using small bold letters. Capital letters written in regular font represent sets. Matrices are denoted using bold capital letters.

Individual elements of vectors and matrices are accessed using the subscript notation in which the original symbol of the vector or the matrix is written using a regular font, i.e., $v_i$ is the $i$-th element of one-dimensional vector $\mathbf{v}$. Overall, vectors are preferred to matrices even for multidimensional data. Matrices are used only when matrix operations can simplify the notation. The syntax $\|\mathbf{x}\|_2$ denotes Euclidean norm and $\mathbf{x}^\top$ represents transpose of the vector $\mathbf{x}$.

For some set $S$, $i \in S$ is either enumeration over the elements of the set or the test for containment. The meaning of the operation is apparent from the context. The operator $|\cdot|$ yields the cardinality of the set.

The symbol $\mathbb{R}$ denotes real numbers. The syntax $\mathbb{R}_{\geq 0}$ is the subset of non-negative real numbers. Without loss of generality, other valid range expressions may appear in the subscript. The symbols $\mathbb{Z}$ and $\mathbb{B}$ denote integer and binary numbers, respectively.

The probability of a given expression is written as $\mathbb{P}(\cdot)$ and $\mathbb{E}(\cdot)$ evaluates the expected value of the expression. It will be apparent from the context whether the probability distribution is known precisely or it belongs to some set of probability measures (a.k.a. ambiguity set). In the former case, a tilde adorns random variables.

# Chapter 1

# Introduction

The ability to make intelligent decisions leading an individual or a group to achieve their long-term objectives is a desirable and appreciated trait. When faced with important decisions, our ancestors engaged in various practices to reduce uncertainty and secure a more favourable fortune [1]. Ancient Greeks, for instance, paid tribute to deities in temples and visited oracles. Throughout centuries, these practices evolved to ultimately become obsolete by the development of Mathematics and related disciplines [1], notably Statistics, Computer Science and Operations Research (OR).

The foundations of the current state-of-the-art rational decision-making framework, Mathematical Programming (MP), were set in the early fifties of the last century by the pioneering work of George Dantzig on Linear Programming (LP) described in the book [2]. In MP, a decision problem is encoded as a system of mathematical equations commonly referred to as constraints. The assignment of variables for which the system of equations is feasible represents a valid solution. If there are many alternative solutions, the decision-maker may introduce an objective function to distinguish between them. The objective typically returns the cost or profit of a given solution, which is minimised or maximised, respectively. In LP, the objective function is linear, constraints are defined using linear inequalities, and decision variables are continuous [3]. The theoretical complexity of the simplex method developed for solving LPs in the worst case is exponential in the number of variables. However, on average, the theoretical complexity is linear [4]. The result is corroborated empirically [5].

A formal specification of the objective function and constraints is required in MP. Obtaining such a formulation in the real world is further complicated by the need to deal with uncertainty as some parameters of the problem are not known precisely. Typical reasons for this behaviour are [6]: Heisenberg Uncertainty Principle, manufacturing errors, rounding errors, limited knowledge of initial conditions, reliance on approximate models or predictions about the uncertain future. Overall, uncertainty can be reduced, for instance, by increasing the cost of manufacturing, but it cannot be eradicated. Hence, there is no guarantee that the optimal solution to the deterministic problem will retain its feasibility properties and value of the objective function after the parameters of the problem deviate from their nominal values [7]. Therefore, it may be more pragmatic to find solutions that are more likely to remain feasible and whose objective will not be significantly affected by changes in the problem formulation after the real values of the uncertain parameters are revealed [8]. Solutions that share such properties are called robust. The field of modelling uncertainty in MP has been experiencing considerable interest from the research community for the last two decades. It has started by seminal works of Nemirovski and Ben-Tal [7, 9–11], Bertsimas and Sim [8, 12] and others. Overall, these collective efforts have led to the development of Robust Optimisation (RO) [6] and then Distributionally Robust Optimisation (DRO) [13,14]. Alongside Stochastic Optimisation (SO) [15] which has been developed earlier, together they provide a broad range of modelling techniques for representing uncertain parameters in mathematical models. The suitability of a given framework depends on the wealth of available information about the problem's parameters and their behaviour.

## 1.1 Research Objectives

The thesis aims to develop a methodology for solving large vehicle routing problems with pairwise synchronisation constraints and scheduling problems for satellites whose orbits do not change. Basic formulations of the optimisation models and the solution methods are proposed first for deterministic problems without considering uncertainty. Subsequently, the initial formulations are extended by incorporating various models of uncertainty. For both problem classes, the utility of the presented methodology is validated on a suite of problem instances derived from real-world data sets.

The primary source of uncertainty for the vehicle routing problems considered in the thesis is the duration of visits. The adopted model of uncertainty employs the riskiness index optimisation over scenarios [16]. The approach is transferable for uncertainty in travel time. The problem instances are solved using a multistage optimisation algorithm built on top of a Constraint Programming solver [17].

Scheduling satellite operations is affected by uncertainty in cloud cover forecasts. The assumption that the satellite's orbit is fixed allows for formulations with time discretisation. Several alternative models of uncertainty developed using SO, RO and DRO paradigms are considered. They include a new uncertainty set proposed in the thesis that contains a Vector Autoregression (VAR) time series model [18]. The resultant formulations are solved directly using an off-the-shelf mixed-integer programming solver.

The selection of routing and scheduling was considered carefully. Both problem classes belong to Discrete Optimization (DO) problems and have many engineering applications. For instance, scheduling algorithms are implemented in every operating system, which supports multitasking to optimise throughput, responsiveness, the time of completion, etc. On the other hand, a notable example of a routing problem is a variant of the vehicle routing problem with pick-up and delivery, which is solved online by food delivery companies, courier companies and taxi services. Surprisingly, the problem formulations and solution methods successful in scheduling applications are not effective in routing problems. Therefore, solving large instances of vehicle routing problems requires a different approach. Ultimately, considering the two problem classes provides a comprehensive treatment of uncertainty modelling techniques for DO problems.

Moreover, routing and scheduling applications are ubiquitous in the real world. This fact is used as an opportunity to concentrate the research on practical applications modelled using real data. In particular, combined routing and scheduling of home care workers with uncertain visit duration is considered, which is an example of the Vehicle Routing Problem with Time Windows and Synchronized Visits (VRPTWSync).

The featured scheduling problem is the emerging application of scheduling Satellite Quantum Key Distribution (SatQKD) with uncertain cloud cover.

The determination to use real data exerted a significant influence on the shape of this work.

Firstly, the size of the problems considered and their complexity are analogous to the real-world counterparts. Scheduling problems are not artificially made more complicated than they are supposed to be in the real world. Simultaneously, the size of vehicle routing problems encountered in practical applications is not artificially reduced. Similarly, all operational constraints the domain experts reported are present in the problem formulation.

Secondly, in modelling the optimisation problem, real data is used whenever available or realistic approximation schemes are employed otherwise. For instance, to model travel time in the vehicle routing problem, a routing engine that relies on a real network of roads accessible for pedestrians is used. In a similar vein, the occurrence and duration of communication windows between a satellite and a ground station were computed using orbital dynamics models.

Overall, consistent reliance on real data has made the process of obtaining results more challenging. For instance, employing a routing engine to computing the shortest path in a real city a pedestrian may follow requires more work than assuming Euclidean distance. Nonetheless, it made the problem formulations more authentic. It also does not require introducing additional assumptions as real-world third party organisations generated data. Most importantly, many contributions of this thesis can be motivated by the requirements of practical applications, and its impact is demonstrated on problem instances derived from real-world data sets.

## 1.2 Contributions

Notable contributions discussed in the thesis are briefly introduced below.

### 1.2.1 Contributions to Methodology

A new variant of the polyhedral uncertainty set built as a combination of the box uncertainty set and the Vector Autoregression (VAR) time series model contributes to the RO methodology. The uncertainty set is designed for representing parameters whose spatial and temporal correlations can be modelled using a VAR process. The target application of the uncertainty set is modelling weather conditions that are observed in different locations and change over time. To the best knowledge of the author, as of this writing, no other uncertainty set described in the literature is capable of modelling spatial and temporal correlations using a polyhedron, which does not increase the complexity of the solution procedure when applied to LP.

A contribution to optimisation methods is a new multistage algorithm for solving the deterministic VRPTWSync. The algorithm can solve the largest problem instances considered in the academic literature as of this writing. Specifically, the problem instances considered in the thesis were five times larger than the size of problems in the popular benchmark [19]. Furthermore, the multistage solution algorithm reported five new best solutions for this benchmark. Improving the best-known results is critical because researchers use benchmark problems to validate their ideas.

Encouraged by positive results of the multistage algorithm in solving the deterministic vehicle routing problem, the author extended the solution method to use the Essential Riskiness Index (ERI) to minimise the probability and the magnitude of delay in arrival to the customer. The riskiness index is evaluated over a finite set of scenarios representing the duration of individual home care visits recorded in the past. Most importantly, the ERI criterion can be evaluated in polynomial time to the number of scenarios and integrated with constraint programming solvers by formulating a custom constraint. Such a constraint was applied in a new variant of the multistage algorithm to solve the suite of real-world home healthcare problems. To the best knowledge of the author, as of this writing, it is the first time the riskiness index optimisation is used to solve large real-world problems. Previously it has been used to solve five times smaller synthetic problem instances without synchronised visits and side constraints imperative for the application in practice.

### 1.2.2 Contributions to Science

The SatQKD was formulated for the first time as an optimisation problem and communicated to the Physics community in a journal publication. As of this writing, SatQKD has been successfully demonstrated in experiments. However, it reasonable to expect that an evaluation of the design of a commercial SatQKD communication system requires estimates of how many quantum keys could be reliably delivered to the given network of ground stations. The solution to the deterministic variant of the optimisation problem proposed in the thesis is the schedule of satellite data transfers that deliver the largest number of keys to ground stations proportional to the importance of nodes in the network. Therefore, the decision-makers could use the formulation to determine whether a given communications system could deliver the number of keys sufficient for operations. To the best knowledge of the author, as of this writing, the optimisation problem proposed is the first rigorous attempt to answer this question. Besides comparing alternative designs of a system, the formulation can be used to find orbital parameters of the satellite that maximise the number of keys delivered to the system.

### 1.2.3 Contributions to Applications

The deterministic formulation was transformed into five alternative models for SatQKD with uncertain cloud cover developed using RO, DRO and SO frameworks. The deterministic formulation was a prerequisite for this work. The derived models can be reduced back to the deterministic counterpart by eliminating uncertainty using configuration parameters of the given model. Overall, the problem formulations based on official weather forecasts are effectively solvable to optimality. Furthermore, one could find circumstances such as particular weather conditions in which using a given model is beneficial. For instance, some models perform better if cloud cover forecasts tend to contain significant errors. Others give an advantage in adverse weather conditions etc. Overall, among the models compared, the DRO and the VAR models consistently outperformed the deterministic model, which motivates the research efforts presented in the thesis and demonstrates the utility of the models developed for practical applications.

The daily routing and scheduling problems were solved for the largest home care organisation which operates in Scotland. The company employs 2700 carers and delivers 95000 home visits every week to 3000 elderly and sick who live in the Glasgow area. The scheduling software developed by the author was used for a pilot study in the organisation. The schedules found by the multistage solution algorithm either reduced travel time by approximately three times or used up to 10% less workforce depending on the configuration of the objective. The results convinced the company board that the cost-effectiveness of schedules developed by humans could be considerably improved, and the scheduling problems they are dealing with are solvable in practice using existing technology. Consequently, the company is planning to introduce scheduling optimisation in the future. Based on the current budget of the organisation, which exceeded 40 mln GBP per annum in 2018, and the scale of operations, even a small relative improvement should make a noteworthy impact in the real world.

## 1.3 Research Output

The majority of the research work presented in the thesis has already been published as journal articles.

The deterministic variant of the SatQKD problem has been described in [20] and accompanied by a computational study using historical weather information. The article presents an extended treatment of the topic, which includes contributions of other authors, such as the optical model used to measure the number of cryptographic keys delivered over the atmospheric channel and the algorithm to compute the elevation angle between the satellite and the ground station. The manuscript that describes the models with the treatment of uncertainty and its corresponding data set is currently under review. The implementation of all models and software to generate problem instances and analyse results is available open source [21]. The same applies to problem instances used in the computational study of the deterministic formulation [22].

The deterministic variant of the multistage algorithm has been published in [23]. The thesis expands the content discussed in the article to cover additional formulations of the objective function to reduce the workforce and the ERI optimisation over scenarios. The source code repository is publicly available [24]. Furthermore, the solutions to the VRPTWSync [25] benchmark and Home Health Care Scheduling and Routing Problem (HCSRP) problem instances [26] are available online for testing and benchmarking purposes.

The author also investigated the opportunity for improvement in the algorithm to find the shortest paths with resource constraints by using efficient indexing structures, which is a subproblem for solving a VRP using column generation. The work has been published as a conference paper [27]. Nonetheless, the topic has not explored further and was not included in the thesis because column generation does not allow for solving instances with a large number of visits.

## 1.4 Structure of the Thesis

The chapter *Literature Review* opens with an accelerated introduction to MP and SO. It is an opportunity to introduce the core terminology, briefly cover fundamental building blocks of formulations for optimisation problems with uncertainty and discuss recent advancements in decision criteria such as riskiness indices. The state of the art frameworks for optimisation with uncertainty developed more recently: RO and DRO are presented next. Having discussed the methodology, the following solution methods are described: Mixed-Integer Programming (MIP), Constraint Programming (CP), Local Search (LS) and important metaheuristics for DO. Overall, the literature review is dedicated to presenting the methodology. The contributions that are pivotal for the applications considered in the thesis are mentioned in the opening sections of subsequent chapters.

The chapter *Methodology: Satellite Quantum Key Distribution* begins with the description of optimisation problems related to satellite operations scheduling. The study concentrates on the satellites placed in the Lower Earth Orbit (LEO) regime whose orbits remain the same throughout the scheduling horizon. Among available optimisation problems in this domain, the focus is on the SatQKD problem, which is an emerging application and has not been considered as an optimisation problem. The MIP formulation for SatQKD as a deterministic optimisation problem is proposed first. In the real world, cloud cover conditions that restrict the number of keys delivered to a ground station are not known precisely at the time the schedule is computed. Moreover, such predictions tend to lose accuracy over time significantly. To address this issue, several models that describe the uncertainty of cloud cover predictions using the principles of RO, DRO and SO are presented next. An uncertainty set built using VAR time series model that describes spatial and temporal correlations between random variables, which weather conditions certainty exhibit is the contribution to the methodology.

The chapter *Application: Satellite Quantum Key Distribution* presents computational results obtained by solving optimisation models introduced in the preceding chapter. The literature review on applications of optimisation methods employed in the scheduling of satellite operations is provided first. Notably, the modelling approach with time discretisation is recognised in the literature as an effective technique for scheduling satellites whose orbits are not changed during optimisation. Then, the design of a quantum key distribution system in the UK suggested by domain experts [20] and adopted in the thesis is outlined. The analysis of results obtained using a deterministic model is focused on the assessment of the performance of the communications system given historical weather observations. Then, a featured example demonstrates the importance of optimisation in setting the initial orbital parameters of the satellite to maximise the number of keys delivered to the network of ground stations. Following the discussion of results obtained using historical weather conditions, scheduling satellite operations using official weather forecasts and the models with the treatment of uncertainty are presented next. For the performance evaluation, instead of weather predictions, actual cloud cover conditions observed at the time the schedules would be executed are employed. The presentation of results is concluded by studying which formulations lead to more effective solutions in given circumstances, i.e., adverse weather conditions. Following a comprehensive treatment of cloud cover uncertainty in scheduling applications, the subsequent chapters focus on vehicle routing problems.

The main result in the chapter *Methodology: Vehicle Routing Problem with Side Constraints* is the multistage algorithm developed to solve large real-world instances of the vehicle routing problems with additional constraints required for a given application domain, i.e. HHC. Before explaining the details of the algorithm, a MIP and a CP formulation of the deterministic Vehicle Routing Problem with Side Constraints (VRPSC) problem are outlined. The details of the algorithm for computing the ERI in optimisation over a finite set of scenarios are described next. It is a novel decision criterion that penalises both the probability of late arrival to the visit's location and the magnitude of the violation. The design of the multistage optimisation algorithm followed by the algorithmic details of each stage close the chapter. The effectiveness of the proposed solution method is demonstrated in the next chapter.

The chapter *Application: Vehicle Routing Problem with Side Constraints* presents the computational results of the multistage optimisation algorithm. The discussion of results is preceded by the literature reviews of solution methods applied to routing and scheduling in HHC applications and vehicle routing under uncertainty. An important observation from the survey is the modest size of problem instances solved in the literature if a pairwise synchronisation between the start time of visits was considered. The analysis of the computational results is commenced by solving a popular benchmark of the VRPTWSync and comparing the solutions obtained using the multistage algorithm with the results published in the literature. Overall, the benchmark set of problems contains relatively small instances (i.e., less than 80 visits). Nonetheless, the purpose of the comparison was to demonstrate that the multistage algorithm is competitive with other solution methods presented in the literature. The computational results for a suite of real-world problem instances of the Home Health Care Scheduling and Routing Problem (HCSRP) obtained from the largest provider of these services in Scotland are discussed next. The problem instances are at least five times larger than the previous benchmark. Despite the larger size, the problem instances remain effectively solvable by the multistage algorithm. Each problem instance was solved three times: minimising the total travel time, reducing the number of carers needed to run the schedule and minimising the riskiness index of starting the visit with a delay. The solutions found by the algorithm considerably outperform the schedules computed by human planners.

The final chapter, *Conclusions*, summarises the work, recapitulates the most important contributions of the thesis and discusses promising research directions in the future.

# Chapter 2

# Literature Review

This chapter discusses techniques for modelling uncertainty in MP. The treatment of the subject is focused on presenting the most relevant concepts in SO, RO and DRO. The reader who is already familiar with the topic will be introduced to the terminology used in the thesis and presented recent developments in the literature. On the other hand, the reader who aims to develop expertise in the field will be offered an accelerated and unified introduction to the methodology accompanied by references for further self-study.

## 2.1 Mathematical Programming

An essential prerequisite for solving an optimisation problem to provable optimality or obtaining a certificate of infeasibility is a formal specification of the problem as a mathematical program. It consists of three parts: declaration of the decision variables, definition of constraints and the objective function. The last two components are optional. Mathematical programs that have no constraints are called unconstrained optimisation problems. On the other hand, formulations that lack the objective function are satisfaction problems.

Understandably, the properties of the functions used in the formulation of the objective and constraints (i.e., boundedness, convexity, ability to compute the value of a function or to find sub-gradients and super-gradients for the given input in polynomial time) and the type of decision variables (i.e., continuous, integer, binary) have an impact on the difficulty of solving the optimisation problem.

The following example instance of a Second-Order Conic (SOC) mathematical program illustrates different families of constraints.

$$\min \mathbf{c}^\top \mathbf{x} \tag{2.1}$$

$$\text{s.t.:} \quad \mathbf{A}_L \mathbf{x} + \mathbf{b}_L \leq 0 \tag{2.2}$$

$$\|\mathbf{A}_{Q_i}\mathbf{x}\|_2^2 + \mathbf{b}_{Q_i}^\top \mathbf{x} + c_{Q_i} \leq 0 \qquad \forall i \in 1..m_Q \tag{2.3}$$

$$\left\|\mathbf{A}_{S_j}\mathbf{x} + \mathbf{b}_{S_j}\right\|_2 + \mathbf{c}_{S_j}^\top \mathbf{x} + d_{S_j} \leq 0 \qquad \forall j \in 1..m_S \tag{2.4}$$

$$\mathbf{x} \in \mathbb{R}_{\geq 0}^n \tag{2.5}$$

Formally, $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{A}_L \in \mathbb{R}^{n \times m_L}$, $\mathbf{b} \in \mathbb{R}^{m_L}$, $\mathbf{A}_{Q_i} \in \mathbb{R}^{n \times n_{Q_i}}$, $\mathbf{b}_{Q_i} \in \mathbb{R}^{n_{Q_i}}$, $c_{Q_i} \in \mathbb{R}$, $\mathbf{A}_{S_j} \in \mathbb{R}^{n \times n_{S_j}}$, $\mathbf{b}_{S_j} \in \mathbb{R}^{n_{S_j}}$, $\mathbf{c}_{S_j} \in \mathbb{R}^n$, $d_{S_j} \in \mathbb{R}$.

The objective of the hypothetical optimisation problem is to minimise the value of the dot product for the vector of continuous decision variables $\mathbf{x}$ whose elements are restricted to the non-negative orthant. A feasible solution must satisfy the families of Linear Constraints (2.2), Quadratic Constraints (2.3) and Second-Order Conic Constraints (2.4). The syntax $\|\cdot\|_2$ denotes the Euclidean norm. For an accelerated introduction to norms, the interested reader is referred to A.1.

The class of Second-Order Conic Programming (SOCP) comprises Quadratic Programming (QP) and Linear Programming (LP) as special cases. Such programs can be effectively solved using interior-point methods. Positive Semidefinite Programming (PSDP), which is the nearest generalisation of Second-Order Conic Programming (SOCP), requires a different solver technology and is less scalable. For instance, as of this writing, the author is not aware of Positive Semidefinite Programming (PSDP) applications in routing and scheduling problems besides small instances (i.e., 12 customers and one vehicle) [16]. Since the primary aim of the thesis is solving large problem instances arising in the real world, PSDP is outside the scope.

The family of supported decision variable types in MP includes integers and binary variables to model DO problems. A mathematical program whose objective function and constraints are linear expressions but some of the decision variables are integers is called a Mixed-Integer Linear Programming (MILP). Unfortunately, although LP's are practically solvable in polynomial time, solving a MILP is NP-hard in general.

In its pure form, MP and the solver technology supporting it are dedicated exclusively to deterministic optimisation problems. They do not offer even basic support for modelling uncertainty. The next section presents the extension of MP known as Stochastic Programming (SP), which provides structures for representing parameters affected by uncertainty. In particular, the section attempts to answer the question about the circumstances in which uncertainty modelling is advisable and reviews the portfolio of modelling techniques. Their applicability depends on the type of uncertainty as well as the attitude of the decision-maker towards uncertainty. SP and SO are considered as different concepts. The first is a set of modelling techniques for representing uncertainty in parameters of an optimisation program. In contrast, SO is a methodology for solving SP formulations in which the uncertainty can be fully characterised by a single probability distribution that is known to the decision-maker.

### Stochastic Programming

Consider a hypothetical mathematical program in which the vector $\mathbf{x}$ contains decision variables, and the vector $\mathbf{u}$ stores parameters whose exact values are not known precisely. As a result, in the sketch of the formulation below, uncertainty is not formally specified. The issue is handled in subsequent sections by transforming the formulation according to various techniques for modelling uncertainty.

$$\begin{aligned} & \min_{\mathbf{x}} f(\mathbf{x}, \mathbf{u}) & \\ \text{s.t.:} \quad & g_i(\mathbf{x}, \mathbf{u}) \leq 0 & \forall i \in 1..m \\ & \mathbf{x} \in \mathbb{R}^n & \end{aligned}$$

Let consider arbitrary functions $f$ and $g_i$ $\forall i \in 1..m$. The properties of the functions will become important in discussion of Computational Complexity which is deferred until Section 2.2.

The treatment of uncertain coefficients is restricted to the objective function and the left-hand side of inequality constraints. Ultimately, such treatment of constraints is without the loss of generality. Uncertainty affecting the right-hand side of the inequality constraint can be moved to the left-hand side. The lack of equality constraints is not a coincidence because an experienced modeller would avoid using such constraints when uncertain parameters are present [7].

In presenting SP, the term a realisation of uncertainty and a scenario appear frequently and have the following meaning. The realisation of uncertainty is the process of revealing values assumed by uncertain parameters. The assignment of values observed for all uncertain parameters at the same time is called a scenario. Understandably, values of uncertain parameters are not known precisely to the decision-maker at the time when the optimisation problem is being solved. The uncertainty is revealed after the final solution is selected.

### Mean Value Problem

The following section considers the consequences of setting parameters affected by uncertainty to some nominal values. For instance, the decision-maker may use a sample of scenarios containing values assumed by uncertain parameters in the past to estimate the expected value for each parameter. The optimisation problem in which expected values replaced uncertain parameters is called the mean value problem. As a result, the following formulation is obtained.

$$\begin{aligned} & \min_{\mathbf{x}} f(\mathbf{x}, \mathbb{E}_{\mathbb{P}}(\tilde{\mathbf{u}}))) & \\ \text{s.t.:} \quad & g_i(\mathbf{x}, \mathbb{E}_{\mathbb{P}}(\tilde{\mathbf{u}})) \leq 0 & \forall i \in 1..m \\ & \mathbf{x} \in \mathbb{R}^n & \end{aligned}$$

The random variable $\tilde{\mathbf{u}}$ is adorned with a tilde to emphasise it is distributed following some known probability distribution $\mathbb{P}$. Formulations presented in the sequel follow the same convention.

The mathematical program is essentially an attempt to recast the problem affected by uncertainty to the model suitable for the deterministic case. Understandably, researchers who made methodological contributions to modelling uncertainty motivated their efforts by presenting examples for which adopting the strategy above is counterproductive.

Considering theoretical arguments, replacing a random variable by its expectation in a convex function underestimates the expected value of the function [16]. It is a conclusion derived from Jensen's Inequality for the Probability Theory 2.6.

$$f(\mathbf{x}, \mathbb{E}_{\mathbb{P}}(\tilde{\mathbf{u}})) \leq \mathbb{E}_{\mathbb{P}}(f(\mathbf{x}, \tilde{\mathbf{u}})) \tag{2.6}$$

Ultimately, one could create an instance of an optimisation problem in which the difference between the solution to the stochastic problem and the solution to the mean value problem could be arbitrarily large. The procedure to create such instances for the shortest path problem with a deadline or the travelling salesman problem is explained by [28][Proposition 3.6]. Most importantly, similar results are not limited to artificial instances built to emphasise the fallacy of solving the mean value problem formulation. The researchers [7] who studied optimal solutions to the LP problems from the Netlib benchmark reported that introducing small perturbations into parameters that are described with high numerical precision may lead to significant violations of constraints by optimal solutions to deterministic problems. Specifically, they analysed the effect of perturbations in inequality constraints which contained coefficients that were floating-point numbers with more than two digits after the decimal point. Understandably, such parameters in science and engineering applications are likely to be affected by uncertainty. After introducing perturbations at 1% level of the nominal value in selected parameters, deterministic solutions violated the inequality constraints by at least 5% in 30% of the benchmark problems. In extreme cases, the violation was more than fourfold [7]. Similar observations generalise to the whole class of LP.

By geometric arguments, an optimal solution of an LP is located in the vertex of the polyhedron defined by the constraints. At the vertex, a subset of constraints is satisfied with equality, without slack. Consequently, if those constraints contain uncertain parameters which are multiplied by decision variables that assume large values, then random perturbations may lead to significant violations of the constraints.

To conclude the discussion of using nominal values instead of modelling uncertainty, whether ignoring uncertainty may lead to infeasibility in practice and the cost associated with the solution that remains feasible in the face of relevant perturbations seems to be problem dependent [7]. For instance, the optimal solutions to the mean value problem and the stochastic program were similar in the routing problems considered by [28], when the deadlines were either tight or loose. Ultimately, recasting the problem formulation affected by uncertainty to the mean value problem should be supported by the analysis of potential consequences. The methodology adopted by [7] is an example of a formal approach to answer this question.

### Optimisation for the Worst-Case Scenario

Intuitively, solutions that satisfy all constraints affected by uncertainty for all values uncertain parameters may assume should be appealing in application domains in which a violation of an uncertain constraint may lead to an irrecoverable loss, i.e., a collapse of a building [6] or losing a spacecraft. Another well-motivated example arises whenever the difference in cost between a solution that always remains feasible and the optimal solution to the nominal problem is small.

Robust Optimisation is a methodology that aims to find a solution that remains feasible for all scenarios and is optimal in the worst-case. In other words, the optimal solution in RO has the lowest maximum cost evaluated for all possible scenarios. A feasible solution to a RO problem is called a robust solution for short. The range of values an uncertain parameter may assume is modelled using a set, referred to as an uncertainty set. In principle, it should contain all realisations of uncertainty for which the robust solution should remain feasible.

Suppose the vector of uncertain parameters $\mathbf{u}$ could be equal to any vector contained in the set $\mathcal{U}$. Consequently, one could formulate the optimisation problem below.

$$\begin{aligned}
&\min_{\mathbf{x}} \max_{\mathbf{u} \in \mathcal{U}} f(\mathbf{x}, \mathbf{u}) \\
\text{s.t.:} \quad & g_i(\mathbf{x}, \mathbf{u}) \leq 0 && \forall \mathbf{u} \in \mathcal{U} \quad \forall i \in 1..m \\
& \mathbf{x} \in \mathbb{R}^n
\end{aligned}$$

Uncertainty sets model values uncertain parameters could assume using purely geometrical arguments. Besides support, no explicit assumptions are enforced on the probability distribution, which generates values for uncertain parameters. For that reason, uncertain parameters are not treated as random variables. This fact is expressed by not adorning the vector of uncertain parameters with the tilde. If the uncertainty set is a singleton, then the formulation reduces to the deterministic case.

Overall, the formulation above is called a Robust Counterpart (RC). Although suitable for explaining the principles behind RO, it will not be accepted by a MIP solver as a valid formulation. Firstly, the min-max objective needs to be transformed into an equivalent representation that is exclusively a minimisation or maximisation problem. Secondly, the number of constraints is semi-infinite and should be replaced by a finite set of constraints. Section 2.2.2 presents the RO framework and discusses the relevant literature in detail. It demonstrates the transcription techniques which turn high-level RO formulations into valid mathematical programs. Furthermore, it presents definitions of popular uncertainty sets and outlines the tractability results.

The straightforward principles of RO are overly simplistic for some applications. In particular, the RO formulation may be infeasible if the uncertainty set is large. On the other hand, the solution may exist if the decision-maker replaces the absolute constraint satisfaction requirement by a probabilistic guarantee at some relevant confidence level [29]. Secondly, for problem instances that remain feasible, the worst-case scenario may not be a representative scenario leading to overly conservative decisions. Consequently, it may be adequate to optimise the expected cost of a solution. Nonetheless, both workarounds require a framework that models uncertain parameters as random variables.

### Optimisation of Expected Value

Parameters affected by uncertainty are modelled as random variables in SO and DRO. The methodologies are detailed in Sections 2.2.1 and 2.2.3, respectively. For the presentation of the modelling techniques here, it is sufficient to mention the assumption in SO that the probability distribution is known precisely to the decision-maker or can be estimated and considered as unique. Uncertainty which can be represented precisely as a given probability distribution is called risk [30]. On the other hand, if the decision-maker is not sure about the exact definition of the probability distribution, the DRO methodology suggests to define a set of probability measures and perform optimisation for the worst-case probability distribution from that set. Uncertainty affected by the lack of confidence in the probability distribution is called ambiguity [30], and the set of probability measures defined to represent such uncertainty is called an ambiguity set. Section 2.2.3 outlines the approaches to define ambiguity sets proposed in the literature.

Following the principles of SO, it is assumed that uncertain parameters are distributed according to the given probability measure $\mathbb{P}$. As a result, one can derive the problem formulation below.

$$\begin{aligned} & \min_{\mathbf{x}} \mathbb{E}_{\mathbb{P}}(f(\mathbf{x}, \tilde{\mathbf{u}})) \\ \text{s.t.:} \quad & \mathbb{P}(g_i(\mathbf{x}, \tilde{\mathbf{u}}) \leq 0 \quad \forall i \in 1..m) \geq 1 - \epsilon \\ & \mathbf{x} \in \mathbb{R}^n \end{aligned} \tag{2.7}$$

For a given assignment of decision variables $\mathbf{x}$, a solution is considered feasible if Constraint (2.7) is satisfied with the prescribed probabilistic guarantee $1 - \epsilon$. A constraint of this kind is called a Chance Constraint (CC) [31]. Section 2.2.1 explains this concept in more detail.

On the other hand, the probability distribution itself is not known precisely in the face of ambiguity. For that reason, one should follow the DRO methodology and define the family of probability distributions $\mathcal{P}$, which, if appropriately described, should include the true but unknown probability distribution $\mathbb{P}$ with high confidence. Conceptually, the ambiguity set exemplifies knowledge about the probability distribution and its properties. The more vague its definition is, the larger the ambiguity set should be selected. Adopting the DRO paradigm, one can formulate the uncertain mathematical program considered in this section as follows.

$$\begin{aligned} & \min_{\mathbf{x}, \mathbb{P} \in \mathcal{P}} \mathbb{E}_{\mathbb{P}}(f(\mathbf{x}, \tilde{\mathbf{u}})) \\ \text{s.t.:} \quad & \inf_{\mathbb{P} \in \mathcal{P}} \mathbb{P}(g_i(\mathbf{x}, \tilde{\mathbf{u}}) \leq 0 \quad \forall i \in 1..m) \geq 1 - \epsilon \\ & \mathbf{x} \in \mathbb{R}^n \end{aligned} \tag{2.8}$$

Constraint 2.8 is called a Distributionally Robust Chance Constraint (DRCC). The formulation above is one of several alternatives. The variant with the DRCC is presented for the purpose of exposition. Decision-makers frequently use the worst-case expectation as the objective function but model constraints according to the RO paradigm.

Similarly to the situation with the RC, the SO and the DRO formulations are not compliant with the standard format of mathematical programs. The available reformulation techniques are presented in Sections 2.2.1 and 2.2.3, respectively.

Ultimately, DRO generalises SO and RO. Stochastic Optimisation is a special case of DRO in which the ambiguity set is a singleton, i.e., it contains only one probability distribution. On the other hand, if no assumptions on the probability distribution other than the support set are available, DRO reduces to RO. The ambiguity set in that case is a singleton with an implicit probability distribution that allocates all probability mass to the worst-case scenario.

In the remainder of the section, some more advanced modelling techniques compared to the pure evaluation of the probability and the expectation of a random variable are presented. Henceforth, the SO formulations are presented only for the ease of exposition and succinctness. The adaptation of the formulations to the DRO methodology should be straightforward. In essence, an expression involving a random variable should be evaluated for the worst-case probability distribution over a given ambiguity set. The modelling techniques are not directly transferable to RO because the framework lacks support for random variables.

### Optimisation of Value at Risk

A satisfaction problem subject to a CC could be transformed into a minimisation problem with the right-hand side of the inequality inside the CC as the objective.

$$\begin{aligned} & \min z \\ \text{s.t.:} \quad & \mathbb{P}(\mathbf{c}^\top \mathbf{x} \geq z) \leq \epsilon \\ & \mathbf{x} \in \mathcal{X} \end{aligned}$$

The optimal objective value of the optimisation problem above is called the Value-at-Risk (V@R) [32]. It used to be a prominent risk measure with ubiquitous applications in finance and engineering. Nowadays, the V@R has been partially superseded by its convex upper bound known as the Conditional Value-at-Risk (CV@R) [33], which are covered in more detail in Section 2.2.1. For a thorough discussion of the differences in mathematical properties of both measures, see [34].

### Optimisation of Certainty Equivalent

Alternative definitions of the objective functions could be more appropriate than the expected value in some application contexts. Such applications arise when the pure expected value is overly simplistic, i.e., it does not differentiate between probability distributions that have the same expectation. Similarly, it does not guard against solutions that may allow substantial losses. Finally, bare minimisation of the expected value of a function is risk neutral [14], and it may not be an adequate decision criterion because a considerable proportion of the human population tends to exhibit risk aversion. Besides Optimised Certainty Equivalent (OCE), the subsequent sections cover optimisation of a riskiness index and the expected (dis)utility, which are closely related concepts with roots in the Expected Utility Theory [35].

The Expected Utility Theory attempts to formally describe a behaviour of a rational decision-maker facing a risky decision. The theory postulates that preferences of an individual can be modelled using some utility function: $u : \mathbb{R} \to (-\infty, +\infty]$. Conceptually, when faced with two mutually exclusive decisions whose uncertain outcomes are described by random variables, i.e., $\tilde{a}$ and $\tilde{b}$, a rational decision-maker who acts following some utility function $u$ should prefer the decision that has a higher expected utility, i.e., $\mathbb{E}_{\mathbb{P}}(u(\tilde{a})) \geq \mathbb{E}_{\mathbb{P}}(u(\tilde{b}))$. Arguably, the utility functions used in literature have simple definitions, i.e., Piecewise Linear Function 2.9 [36] presented below:

$$u_{\alpha_1,\alpha_2}(x) = \begin{cases} \alpha_1 x & \text{if } x \leq 0 \\ \alpha_2 x & \text{if } x > 0 \end{cases} \tag{2.9}$$

where $\alpha_1 \in [0, 1)$ and $\alpha_2 > 1$. Function 2.9 comprises a ramp utility function $u(x) = \max\{0, x\}$ as a special case.

Exponential Function 2.10 [37] is another commonly used formula.

$$u_\alpha(x) = -\frac{e^{-\alpha x}}{\alpha} \tag{2.10}$$

Utility functions lack intercept because it does not affect the ordering of random variables induced by a given function.

On the other hand, considering minimisation of a loss, a rational decision-maker should act to minimise expected disutility. Given a utility function $u$, the disutility function $d$ is obtained using the following transformation: $d(x) = -u(-x)$.

Selecting an appropriate utility function to precisely describe the preferences of an individual could be a complicated task since decisions taken in the face of risk may depend on factors including personal circumstances of the individual, such as wealth. Although the exact definition of the utility function may be challenging to establish, its general properties have strong justification. The utility should increase as an individual accumulates more wealth (the non-decreasing property). Furthermore, the marginal increase in utility due to the favourable outcome should be lower for wealthier individuals (the convexity property).

A considerable amount of the human population is risk-averse when making rational decisions in the face of risk, i.e., they are unwilling to accept uncertain decisions for which the expected utility is zero. A possible measure of the degree of risk aversion is the absolute risk aversion coefficient [38].

$$A(x) = -\frac{u''(x)}{u'(x)} \tag{2.11}$$

An individual is risk-averse if the absolute risk aversion coefficient is positive. The coefficient is zero when the utility function is risk-neutral, i.e., the identity function, a linear function. The negative risk aversion coefficient corresponds to the risk willing attitude. The absolute risk aversion coefficient is constant for the exponential utility function.

If the decision-maker is risk-averse, the following equation relation holds for all random variables [36].

$$\mathbb{E}_{\mathbb{P}}(u(\tilde{x})) \leq u(\mathbb{E}_{\mathbb{P}}(\tilde{x})) \tag{2.12}$$

The inequality is tight if the decision-maker is risk-neutral. Suppose the utility function is invertible, i.e., the function is strictly increasing, then inequality 2.12 could be used to derive a lower bound on the expected value of the random variable, which is known as the Classical Certainty Equivalent (CCE) [36].

$$\mathrm{C}_{\mathrm{CCE}}(\tilde{x}) = u^{-1}\mathbb{E}_{\mathbb{P}}(u(\tilde{x})) \leq \mathbb{E}_{\mathbb{P}}(\tilde{x}) \tag{2.13}$$

The CCE could be used to induce ordering between random variables. The ordering is consistent with the utility function. Furthermore, the CCE has a plausible interpretation as a sure reward uncertain gamble should generate to make the decision-maker indifferent whether to accept or reject the bet.

There are alternative definitions of certainty equivalents that have similar intuitive interpretations and require less stringent assumptions on the properties of utility functions.

The Optimised Certainty Equivalent (OCE) [36] is compatible with a family of concave utility functions that preserve the inequality $u(x) \leq x \quad \forall x \in X$ and may not be invertible.

$$\mathrm{C}_{\mathrm{OCE}}(\tilde{x}) = \sup_{\gamma \in \mathbb{R}} \{\gamma + \mathbb{E}_{\mathbb{P}}(u(\tilde{x} - \gamma))\} \tag{2.14}$$

The quantity intuitively corresponds to the present value of the future uncertain outcome [36].

Finally, the u-Mean Certainty Equivalent (uMCE) defined below is interpreted as the sure amount which can be subtracted from the random variable to make the rational decision-maker indifferent about the gamble [39].

$$\mathbb{E}_{\mathbb{P}}(u(\tilde{x} - \mathrm{C}_{\mathrm{uMCE}}(\tilde{x}))) = 0 \tag{2.15}$$

Overall, familiarity with certainty equivalents and utility functions is vital for defining objective functions that adequately model the behaviour of decision-makers in the face of risk. The interest in this area has been rekindled recently following the development of new decision criteria which improve modelling techniques available for CC's.

The downside of the V@R model presented in Section 2.1 is ignoring the magnitude of constraints' violations which may be arbitrarily large. Arguably, a sensible decision criterion would minimise the probability that constraints are violated and incentivise solutions for which the likelihood of a constraint violation is inversely proportional to its magnitude. Riskiness indices are the decision criteria that have such appealing properties.

The first riskiness index, the Economic Index of Riskiness, has been proposed by Aumann and Serrano [40]. It is defined as the reciprocal of the constant absolute risk aversion coefficient of the decision-maker who is indifferent to accepting or rejecting the decision [40]. The researchers have described the properties of the index and derived its analytical formula for a random variable that follows a normal distribution. The definition of the index was further extended by [41] to handle edge cases: the decisions which never cause loss and the decisions that are expected to incur a loss. The same researcher derived analytical formulas for several well-known probability distribution functions [41].

Other authors proposed alternative definitions of riskiness indices: the Requirements Violation Index (RVI) [42], the Essential Riskiness Index (ERI) [43] and the Service Fulfilment Index (SFI) [16]. Interestingly, all indices mentioned above were constructed following the same approach. For a given random variable, the index assumes the value of the lowest parameter for which a selected certainty equivalent defined using a given disutility function satisfies some prescribed inequality constraint. For instance, the ERI defined below was first proposed by [43].

$$\rho_{\text{ERI}}(\tilde{x}) = \min_{\alpha \in \mathbb{R}_{\geq 0}} \{\alpha \ \mid \mathbb{E}_{\mathbb{P}}\{(\max\{\tilde{x}, -\alpha\}) \leq 0\} \tag{2.16}$$

It is formulated using the uMCE with a ramp disutility function. Similarly, the RVI [42] is built on top of the CCE for an exponential disutility function. The SFI [16] is based on the CV@R, which is a special case of OCE, and the ramp disutility function.

Overall, riskiness indices are defined to follow the same set of conventions and properties, illustrated below for an arbitrary riskiness index represented using the symbol $\rho(\tilde{x})$. These properties hold for each riskiness index mentioned above.

1. $\rho(\tilde{x}) = \infty$ if $\mathbb{E}_{\mathbb{P}}(\tilde{x}) > 0$

2. $\rho(\tilde{x}) = 0$ if $\mathbb{P}(\tilde{x} \leq 0) = 1$

3. $\rho(\tilde{x})$ is a convex function.

4. $\mathbb{P}(\tilde{x} > \theta\rho(\tilde{x})) \leq \phi(\theta^{-1})$

Properties 1 and 2 are conventions. Property 3 has important implications allowing for effective evaluation of a given riskiness index. Finally, Property 4 attests that the probability of a violation of a constraint is inversely proportional to the magnitude of the violation. Understandably, alternative definitions of riskiness indices differ in the strength of the probabilistic guarantees they provide. The RVI offers stronger guarantees than the SFI and the ERI. However, the RVI is also more difficult to optimise compared to the remaining formulations [43]. Ultimately, riskiness indices made maximisation of probability that a CC is satisfied obsolete. This transition is apparent in the Vehicle Routing Problem with Time Windows (VRPTW) focused on minimising the delays in arrival to customer locations [42, 43].

To conclude, the section presented high-level modelling techniques for defining programs affected by uncertainty, such as expectations, CC's, OCE's, etc. They can be evaluated for a single probability distribution or the worst-case distribution in the ambiguity set. This approach was adopted to distinguish between the definition of the optimisation problem and solving it. The formulation of the right problem to optimise is the responsibility of the decision-maker, whereas software frameworks could partially or fully facilitate the process of deriving the reformulation and finding a solution. During a reformulation, high-level structures are translated into pure mathematical programs which consist only of deterministic coefficients, a finite number of constraints, and decision variables. The following sections will introduce the reformulation techniques and discuss the computational complexity of the resultant programs.

## 2.2 Frameworks for Modelling Uncertainty

### 2.2.1 Stochastic Optimisation

Stochastic Optimisation assumes that uncertain parameters in the formulation of the optimisation problem are modelled as random variables whose probability distribution functions are known precisely. The probability distributions should be independent of the solution vector. The expressions containing random variables should also be well defined and finite for every feasible solution [44]. The expressions involving operations on variables present in constraints and the objective function can be computed from the probability distribution definition. It is convenient when closed-form expressions or closed-form approximations are available. For that reason, researchers commonly assume a normal distribution, Bernoulli distribution or discrete uniform distribution defined over the set of scenarios.

On the other hand, formulations that require multidimensional integration, which is computationally prohibitive due to the course of dimensionality [14, 45], are avoided. Similarly, even for moderate-sized stochastic programs, expressions involving random variables could be challenging to evaluate exactly due to the prohibitive number of scenarios to consider. Motivated by the need to harness the computational complexity, the initial (a.k.a. "true") stochastic program could be approximated by the optimisation problem in which some sample of a finite size replaces the complete set of scenarios. It is either available beforehand as a dataset of scenarios or could be obtained by sampling the probability distribution whose definition is known.

The coverage of SO in this section is restricted to the Sample Average Approximation (SAA) and the CV@R. The knowledge of these techniques should be sufficient to use SO in SP and to understand contributions to applications described in the literature cited in the thesis. For a more thorough treatment of the topic, the interested reader is redirected to the book [15].

### Sample Average Approximation

The Sample Average Approximation (SAA) is one of the most widely applied sampling-based approximation schemes for stochastic programs. It was developed to approximate the expected value of a function that does not have a closed-form expression, or the expected value is difficult to compute from the definition. An example optimisation problem that could be approximated using the SAA is defined below.

$$\min_{\mathbf{x}\in\mathcal{X}} \mathbb{E}_{\mathbb{P}}(f(\mathbf{x},\tilde{\mathbf{z}})) \tag{2.17}$$

where $\mathcal{X}$ is the set of feasible solutions, $f$ is some function which takes an input a candidate solution $\mathbf{x}$ and a random variable $\tilde{\mathbf{z}}$ that is distributed according to some probability distribution $\mathbb{P}$.

Instead of computing the expected value from the definition, the SAA adopts a pragmatic Monte-Carlo approach. The expected value is estimated by the arithmetic mean computed over a single sample containing a finite number of independent and identically distributed scenarios sampled from the probability distribution $\mathbb{P}$ [44].

$$f_{\mathcal{Z}^N}(\mathbf{x}) = \frac{1}{|\mathcal{Z}^N|}\sum_{\mathbf{z}\in\mathcal{Z}^N} f(\mathbf{x},\mathbf{z}) \tag{2.18}$$

The estimator $f_{\mathcal{Z}^N}(\cdot)$ estimates the expected value of the function $f$ from Formulation 2.17. The sample $\mathcal{Z}^N$ contains $N$ independent and identically distributed scenarios $\{\mathbf{z}^1,\ldots,\mathbf{z}^N\}$. Estimator 2.18 is called the standard Monte Carlo estimator [44]. The estimate is a random variable itself as it depends on the concrete sample $\mathcal{Z}^N$.

Taken together, the subject of the SAA is solving the following optimisation problem to approximate the optimal objective value and the optimal solution of the "true" problem.

$$\min_{\mathbf{x}\in\mathcal{X}} f_{\mathcal{Z}^N}(\mathbf{x}) \tag{2.19}$$

The resultant optimisation program is deterministic because the sample is fixed to some concrete realisation. Furthermore, it is apparent that for the approximation scheme to work effectively, the expression under the expectation should be easy to calculate for the given input. Let denote $\mathcal{S}^*_{\mathcal{Z}^N}$ to be the set of the optimal solutions and $\nu_{\mathcal{Z}^N}$ to be the optimal objective value of the SAA optimisation problem. The set of the "true" optimal solutions and the "true" objective value are $\mathcal{S}^*$ and $\nu$, respectively. Intuitively, the quality of these estimates depends on the realisation of the sample and its size; hence $\mathcal{Z}^N$ appears in the subscript for their counterparts obtained by solving the SAA problem.

The dependence on the particular realisation of a sample is less pronounced when estimators are strongly consistent, i.e., when they converge almost surely (a.k.a., with probability one) to their "true" counterparts as the sample size increases to infinity. For instance, the standard Monte-Carlo estimator of the objective value used in the SAA is strongly consistent when the property known as uniform convergence holds [44]:

$$|f_{\mathcal{Z}^N}(\mathbf{x}) - \mathbb{E}_{\mathbb{P}}(f(\mathbf{x}, \tilde{\mathbf{z}}))| < \epsilon \qquad \forall N \geq N_0 \quad \forall \mathbf{x} \in \mathcal{X} \tag{2.20}$$

Therefore, for any sample $\mathcal{Z}_N$ whose size is greater or equal to some threshold sample size $N_0$, the difference between the estimate of the expectation and the "true" value is bounded above by some positive $\epsilon$. For instance, the uniform convergence arises when the function $f$ is convex and continuous in the decision variables $\mathbf{x}$ regardless of the value assumed by the random variable [44]. Moreover, the feasible set $\mathcal{X}$ should be compact and convex.

Intuitively, the sample size is a compromise between the quality of the approximation and the effort required to evaluate the expectation. The effectiveness in which the approximate values reflect their true counterparts with respect to increasing the sample size is known as the convergence rate. The sequel explores that subject in more detail.

Estimator 2.18 is biased as its expectation underestimates the "true" value [44]:

$$\mathbb{E}_{\mathbb{P}}(\nu_{\mathcal{Z}^N}) \leq \nu^* \tag{2.21}$$

Suppose the variance of the estimator is finite. As a consequence of the Central Limit Theorem, the bias of the estimator converges in distribution to a normal distribution with zero mean [44].

$$\sqrt{N}\left(f_{\mathcal{Z}^N}(\mathbf{x}) - \mathbb{E}_{\mathbb{P}}(f(\mathbf{x}, \tilde{\mathbf{z}}))\right) \xrightarrow{d} \tilde{v}(\mathbf{x}) \sim \mathcal{N}(0, \sigma^2(\mathbf{x})) \tag{2.22}$$

where $\sigma^2(\mathbf{x})$ is the variance of the expression $f(\mathbf{x}, \tilde{\mathbf{z}})$ and $N$ is the sample size. Consequently, the estimator $f_{\mathcal{Z}^N}(\mathbf{x})$ converges in distribution to a normal distribution with the mean $\mathbb{E}_{\mathbb{P}}(f(\mathbf{x}, \tilde{\mathbf{z}}))$ and variance $\frac{\sigma^2(\mathbf{x})}{N}$. Therefore, the variance of the estimator is inversely proportional to the sample size.

Furthermore, suppose the objective function is continuous in the Lipschitz sense, then the analogous result could be derived for the set of optimal solutions $\mathcal{S}^*$ [44]:

$$\sqrt{N}\left(f_{\mathcal{Z}^N}(\mathbf{x}^*) - \mathbb{E}_{\mathbb{P}}(f(\mathbf{x}^*), \tilde{\mathbf{z}})\right) \xrightarrow{d} \inf_{\mathbf{x} \in \mathcal{S}^*} \tilde{v}(\mathbf{x}) \tag{2.23}$$

where $\mathbf{x}^* \in \mathcal{S}^*$.

Unless $\mathcal{S}^*$ is a singleton, the random variable $\tilde{v}(\mathbf{x})$ on the right-hand side is not normally distributed. Hence, $f_{\mathcal{Z}^N}(\mathbf{x}^*)$ does not converge to a normal distribution in general as the sample size grows to infinity. Furthermore, one could demonstrate that when a problem has multiple optional solutions, the rate of convergence cannot be faster than $N^{-\frac{1}{2}}$ [46]. The bias and its relation to the sample size can be described more precisely after a given optimisation problem and some probability distribution are assumed, see e.g. [47]. The restriction on convergence rate does not apply for problems that have a unique optimal solution [44].

The stopping criterion for deterministic optimisation is reaching a satisfactory optimality gap. Specifically, the optimality gap of the "true" problem is expressed using the equation:

$$\delta(\mathbf{x}) = f(\mathbf{x}, \tilde{\mathbf{z}}) - \nu^* \tag{2.24}$$

The optimality gap in the SAA problem could be estimated using the expression below:

$$\delta_{\mathcal{Z}^N}(\mathbf{x}) = f_{\mathcal{Z}^N}(\mathbf{x}) - \nu_{\mathcal{Z}^N} \tag{2.25}$$

More precisely, the formula estimates the upper bound of the optimality gap due to the bias of the estimator of the objective value $\nu$ emphasised in Equation 2.21. The operands in Expression 2.25 can be evaluated for a given sample. Besides dependence on a concrete sample, Estimator 2.25 is also not asymptotically normal as the estimator $\nu_{\mathcal{Z}^N}$ neither converges in distribution to a normal distribution in general.

For that reason, the confidence interval for the optimality gap could be computed through the bootstrapping procedure or by applying so-called the batch-means approach [48]. The latter requires a suite of $K$ samples $\mathcal{Z}_1^N, \dots, \mathcal{Z}_K^N$ as input. According to the batch-means approach, the mean of the estimates for individual samples is the estimate of the optimality gap:

$$\bar{\delta}_K(\mathbf{x}) = \frac{1}{K} \sum_{k=1}^{K} f_{\mathcal{Z}_k^N}(\mathbf{x}) \tag{2.26}$$

The $(1-\alpha)$-level confidence interval of the estimate above is:

$$\left[0, \bar{\delta}_K(\mathbf{x}) + \frac{z_\alpha \sigma_K}{\sqrt{K}}\right] \tag{2.27}$$

where $z_\alpha$ is $(1-\alpha)$-quantile of the standard normal distribution and $\sigma_K$ is the standard deviation of the sample, i.e., $\sigma_K^2 = \frac{1}{K-1} \sum_{k=1}^{K} (\delta_{\mathcal{Z}_k^N}(\mathbf{x}) - \bar{\delta}_K(\mathbf{x}))^2$.

Overall, Estimator 2.26 enforces only mild additional assumptions, such as finite second moments. On the other hand, it relies on the Central Limit Theorem, so the number of samples should be at least 20-30, which could lead to computational difficulties in cases when SAA is challenging to compute. Specifically, the literature indicates that application contexts with high variance and a flat objective function are unsuitable for the SAA approach [49]. Other estimators of the optimality gap proposed in the literature reduce the overall computational complexity [46] at the expense of more stringent assumptions such as the compactness of the feasible set, etc. Furthermore, the optimality gap estimate is often larger to provide equivalent probabilistic guarantees to Estimator 2.26 [44].

Similarly to deterministic optimisation, a substantial optimality gap in SO is not a definitive indicator of the poor quality of the incumbent solution. Other plausible explanations could be the high variance of the optimality gap estimator or the intrinsic bias. Formulas presented before suggest that increasing sample size should be an effective way of reducing bias. On the other hand, variance could be reduced by applying more advanced sampling techniques, such as stratified sampling, see [44] for the survey of variance reduction techniques.

Nonetheless, the bias reduction techniques are not infallible and may even contribute to the overestimation of the upper bound of the optimality gap. That enforces adopting larger sample sizes to attain the desired theoretical probability guarantees. For that reason, it is common to encounter in practice the SAA applied in an iterative manner with increasing sample sizes until a solution with the desired optimality gap estimate is found. Furthermore, for increasing the confidence in the estimate, the optimality gap estimation could utilise a larger sample than the one used for computing the optimal solution, see, e.g., [50]. In practical applications, even small sample sizes are sufficient to find solutions that are close to optimal solutions for the “true” problem [29].

In a similar vein to the estimation of the optimality gap and its confidence intervals, one could derive probability guarantees that the optimal solution found by SAA is the optimal solution to the “true” problem. Specifically, the authors [29] showed that given a sufficiently large sample, the optimal solution to the SAA problem is almost surely the optimal solution to the “true” problem. Similarly, a marginal increase in the sample size leads to an exponential increase in the likelihood that the approximation scheme converged to the optimal solution of the “true” problem [29]. Intuitively, the required sample size grows logarithmically with the size of the set of feasible solutions and the reciprocal of the probabilistic guarantee required [29]. However, conducting such analysis and finding the appropriate sample size using theoretical derivations requires information about the problem that could be challenging to obtain, such as the size of the feasible set. Similarly to the estimation of the optimality gap, the computed sample size could be larger than the sample size effectively needed to provide the desired probability guarantee [29].

Decomposition techniques such as Bender’s decomposition extend scalability limits of the SAA to large sample sizes. The authors [28] studied two-stage stochastic routing problems: the shortest path problem and the travelling salesman problem, in which the objective function was the sum of the total travel time and the expected violation of the deadline. For two-stage problems, it becomes imperative to assume a relatively complete recourse, i.e., the second-stage problem is feasible for each first-stage solution and each scenario. The authors demonstrated that the number of optimality cuts generated and the number of branch-and-bound nodes explored does not vary significantly with the sample size [28]. It is a positive result which confirms that the computational time observed in practice grows linearly with the sample size. Consequently, the SAA with Bender's decomposition is a scalable approximation scheme.

### Conditional Value-at-Risk

The presence of CC's may pose a challenge for SO because their evaluation is NP-hard for arbitrary distributions. For instance, evaluation of the distribution of a weighted sum of independent random variables that are uniformly distributed is an intractable problem [51]. Individual chance constraints can be approximated using the CV@R [34, 52], whose definition is provided below.

$$\mathrm{CV@R}_{1-\epsilon}(\tilde{x}) = \min_{\beta} \left\{ \beta + \frac{1}{\epsilon}\mathbb{E}_{\mathbb{P}}([\tilde{x} - \beta]^{+}) \right\}$$

Conditional Value-at-Risk is the tightest approximation scheme possible for individual CC's [51]. If the closed-form expression of the CV@R is not available for the given probability distribution, the risk measure could be approximated using the SAA. Otherwise, one could replace the probabilistic constraints with a collection of constraints defined for each scenario from the sample [53]. If the sample size or the probabilistic guarantee make solving the formulation effectively intractable, then bounds on the CV@R developed for RO and DRO [54] may provide a viable alternative.

Formulations in which a single probabilistic guarantee binds two or more CC's are called joint chance-constrained problems. A pragmatic approach to solving such formulations is based on Bonferroni's inequality. A joint constraint is decomposed into individual CC's restricted by new, tighter probabilistic guarantees whose sum is bounded from below by the initial probability guarantee [55]. However, this approach may be overly conservative if constraints are correlated because the new bounds satisfying Bonferroni's inequality will be tighter than the original formulation [54]. Furthermore, it is not clear how the values of the individual probabilistic guarantees should be derived [54]. Such obstacles motivated the authors to develop a new bound on a joint CC that is tighter than the decomposition of a joint constraint relying on Bonferroni's inequality, albeit this approach is based on RO.

Taken together, SO used to be the dominating approach in SP until the end of the last century. The practitioners faced with the lack of better alternatives had to accept or bypass the following limitations highlighted by [14, 56]. Arguably, the assumption that the probability distribution is known precisely does not have strong justification. Moreover, solving SO formulations is effectively intractable for multidimensional probability distributions. Finally, out-of-sample performance may not be satisfactory.

Alternative methodologies free of the limitations mentioned above, RO and especially DRO, will be covered in the following sections. To conclude, it seems that SO, which is a less advanced methodology developed for the same purpose, has been superseded by DRO. This transition is without any loss because SO is a special case of the DRO. Stochastic Optimisation remains an attractive framework for stochastic programs in which the worst-case probability distribution can be constructed [14].

### 2.2.2 Robust Optimisation

Robust Optimisation is a solution methodology developed for deriving tractable reformulations of the optimisation problems whose parameters are affected by uncertainty. Section 2.1 briefly mentioned RO and introduced the concept of the uncertainty set. The following section provides a thorough treatment of RO covering the core assumptions, reformulation techniques, important uncertainty sets and the complexity results. Unless stated otherwise, the scope is restricted to LP's. The section concludes with a brief outline of special cases for which attractive tractability results are available.

### Core Assumptions

Robust Optimisation aims to find an optimal solution that minimises or maximises a given objective function and satisfies all constraints for all possible values from the uncertainty set that uncertain parameters may assume. Understandably, the solution has to be known before the uncertain parameters assume their values. The assignments of variables that lie outside the boundaries of the uncertainty set are considered impossible. The fact that a robust solution is decided before the uncertainty is realised is referred to as making decisions here-and-now [6].

The RO framework is applicable for optimisation problems in which uncertain parameters are present only on the left-hand side of constraints. Ultimately, it is merely a format convention that has no impact on applications. Formulations that do not satisfy this property can be transformed into equivalent models which obey it. For instance, uncertain parameters on the right-hand side of the constraints can be moved to the left-hand side. In a similar vein, the objective function with uncertain parameters can be transformed into an inequality constraint bounded by a variable which is the new objective in the derived formulation.

The uncertainty set $\mathcal{U}$ that comprises values of uncertain parameters for all constraints must be decomposable into a Cartesian product of uncertainty sets defined for each constraint individually. For instance, if the problem has $m$ uncertain constraints, then $U = U_1 \times \cdots \times U_m$. Such a structure is called constraint-wise uncertainty. A formulation with the uncertainty that is not constraint-wise can always be transformed into a formulation in which uncertainty is constraint-wise [57].

For every constraint $i$, the elementary uncertainty set $\mathcal{U}_i$ should be convex. Otherwise, it must be replaced by the smallest convex set that contains $\mathcal{U}_i$. Replacing the uncertainty set $\mathcal{U}_i$ by its convex hull in an uncertain LP has no impact on the feasibility and optimality of the solution to the original problem [6], i.e., feasible or optimal solutions for an uncertainty set $\mathcal{U}_i$ will remain such for its convex counterpart.

Having outlined the assumptions of the RO methodology, the reformulation technique to transform an RC with a semi-infinite number of constraints into a finite-size mathematical program is introduced next.

### Reformulation Example

The definition of uncertainty set has pivotal implications on the difficulty of solving the optimisation problem. This effect is illustrated using an abstract LP in which coefficients of the matrix **A** are uncertain. The following example in different variants appeared in the introductions to RO [57, 58].

$$\begin{aligned} & \min \mathbf{c}^\top \mathbf{x} & \\ \text{s.t.:}\quad & \mathbf{A}\mathbf{x} \leq \mathbf{b} & \forall \mathbf{a}_1 \in \mathbf{U}_1, \ldots, \mathbf{a}_n \in \mathbf{U}_m \\ & \mathbf{x} \in \mathbb{R}^n & \end{aligned}$$

The vector $\mathbf{a}_i$ is the i-th row of the matrix $A$. The vector can assume any value that belongs to the uncertainty set $\mathcal{U}_i$.

According to the current specification, the formulation in which uncertainty sets $\mathcal{U}_i$ for $i \in [1, \ldots, m]$ are continuous declares a semi-infinite number of constraints. Without loss of generality, one can pick some set of constraints $i \in [1, \ldots, m]$ and derive its equivalent formulation.

Alternatively, the vector $\mathbf{a}_i$ can be represented as a perturbation of nominal values, i.e., $\mathbf{a}_i = \mathbf{a}_i^0 + \mathbf{P}\boldsymbol{\zeta}_i \quad \forall \boldsymbol{\zeta}_i \in \mathcal{Z}_i$, where $\mathcal{Z}_i$ is the uncertainty set describing perturbations from the nominal vector $\mathbf{a}^0$, and $\mathbf{P}$ is the matrix of coefficients controlling the magnitude of perturbations. The vector $\boldsymbol{\zeta}$ is called the primitive uncertainty [57]. The perturbation model could be extended further to describe correlated data [8]. In the latter case, there could be $k$ independent sources of uncertainty $\mathcal{Z}^0, \ldots, \mathcal{Z}^k$ which together affect parameters which appear in a constraint.

Consequently, one can rewrite the constraint using the new representation of the uncertain vector $\mathbf{a}_i$.

$$(\mathbf{a}_i^0 + \mathbf{P}\boldsymbol{\zeta})^\top \mathbf{x} \leq b \qquad\qquad \forall \boldsymbol{\zeta} \in \mathcal{Z}_i$$

Then, the universal quantifier is replaced by embedding an inner maximisation of the expression $(\mathbf{a}_i^0 + \mathbf{P}\boldsymbol{\zeta})^\top \mathbf{x}$ over the set $\boldsymbol{Z}_i$ in the constraint. It corresponds to the worst-case reformulation [57].

$$(\mathbf{a}_i^0)^\top \mathbf{x} + \max_{\forall \boldsymbol{\zeta} \in \mathcal{Z}_i} \boldsymbol{\zeta}^\top (\mathbf{P}^\top \mathbf{x}) \leq b$$

For the purpose of the exposition, one can assume that $\mathcal{Z}$ is a polyhedron, i.e., $\mathcal{Z} = \{\boldsymbol{\zeta} : \mathbf{D}\boldsymbol{\zeta} \leq \mathbf{d}\}$. Furthermore, assuming that the polyhedron has an interior point (a.k.a., Slater's condition), the strong duality holds and one can replace the inner maximisation problem by its dual.

$$\begin{gathered}
(\mathbf{a}_i^0)^\top \mathbf{x} + \max_{\forall \boldsymbol{\zeta} : \mathbf{D}\boldsymbol{\zeta} \leq \mathbf{d}} \boldsymbol{\zeta}^\top (\mathbf{P}^\top \mathbf{x}) \leq b \\
\Updownarrow \\
(\mathbf{a}_i^0)^\top \mathbf{x} + \min_{\exists \boldsymbol{\pi} : \boldsymbol{\pi}^\top \mathbf{D} = \mathbf{P}^\top \mathbf{x}, \boldsymbol{\pi} \geq 0} \boldsymbol{\pi}^\top \mathbf{d} \leq b
\end{gathered}$$

The constraint which had to be satisfied for all $\boldsymbol{\zeta} \in \mathcal{Z}$ was replaced by a constraint that must hold for only one vector $\boldsymbol{\pi}$ of non-negative real numbers. Therefore, one can move the inner dual problem into the original formulation, which completes the reformulation.

$$\begin{array}{c} (\mathbf{a}_i^0 + \mathbf{P}\boldsymbol{\zeta})^\top \mathbf{x} \leq b \quad \forall \boldsymbol{\zeta} \in \mathcal{Z}_i \\ \mathcal{Z}_i = \{\boldsymbol{\zeta} : \mathbf{D}\boldsymbol{\zeta} \leq \mathbf{d}\} \end{array} \quad \Leftrightarrow \quad \begin{array}{r} (\mathbf{a}_i^0)^\top \mathbf{x} + \mathbf{d}^\top \boldsymbol{\pi} \leq b \\ \mathbf{D}^\top \boldsymbol{\pi} = \mathbf{P}^\top \mathbf{x} \\ \boldsymbol{\pi} \geq 0 \end{array}$$

A reformulation with analogous steps could be conducted for the uncertainty set, which is a box, an ellipsoid, a cone or an arbitrary convex set. See [57] for the overview of the resultant formulations for different classes of uncertainty sets.

The example is concluded by highlighting some insightful observations which apply to all RCs in general.

Given an RC with a convex uncertainty set, the reformulation produces a finite-size deterministic formulation with a fixed number of constraints and variables. The resultant formulation has more constraints and variables than the initial formulation, but the universal quantifier over the uncertainty set is eliminated from the formulation.

The convexity property of the uncertainty sets was essential to derive the dual reformulation of the inner maximisation problem. Analogous transformations can be applied to other convex uncertainty sets by employing either conic duality or Fenchel's duality. Although the reformulation seems to be the default approach to optimise the RC, it could also be solved directly using so-called the adversarial approach, which is described in Section 2.3.1.

The dual of the inner maximisation problem whose structure depends on the definition of the uncertainty set became a part of the final formulation. Consequently, the structure of the uncertainty set has a direct impact on the difficulty of solving the RC.

The literature on RO provides a vast array of techniques to define uncertainty sets. The utility of a given approach depends on multiple factors: the application domain, available information, the difficulty of the resultant formulation, the need to control the size of the uncertainty set, etc. The following sections review important definitions of the uncertainty sets. Uncertainty sets that attempt to enforce some distributional properties are not presented because nowadays DRO is a methodology of its own, and it is discussed separately in Section 2.2.3.

### Column-Wise Convex Uncertainty Set

The concept of using convex sets to express all possible values uncertain parameters may assume was proposed by Soyster [59] in the study of uncertain LP's in which decision variables were restricted the non-negative orthant. In contrast to constraint-wise uncertainty promoted in RO, Soyster [59] designed the uncertainty set to capture parameters that appear in the same column of different constraints. Hence, each uncertain parameter in a given constraint comes from a separate uncertainty set. Since all decision variables are non-negative, the robust solution could be computed by setting each uncertain parameter to the largest possible value it may assume, i.e., $A_{ij} = \sup_{\mathbf{a} \in \mathcal{U}_i} a_j$. Understandably, the obtained robust solutions could be overly conservative if all parameters are unlikely to assume the worst-case values simultaneously.

An important special case of the uncertainty sets pioneered by Soyster is the Box Uncertainty Set (2.28), which defines the lower and the upper bound uncertain parameters may assume. Arguably, it is the most limited information about uncertainty one could provide and an essential first step to defining more advanced uncertainty sets.

$$\mathcal{U} = \{\mathbf{a} \mid \overline{\mathbf{a}} \leq \mathbf{a} \leq \underline{\mathbf{a}}\} \tag{2.28}$$

Robust Counterpart of an uncertain LP with a box uncertainty set is an LP.

### Ellipsoidal Uncertainty Set

The Ellipsoidal Uncertainty Set (2.29) for handling uncertainty in LP's was proposed in a seminal paper by Ben-Tal and Nemirovski [10] who revived interest in RO. The primary motivation for the new uncertainty set was to provide a less conservative alternative to the box uncertainty set, which at that time was the only uncertainty set available. Intuitively, uncertain parameters modelled using an ellipsoid cannot assume together extreme perturbations from nominal values, which are located in the corners of the box.

A reasonable way to define the ellipsoidal uncertainty set is to estimate a covariance matrix for uncertain parameters. It is commonly assumed the matrix is positive definite, thus invertible. Another approach is to find an ellipsoid of minimum volume that contains the sample of historical observations of values assumed by uncertain parameters [10].

$$\mathcal{U} = \left\{ \tilde{\mathbf{a}} \;\middle|\; (\tilde{\mathbf{a}} - \mathbf{a}^0)^\top \Sigma^{-1} (\tilde{\mathbf{a}} - \mathbf{a}^0) \leq \rho^2 \right\} \tag{2.29}$$

The definition of the ellipsoidal uncertainty set contains a quadratic component.

Therefore, a RC of an uncertain LP with ellipsoidal uncertainty sets is a SOCP.

The uncertainty set contains the configuration parameter $\rho$, which has to be set by the decision-maker. Ultimately, using smaller uncertainty sets that do not cover all possible perturbations leads to less conservative solutions. The methodological approach to finding the volume of the uncertainty set is assuming some stochastic model of perturbations and studying the probabilistic guarantees that the uncertain linear inequality constraint is satisfied by the robust optimal solution. Assuming symmetric and independent perturbations $\zeta_i$ supported on the interval $[-1, 1]$, the probability that the CC $\mathbb{P}\left(\tilde{\mathbf{a}}^\top \mathbf{x}^* \leq b\right)$ is violated is bounded by $e^{-\frac{\rho^2}{2}}$, which follows from Chebyshev's inequality, see [7][Proposition 1].

### Budgeted Uncertainty Set

The Budgeted Uncertainty Set [8, 12] defined below is an alternative to the Ellipsoidal Uncertainty Set.

$$\mathcal{U}_\Gamma = \left\{ \mathbf{a} \;\middle|\; \underline{\mathbf{a}} \leq \mathbf{a} \leq \overline{\mathbf{a}}, a_i = \frac{\underline{a}_i + \overline{a}_i}{2}, i \in I, I \subseteq \{1, \ldots, n\}, \|I\| \leq \Gamma \right\} \tag{2.30}$$

Given some integer configuration parameter customarily denoted as $\Gamma$, the Budgeted Uncertainty Set allows no more than $\Gamma$ uncertain coefficients to deviate from their nominal values. For that reason, the budgeted uncertainty set is also known as a Gamma-uncertainty Set [32], and a Cardinality Constrained Uncertainty Set [58].

Overall, the uncertainty set is a generalization of the box uncertainty. In its pure form, Uncertainty Set (2.30) is non-convex due to the cardinality constraint, so the reformulation is derived for its convex relaxation [58][Theorem 1], which is a polytope. The RC of an uncertain LP with the budgeted uncertainty set is an LP.

The parameter $\Gamma$ controls the trade-off between conservativeness of the solution and the probability guarantees that uncertain constraints are satisfied for the assumed stochastic model of perturbations. For perturbations that are symmetric and affect each uncertain parameter independently, the probability that the CC $\mathbb{P}\left(\tilde{\mathbf{a}}^\top \mathbf{x}^* \leq b\right)$ is violated by the robust solution is at most $e^{-\frac{\Gamma^2}{2n}}$, where $n$ is the number of uncertain parameters [8]. As a result, bounds become sharper as the number of uncertain parameters increases.

### Norm-Bounded Uncertainty Set

Uncertainty sets can be defined as sets of perturbed vectors whose distance from the vector of nominal values is below some prescribed threshold. The distance could be measured according to a norm selected by the decision-maker. Following [60], the section presents the Norm-Bounded Uncertainty Set (2.31), where $\|\cdot\|$ is an arbitrary norm, $\mathbf{M}$ is an invertible matrix, and $\Delta \geq 0$ is the radius. A function to be called a norm must satisfy a set of properties explained in A.1. The appendix also contains example norms adopted in the literature for optimisation under uncertainty. The operator vec transforms a matrix into a vector by flattening it in the row-major order.

$$\mathcal{U} = \left\{\mathbf{A} \mid \mathbf{M} \left\|\operatorname{vec}(\mathbf{A}^0) - \operatorname{vec}(\mathbf{A})\right\| \leq \Delta\right\} \tag{2.31}$$

A linear inequality constraint with the uncertainty set bounded by the norm $\|\cdot\|$ leads to the resultant formulation [60][Theorem 2], where $\|\cdot\|^*$ denotes the dual norm.

$$\mathbf{a}_i^\top \mathbf{x} + \Delta \left\|\mathbf{M}^{\top^{-1}}\right\|^* \leq b_i \tag{2.32}$$

Overall, the Norm-Bounded Uncertainty Set provides a generic framework which unifies several important classes of uncertainty sets proposed in the literature [60], i.e., the box uncertainty set ($\{\boldsymbol{\zeta} \mid \|\boldsymbol{\zeta}\|_\infty \leq 1\}$), the ellipsoidal uncertainty set ($\left\{\boldsymbol{\zeta} \;\middle|\; \|\boldsymbol{\zeta}\|_2 \leq \rho^2\right\}$), and the budgeted uncertainty set ($\{\boldsymbol{\zeta} \mid \|\boldsymbol{\zeta}\|_1 \leq \Gamma, \|\boldsymbol{\zeta}\|_\infty \leq 1\}$).

Finally, modelling an uncertainty set as bounded is without loss of generality [32]. If an uncertainty set contains a ray (i.e., a half-line which starts at a given point and expands infinitely in one direction), then optimisation of the RC will behave as it would have hidden feasibility cuts preventing the inner maximisation from obtaining positive values in the vector of perturbations which lie on that ray. Otherwise, the inner maximisation problem would be unbounded [32].

### Data-Driven Uncertainty Set

Suppose the decision-maker has an access to a sample of $n$ past realizations of uncertain parameters, i.e., $\{\mathbf{a}_1, \ldots, \mathbf{a}_n\}$ that are independently and identically distributed.

Intuitively, an uncertainty set could be defined as a convex combination of scenarios observed in the past:

$$\mathcal{U} = \left\{ \mathbf{a} \;\middle|\; \mathbf{a} = \sum_{i=1}^{n} q_i \mathbf{a}_i,\ \sum_{i=1}^{n} q_i = 1,\ \mathbf{q} \geq 0 \right\} \tag{2.33}$$

Uncertainty Set (2.33) is a special case of the more generic uncertainty set proposed by [61]:

$$\begin{aligned} \mathcal{U}_\alpha = \operatorname{conv} \Bigg( \Bigg\{ \frac{1}{\alpha} \sum_{i \in I} p_i \mathbf{a}_i + \left( 1 - \frac{1}{\alpha} \sum_{i \in I} p_i \right) \mathbf{a}_j \ \mid i \in I, I \subseteq \{1, \ldots, n\}, \\ j \in \{1, \ldots, n\} \setminus I,\ \sum_{i \in I} p_i \leq \alpha \Bigg\} \Bigg) \end{aligned} \tag{2.34}$$

Given $\alpha = 1$, Uncertainty Set (2.34) is equivalent to Uncertainty Set (2.33). Another interesting configuration setting is $\alpha = j/n$ and $p_i = 1/n$ for some positive integer $j \leq n$. Then, Uncertainty Set (2.33) represents a convex hull of centroids of all combinations containing $j$ scenarios [61].

### Tractability of Robust Convex Optimisation

The majority of positive computational complexity results for Robust Linear Optimisation are not transferable to more general classes of uncertain convex optimisation problems. The authors of the book [6][Chapters 6-8] provide an extensive treatment of this topic, including special cases in which the RC of an uncertain SOC and PSD programs remain solvable in polynomial time.

The RC of an uncertain SOC program with a linear objective and uncertain SOC-representable constraints with a polyhedral uncertainty set is intractable [6][Chapter 5]. This negative result stems from the fact that maximisation of a convex function over a polytope is NP-hard. For instance, maximisation of a quadratic expression $\mathbf{x}^\top \mathbf{A}\mathbf{x}$, where $\mathbf{A}$ is a PSD matrix and $\|\mathbf{x}\|_\infty \leq 1$ is NP-hard, see [6][Section 5.2]. As a consequence, positive tractability results are limited to special cases.

The RC of an uncertain SOC programs if the uncertainty set $\mathcal{U}$ is an ellipsoid are PSD [9], thus tractable. The RC of an uncertain SOC with a convex hull of a finite set of scenarios as the uncertainty set or a simple interval uncertainty are also tractable [6][Chapter 6]. However, if the uncertainty set is generalized to an intersection of ellipsoids, then the RC of an uncertain SOC program becomes NP-hard [9].

The RC of an uncertain PSD constraint with an ellipsoidal uncertainty set is NP-hard [9]. The robust reformulation of an uncertain PSD program with the convex hull of a finite set of scenarios remains tractable [9][Chapter 7].

### Tractability of Robust Discrete Optimisation

The discussion of the theoretical complexity of solving uncertain DO problems is relevant only for problems whose deterministic variant is solvable in polynomial time. Within this group, the authors mention [12, 32]: the shortest path, the minimum edge independent set (matching), the minimum assignment, the minimum spanning tree, the minimum cost flow etc. Problems that are NP-hard in the deterministic case will not become tractable after their formulations are extended to account for uncertain parameters. Furthermore, it is common in the literature to assume that the structural properties of feasible solutions to DO problems are not affected by uncertainty and restrict its treatment to the cost coefficients in the objective function [12, 32, 62].

In the general case, a RC of a DO problem that is tractable in the deterministic variant is NP-hard. Interestingly, this negative result appears for unconstrained combinatorial problems, which are a subclass of DO problems restricted to binary decision variables, with an uncertain objective modelled using the uncertainty set containing two scenarios [32]. If the number of scenarios is finite, then the RC is NP-hard [32][Theorem

1]. It generalises the well-known result of [62] that the uncertain shortest path problem in a graph with two scenarios of edge costs is NP-hard. The problem becomes strongly NP-hard if the uncertainty set is a finite but unbounded set of scenarios [32][Theorem 2]. Similar results hold for arbitrary polytopes and ellipsoids [32][Theorems 4 and 5]. The authors concluded that the particular structure of a combinatorial problem is irrelevant from the complexity perspective. On the other hand, the fact that decision variables are binary is critical [32].

As in the case of non-linear RO, researchers have focused on finding uncertainty sets for specific problems or classes of problems that remain tractable. Besides the interval uncertainty in which the RC has the same complexity as the nominal problem [32], combinatorial problems with uncertainty in the cost function modelled using budgeted uncertainty have a tractable RC [12]. In particular, problems that contain $n$ binary decision variables can be solved to optimality by solving $n + 1$ deterministic instances of the original problem [12]. Consequently, combinatorial problems which have an algorithm of polynomial complexity for the deterministic case are polynomially solvable if the budgeted uncertainty affects the cost function. Similarly, a min-cost flow problem with uncertain cost modelled using the budgeted uncertainty set can be transformed into a larger deterministic min-cost flow problem [12].

The RC of a combinatorial problem with uncertainty in the cost function is NP-hard when modelled using an uncertainty set that allows for describing correlations [32]. To support this observation, the authors remarked that interval uncertainty in which perturbations are independent does not change the complexity of the original problem. The same holds for the budgeted uncertainty. On the other hand, uncertainty sets that are general ellipsoids, polyhedrons, or based on scenarios, which could model correlations, lead to intractable RC's for combinatorial problems whose deterministic case is tractable. An interesting open problem is the tractability of combinatorial optimisation problems with uncorrelated ellipsoid uncertainty set, i.e., non-zero entries are only on the main diagonal. There are no known NP-hard combinatorial problems with uncertainty in the cost function modelled using an uncorrelated ellipsoidal uncertainty set which are tractable in the nominal case [32].

### Multistage Robust Optimisation

For a broad class of optimisation problems arising in the real world, the assumption of RO to make decisions here-and-now can be relaxed. For instance, consider the problem in which the source of uncertainty is a long term forecast. It may be practical to reconsider decisions regarding the future after the real values for some uncertain parameters were observed and the forecast was updated. Problems with that characteristic are known as multi-stage problems, the variables which can be adjusted are referred to as wait-and-see variables [57], and a RC with wait-and-see variables is called an Adjustable Robust Counterpart (ARC) [11].

The formulation below presents a two-stage ARC of an uncertain LP with here-and-now decisions $\mathbf{x}$ and wait-and-see decisions $\mathbf{y}$. The model was proposed as an extension to classical RO by [11].

$$\begin{aligned} & \min \mathbf{c}^{\top}\mathbf{x} & \\ \text{s.t.:} \quad & \mathbf{Ax} + \mathbf{By} \leq \mathbf{d} & \forall \mathbf{A} \in \mathcal{A} \quad \exists \mathbf{B} \in \mathcal{B} \\ & \mathbf{x} \in \mathbb{R}^n, \quad \mathbf{y} \in \mathbb{R}^m & \end{aligned}$$

Symbols $\mathcal{A}$ and $\mathcal{B}$ denote uncertainty sets of the uncertain matrices $\mathbf{A}$ and $\mathbf{B}$, respectively. The matrix $\mathbf{B}$ is customarily called the recourse matrix [11].

An ARC leads to less conservative solutions than the classical RC. To see this, compare the order of the quantifiers next to the uncertain constraint. The set of the feasible solutions for the classical RC, i.e., $\mathbf{Ax} + \mathbf{By} \leq \mathbf{d}$ $\exists \mathbf{B} \in \mathcal{B}$ $\forall \mathbf{A} \in \mathcal{A}$, is a subset of the feasible solutions to the ARC, i.e., $\mathbf{Ax} + \mathbf{By} \leq \mathbf{d}$ $\forall \mathbf{A} \in \mathcal{A}$ $\exists \mathbf{B} \in \mathcal{B}$. See [11] for conditions when an ARC and a RC are equivalent.

Unfortunately, in contrast to positive tractability results of the RC with convex uncertainty sets, solving an ARC is in general NP-hard. This negative result holds even if $\mathcal{A}$ is an arbitrary polyhedron and the matrix $\mathbf{B}$ is fixed, i.e., the uncertainty set $\mathcal{B}$ is a singleton that contains a nominal scenario.

Positive complexity results can be obtained if the wait-and-see variables are replaced by some function of uncertain parameters $\mathbf{A}$ and $\mathbf{B}$. In particular, the ARC of an uncertain LP is tractable if the uncertainty set $\mathcal{A}$ is tractable, the matrix $\mathbf{B}$ is fixed and the variables $\mathbf{y}$ are set to an affine transformation of the matrix $\mathbf{A}$ [11], i.e., $\mathbf{y} = \mathbf{p}+\mathbf{PA}$. Overall, replacing variables by a function of other variables is known as a (parametric) decision rule. In this case, the decision rule is a function of uncertain data. An ARC with linear decision rules is called an Affine Adjustable Robust Counterpart. For a given uncertainty set, it remains in the same complexity class as the classical RC and leads to less conservative solutions. Surprisingly, despite their simplicity, linear decision rules are optimal for some problem classes, see [63]. On the other hand, some optimisation problems may become infeasible if linear decision rule approximation is applied, see [64][Problem 15].

Further research on ARC's evolved in two directions: proposing more complex decision rules which may avoid infeasibility and lead to better solutions, i.e., affine functions with auxiliary variables [65], piecewise linear functions [66], quadratic functions [67], etc., and extending the approach to support wait-and-see binary variables [68].

From a practical standpoint, it is important to recognise that decision rules in adjustable RO indicate the next stage decisions immediately based on the realisation of uncertainty. The optimality of such a decision depends on the problem and the complexity of the decision rule. On the other hand, if the second stage model can be optimised, doing so may lead to better solutions. The solution approach, which involves solving the optimisation problem at subsequent stages assuming that decisions regarding the past cannot change, is known as a folding horizon [57] or a receding horizon [58]. At every stage, the decisions related to the subsequent stage become the here-and-now variables that do not need to follow a decision rule. The solution approach could be adapted to multi-stage problems modelled using a RC, an ARC or a deterministic formulation. Computational studies on inventory problems with folding horizon indicate that classical RO may outperform Adjustable RO by taking better here-and-now decisions [57]. Furthermore, if uncertainty sets are small, deterministic formulations with folding horizon may become viable alternatives for decision-makers less concerned about the worst-case performance due to better overall cost effectiveness [57].

Multistage Robust Optimisation and the multistage algorithm introduced in Chapter 5 are considered different concepts in the thesis. In contrast to Multistage Robust Optimisation presented above, a multistage algorithm is defined as a hybrid method composed of at least two algorithms that sequentially solve an optimisation problem formulated using the same set of decision variables. The first algorithm finds a solution that will be then passed to the second algorithm.

### 2.2.3 Distributionally Robust Optimisation

Distributionally Robust Optimisation (DRO) utilises available sample data and imposes some mild assumptions on structural properties of the probability distribution functions to improve probabilistic guarantees of CC's satisfaction. Achieving equivalent probabilistic guarantees in RO with no sample data available and without imposing any assumptions on probability distribution would require larger uncertainty sets, eventually leading to overly conservative solutions.

A family of the probability distribution functions which have the same support and share common structural properties is referred to as an ambiguity set. Similarly to uncertainty sets in RO, the central feature of the ambiguity sets proposed in the literature is the availability of tractable solution approaches. Moreover, ambiguity sets have adjustable size and are designed to be parametrised from data [56]. Intuitively, the size of an ambiguity set reflects the lack of confidence about the probability distribution that generated sample data. In particular, some ambiguity sets provide means to calculate the probabilistic guarantee that the true probability distribution is contained in the set [56,69]. Understandably, such probabilistic guarantees increase with the sample size. Consequently, one can reduce the size of the ambiguity set without impacting the probabilistic guarantees if a larger data sample is available.

Overall, RO and DRO are strongly connected historically and conceptually. Distributionally Robust Optimisation started as a branch of RO by building uncertainty sets that implicitly encoded distributional information. For instance, in the context of DRCC's, one can transform an ambiguity set into an uncertainty set which offers the same probabilistic guarantee [61]. Similarly, the reformulation technique for moment-based ambiguity sets covered in Section 2.2.3 leads to a classical RC in which the uncertainty set is the support of the ambiguity set. The Statistic Hypothesis Test Ambiguity Set, presented next, exemplifies the close connection between DRO and RO.

### Statistic Hypothesis Test Ambiguity Sets

An ambiguity set could be defined as a confidence region of some statistic hypothesis test [69]. Such an ambiguity set comprises probability distribution functions that are compliant with structural properties assumed by the test and satisfy the null hypothesis for the given sample data at the desired significance level $1 - \alpha$. Ambiguity sets defined in this manner exhibit some noteworthy theoretical features. Most importantly, the confidence region of the statistical test shrinks as the size of sample data increases. Understandably, the details of this process depend on the definition of the statistical test. Some tests induce smaller ambiguity sets with the same guarantee that the probability distribution function which generated the sample data is contained within the ambiguity set [69]. Such ambiguity sets are more appropriate for the given application [69].

The use of statistic hypothesis tests to formulate ambiguity sets was pioneered by [69]. The authors proposed a generic framework to derive ambiguity sets and uncertainty sets, which provide the same predefined probabilistic guarantees that a CC is satisfied. The framework can be used if a sample of independent and identically distributed observations of the random variable generated by the true probability distribution is available and the support set of the probability distribution is known. The central concept behind the framework can be summarised in the following statement.

$$f(\mathbf{x},\mathbf{u}) \leq 0 \quad \forall \mathbf{u} \in \mathcal{U}(\epsilon) \Rightarrow \mathbb{P}^{*}\left(f(\mathbf{x}^{*},\mathbf{u}) \leq 0\right) \geq 1-\epsilon$$

Let $f(\mathbf{x},\mathbf{u})$ be a function that is concave in the vector of uncertain parameters $\mathbf{u}$ for any input vector $\mathbf{x}$. The framework helps to define an uncertainty set $\mathcal{U}(\epsilon)$ parametrised by the desired probability guarantee. The satisfaction of the constraint for any vector of uncertain parameters from the uncertainty set should imply that the CC is satisfied at probabilistic guarantee $1-\epsilon$ for the true probability distribution, which is not known.

The procedure to define an uncertainty set consists of the following steps.

1. The decision-maker defines an ambiguity set $\mathcal{P}(\hat{U}, \alpha, \epsilon)$ given the statistic test, significance level $\alpha$, desired probabilistic guarantee $\epsilon$, and the sample $\hat{U}$.

2. Subsequently, the ambiguous chance constraint is replaced by the worst-case V@R over the ambiguity set.

$$\sup_{\mathbb{P}\in\mathcal{P}} \text{V@R}_{\epsilon,\mathbb{P}}(f(\mathbf{x}, \tilde{\mathbf{u}}))$$

3. The V@R is non-convex which poses challenges for developing a tractable solution approach. For that reason, the V@R should be approximated by a convex and bounded function $g(\mathbf{x}, \hat{U}, \alpha, \epsilon)$ which is positively homogeneous in $\mathbf{x}$, i.e., for any $\hat{U}, \alpha, \epsilon$ $g(a\mathbf{x}, \hat{U}, \alpha, \epsilon) = ag(\mathbf{x}, \hat{U}, \alpha, \epsilon)$. Finding such a bound is the responsibility of the decision-maker.

4. Finally, the uncertainty set $\mathcal{U}$ is derived by analysing the following expression.

$$g(\mathbf{x}, \hat{U}, \alpha, \epsilon) = \sup_{\mathbf{u}\in\mathcal{U}(\epsilon)} \mathbf{u}^\top \mathbf{x}$$

To conclude, the approach presented above is a generic framework applicable for building ambiguity sets for DRCC involving arbitrary functions concave in the vector of uncertain parameters. Arguably, the generic nature and the flexibility of the framework might pose some difficulty for the decision-maker who remains responsible for finding the upper bound for the V@R and then deriving the uncertainty set. For that reason, the researchers [69] presented the framework on a broad range of examples that imitate different application contexts, i.e., the probability distribution with discrete or continuous supports, samples drawn either from a joint distribution or from marginal distributions separately, etc. The RC is designed to be solved using the adversarial approach, covered in Section 2.3.1. For several ambiguity sets, the researchers identified closed-form expressions or efficient algorithms for finding the violated scenarios [69].

### Probability Distance Function Ambiguity Sets

An ambiguity set could be defined as a family of probability distribution functions that remain within some prescribed distance from the given reference distribution. The Kullback-Leibler divergence [70] and the Wasserstein metric [56] are arguably the most popular distance measures between probability distributions used in the definition of ambiguity sets. Due to simplicity in parameterisation from data, a common choice of a reference distribution is a discrete uniform distribution of independent and identically distributed scenarios. The adjustable distance threshold (a.k.a. radius) provides the means to control the size of the ambiguity set. Setting the radius to zero reduces the ambiguity set to a singleton which contains only the reference probability distribution. Distributionally Robust Optimisation over such a set is equivalent to SP.

Formally, the Wasserstein distance [56] is a metric defined for the family of probability distribution functions $\mathcal{P}$ supported on the support set $\mathcal{Z}$ which satisfies the following mild assumption. Given an arbitrary norm $\|\cdot\|$ used in the definition of the Wasserstein distance, the expectation of the norm of a random variable which follows any probability distribution $\mathbb{P} \in \mathcal{P}$ is bounded, i.e., $\mathbb{E}_{\mathbb{P}}(\|\tilde{\zeta}\|) = \int_{\mathcal{Z}} \|\zeta\| \mathbb{P}(d\zeta) < \infty$. Example norms adopted in the literature on optimisation under uncertainty are listed in A.1. Random variables $\zeta_1$ and $\zeta_2$ have marginal probability distributions $\mathbb{P}_1$ and $\mathbb{P}_2$, respectively, and a joint probability distribution $\Pi$. Given the essential prerequisites introduced above, the Wasserstein distance is defined below.

$$\Delta_{\mathrm{W}}(\mathbb{P}_1, \mathbb{P}_2) = \inf_{\Pi} \left\{ \int\limits_{\mathcal{Z}^2} \|\zeta_1 - \zeta_2\| \, \Pi(d\zeta_1, d\zeta_2) \right\}$$

Overall, the Wasserstein distance could be interpreted as a minimum cost required to transform one probability distribution mass into the other [56]. Specifically, the joint probability distribution encodes the transportation plan, and its cost is assessed using the given norm.

Taken together, the Wasserstein ball of the radius $r$ centred at the reference probability distribution $\mathbb{P}_0$ is an ambiguity set.

$$\mathcal{B} = \left\{ \mathbb{P} \in \mathcal{P} \;\middle|\; \Delta_W(\mathbb{P}, \mathbb{P}_0) \leq r^2 \right\}$$

Computing the worst-case expectation of an arbitrary function over the Wasserstein ball is a non-convex optimisation problem [56]. Nonetheless, suppose the Wasserstein ball is defined for probability distribution functions with a convex and closed support set, then for some specific classes of functions. In that case, the calculation of the expected value admits a tractable reformulation. In particular, the expectation of a point-wise maximum of a finite number of functions whose reflection over the x-axis is proper, convex and lower semi-continuous can be formulated as a finite-dimensional convex problem, see [56][Theorem 4.2]. A direct consequence of the theorem leads to the noteworthy result that the expectation of a point-wise maximum of a finite number of affine functions over the Wasserstein ball defined using either $L_1$ or $L_\infty$ norm is an optimal solution of an LP [56]. Interestingly, the value assigned to the radius does not affect the size of the formulation. A similar LP reformulation is available for the expectation of the point-wise minimum of a finite number of affine functions.

In contrast to other ambiguity sets, with an additional restriction to light-tailed distributions, i.e., distributions whose probability in tail declines exponentially, DRO with Wasserstein ball offers a probabilistic upper bound guarantee on the out of sample performance, see [56][Theorem 3.4]. Such a guarantee allows for a meaningful selection of the radius $r$, i.e., for a given performance guarantee $1 - \alpha$ and a sample size $n$, the radius is proportional $log(\alpha^{-1})/n$ [56]. As a result, for some fixed confidence $1 - \alpha$, the radius of the Wasserstein ball could be reduced if a larger data sample is available. Eventually, if the function whose expected value is optimised is upper semi-continuous and bounded, then the upper bound on the out-of-sample performance converges to the optimal value of the objective [56][Theorem 3.6]. The theoretical results outlined above provide strong justification for using Wasserstein ambiguity sets. Nonetheless, for practical applications, the researchers [56] recommend using machine learning techniques, such as bootstrapping, i.e., the holdout method or cross-validation, which may yield stronger out-of-sample performance guarantees for the same radius.

Similar guidelines are relevant for other probabilistic guarantees derived from general-purpose inequalities [69].

### Moment-Based Ambiguity Sets

Moment-based ambiguity sets define families of probability distribution functions by enforcing constraints on moments. Historically, the first ambiguity set of this kind defined using constraints on mean and covariance (a.k.a the first and the second moment, respectively) has been proposed by Delange and Ye in the seminal paper [13]. The authors considered the following ambiguity set:

$$\mathcal{D} = \left\{ \begin{array}{l} \mathbb{P}(\tilde{\mathbf{u}} \in \mathcal{U}) = 1 \\ (\mathbb{E}_{\mathbb{P}}(\tilde{\mathbf{u}}) - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbb{E}_{\mathbb{P}}(\tilde{\mathbf{u}}) - \boldsymbol{\mu}) \leq r_1 \\ \mathbb{E}_{\mathbb{P}}((\tilde{\mathbf{u}} - \boldsymbol{\mu})(\tilde{\mathbf{u}} - \boldsymbol{\mu})^\top) \preceq r_2 \boldsymbol{\Sigma} \end{array} \right\}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are the estimates of mean and covariance of some unknown probability distribution defined over the support set $\mathcal{U}$. The interior of the support set is assumed to contain $\boldsymbol{\mu}$. Furthermore, $\boldsymbol{\Sigma}$ is PSD, thus invertible. The ambiguity set contains probability distributions defined over the support set whose mean is located within an ellipsoidal uncertainty set centred at $\boldsymbol{\mu}$ and covariance matrix belongs to a PSD cone intersected with non-negative orthant. Parameters $r_1$ and $r_2$ represent confidence in the moment estimates.

The authors [13] studied the computational tractability of the following optimisation problem.

$$\min_{\mathbf{x} \in \mathcal{X}} \max_{\mathbb{P} \in \mathcal{D}} \mathbb{E}_{\mathbb{P}}(h(\mathbf{x}, \tilde{\mathbf{u}}))$$

The problem can be reformulated into a convex optimisation problem and solved to a given precision by an ellipsoid method. Computational complexity is polynomial in the number of decision variables and uncertain parameters assuming the following preconditions hold. Both the support set $\mathcal{U}$ and the set of feasible solutions $\mathcal{X}$ are convex, closed, bounded and have polynomial-time separation oracles. The function $h(\mathbf{x}, \mathbf{u})$ is a point-wise maximum of a finite set of functions. Each function is convex in $\mathbf{x}$ and concave in $\mathbf{u}$. Moreover, their subgradient in $\mathbf{x}$ and supergradient in $\mathbf{u}$ can be computed in polynomial time. Important classes of functions which satisfy these assumptions are piecewise linear functions and the CV@R [13]. Practical applications of the framework adopt reformulations into PSD optimisation problems.

Interestingly, the problem will become non-convex in the general case if the equality constraint [13] replaces the linear matrix inequality constraint $\preceq$ in the ambiguity set. Enforcing the linear matrix inequality constraint from below is avoided for similar reasons. Furthermore, in some applications, i.e., portfolio selection, such a constraint is meaningless because it has no impact on the worst-case distribution, which intuitively assumes the largest variance possible.

A typical approach to set parameters $r_1$ and $r_2$ is based on analysing the confidence intervals for mean and covariance. The authors [13] derived them theoretically, assuming that the components of the probability distribution are independent and a ball bounded by some radius contains the support set of the true probability distribution. Moreover, if the conditions above hold, one could compute the probabilistic guarantee that the true probability distribution which generated sample data used to estimate the first and the second moment belongs to the ambiguity set, see [13][Corollary 5].

Researchers who followed the path paved by [13] developed more generic frameworks for defining moment-based ambiguity sets. Besides inequality constraints on the first and second moment, such frameworks allow for an arbitrary number of inequality constraints defined using certain classes of functions of random variables. For instance, researchers [64] consider ambiguity sets defined using inequality constraints on functions whose epigraphs are SOC-representable.

The definition below presents the generic structure of the ambiguity sets proposed by [64].

$$\mathcal{M} = \left\{ \mathbb{P} \in \mathcal{P}(\mathbb{R}^n) \;\middle|\; \begin{array}{l} \tilde{\mathbf{u}} \in \mathbb{R}^n \\ \mathbb{P}(\tilde{\mathbf{u}} \in \mathcal{U}) = 1 \\ \mathbb{E}_{\mathbb{P}}(\tilde{\mathbf{u}}) = \boldsymbol{\mu} \\ \mathbb{E}_{\mathbb{P}}(f_i(\tilde{\mathbf{u}})) \leq r_i \quad \forall i \in I \end{array} \right\}$$

The support set $\mathcal{U}$ is SOC-representable. For each $i \in I$, the function$f_i : \mathbb{R} \Rightarrow \mathbb{R}$ is defined on compact (a.k.a. closed and bounded) sets, and its epigraph is SOC-representable as well. Overall, the framework is flexible enough to model polyhedral and ellipsoidal uncertainty sets, upper bound on variance and piecewise linear functions, etc. [64]. On the other hand, generic semidefinite constraints, which are useful for modelling covariance, are not supported. Instead, as a workaround, researchers [64] suggest a simplistic approximation scheme based on the equivalence between a semidefinite constraint and a set of semi-infinite inequality constraints with a quadratic component.

$$\mathbb{E}_{\mathbb{P}}((\tilde{\mathbf{u}} - \boldsymbol{\mu})(\tilde{\mathbf{u}} - \boldsymbol{\mu})^\top) \preceq \boldsymbol{\Sigma} \iff \mathbb{E}_{\mathbb{P}}((\mathbf{v}^\top(\tilde{\mathbf{u}} - \boldsymbol{\mu}))^2) \leq \mathbf{v}^\top \boldsymbol{\Sigma} \mathbf{v} \quad \forall \mathbf{v} \in \mathbb{R}^n$$

As an immediate consequence, the semidefinite constraint could be approximated by a finite number of inequality constraints, i.e., $\mathbb{E}_{\mathbb{P}}((\mathbf{v}_j^\top(\tilde{\mathbf{u}} - \boldsymbol{\mu}))^2) \leq \mathbf{v}_j^\top \boldsymbol{\Sigma} \mathbf{v}_j \quad \forall j \in J$. Ambiguity sets with such modelling structures are called partial cross moment ambiguity sets. Unfortunately, there are no guidelines how to chose the sample vectors $\mathbf{v}$ [64].

If the ambiguity set is not empty, the worst-case expectation of a linear function over the ambiguity set can be reformulated into a classical RC, see [64][Theorem 1]. The proof relies on the duality theory for moment problems [71]. The sketch of the proof is provided in Appendix A.2. The classical RC can then be transformed into an equivalent SOC optimisation problem. Commercial solvers for this class of problems are more scalable than the solver technology developed for SDP [64]. Furthermore, motivated by performance considerations, the resultant optimisation problem could be an LP if the support set is a box and constraints on variance, i.e., $\mathbb{E}_{\mathbb{P}}((ax + b)^2) \leq v_1$, in the ambiguity set are replaced by upper bounds on absolute deviation, i.e., $\mathbb{E}_{\mathbb{P}}(|ax+b|) \leq v_2$.

Surprisingly, it is prudent to introduce auxiliary variables in the moment-based ambiguity set and relegate functions in the definition of constraints to the support set. By applying this technique (a.k.a. lifting) to the ambiguity set $\mathcal{M}$ and the support set $\mathcal{W}$ one obtains so-called lifted counterparts of the ambiguity set $\overline{\mathcal{M}}$ and the support set $\overline{\mathcal{W}}$.

$$\overline{\mathcal{M}} = \left\{ \overline{\mathbb{P}} \in \mathcal{P}(\mathbb{R}^{n_1} \times \mathbb{R}^{n_2}) \;\middle|\; \begin{array}{l} \tilde{\mathbf{u}}, \tilde{\mathbf{z}} \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \\ \mathbb{P}((\tilde{\mathbf{u}}, \tilde{\mathbf{z}}) \in \overline{\mathcal{W}}) = 1 \\ \mathbb{E}_{\mathbb{P}}(\tilde{\mathbf{u}}) = \boldsymbol{\mu} \\ \mathbb{E}_{\mathbb{P}}(f_i(\tilde{\mathbf{z}})) \leq r \end{array} \right\}$$

$$\overline{\mathcal{W}} = \{(\mathbf{u}, \mathbf{z}) \in \mathbb{R}^{n_1} \times \mathbb{R}^{n_2} \mid \mathbf{u} \in \mathcal{W}, f_i(\mathbf{u}) \leq \mathbf{z} \quad \forall i \in I\}$$

The lifted ambiguity set is equivalent to the initial ambiguity set [64][Proposition 1]. However, the introduction of auxiliary variables has important implications in multistage RO problems approximated using decision rules. The auxiliary variables could remove potential infeasibility introduced by the linear decision rule approximation, which may arise in some problem instances. Furthermore, the auxiliary variables generally should improve the objective value of the final solution [64].

SOC-representable moment-based ambiguity sets are specialisations of a more generic ambiguity set containing probability distributions that satisfy the equality constraint on the first moment and a group of probabilistic inequality constraints expressed using conic-representable sets. The definition of the generic ambiguity set provided below has been proposed by [14].

$$\mathcal{G} = \left\{ \mathbb{P} \in \mathcal{P}(\mathbb{R}^{n}) \;\middle|\; \begin{array}{l} \mathbb{E}_{\mathbb{P}}(\mathbf{A}\tilde{\mathbf{u}}) = \mathbf{b} \\ \mathbb{P}(\tilde{\mathbf{u}} \in \mathcal{C}_i) \in [\underline{p}_i, \overline{p}_i] \quad \forall i \in I \end{array} \right\}$$

The ambiguity set's definition contains a finite number of constraints. For each constraint $i$, the set $\mathcal{C}_i$ is defined over a proper cone $\mathcal{K}_i$ (a.k.a. closed, convex, pointed, with a non-empty interior [72]).

$$\mathcal{C}_i = \{\tilde{\mathbf{u}} \in \mathbb{R}^n \mid \mathbf{D}\mathbf{u} \preceq_{\mathcal{K}_i} \mathbf{c}\}$$

Each set $\mathcal{C}_i$ is associated with the interval $[\underline{p}_i, \overline{p}_i]$ which represents the confidence that the conic-representable expression for a given random variable following the true but unknown probability distribution satisfies the constraint.

For the given vector of decision variables $\mathbf{x}$, the worst-case expectation $\max_{\mathbb{P}\in\mathcal{G}} h(\mathbf{x}, \tilde{\mathbf{u}})$ can be reformulated as a finite conic program, see [14][Theorem 1], if the following assumptions hold. The function $h$ is a point-wise maximum of a finite number of functions that are convex and piecewise affine both in decision variables and in random variables. Furthermore, the set $\mathcal{C}_1$ should be the support set, i.e., $\overline{p}_1 = \underline{p}_1 = 1$. None of the sets $\mathcal{C}_i$ are empty if $\underline{p}_i < \overline{p}_i$. Finally, for each pair of sets $(\mathcal{C}_i, \mathcal{C}_j)$ either one set is fully contained in the other, or both sets are non-overlapping. If the last condition does not hold, then checking if the ambiguity set is not empty is an NP-hard problem in general, see [14][Theorem 2]. The authors demonstrated the flexibility of the framework by modelling the ambiguity set using robust statistics which are less sensitive to outliers, i.e., median, mean-absolute deviations and the Huber loss function are the counterparts of mean, standard deviation and variance, respectively. Furthermore, ambiguity sets defined using robust statistics could offer more attractive performance than the counterparts based on mean and covariance if the probability distribution deviates from normality or the data sample is small [14].

To conclude, the section presented the approaches to define ambiguity sets. Similarly to RO, the DRO models are either reformulated into finite-size mathematical programs and solved using a general-purpose MIP solver or optimised using the adversarial approach. Both solution methods are discussed in the following section. Furthermore, it presents two less common methods applicable to problems with a special structure or problem instances beyond the scalability limits of the MIP solver for the given formulation.

## 2.3 Solution Approaches

The development of RO and DRO led by the research community concentrated on solution frameworks that are theoretically and practically solvable. A RC of an uncertain MIP problem can be reformulated into a finite-size MIP formulation and solved using software available off-the-shelf. The solution approach adopted by commercial MIP solvers such as Gurobi or CPLEX is described in Section 2.3.1.

Even though the reformulation of the RC seems to be the default approach, it is not necessarily the most effective technique. Due to the lazy constraint generation feature, a MIP solver could solve the initial RC formulation in which constraints are defined for every element of the uncertainty set. Such a solution method is referred to as an adversarial approach [57], a cutting-set method with pessimising (worst-case) oracles [73], and a cutting-plane method [74,75]. Section 2.3.1 which explains it adopts the first naming convention.

Ultimately, Robust Mixed-Integer Optimisation is intractable in the general case. Consequently, if the bound derived from solving a continuous relaxation of the MIP formulation is weak, then exact methods may not remain a practical approach for the problem instances above a certain size, which is application dependent. For that reason, Oracle-based Approaches and CP with LS and Metaheuristics described in Sections 2.3.2 and 2.3.3, respectively, may offer a viable workaround for the cases in which MIP is less successful.

### 2.3.1 Mixed Integer Linear Programming

Mixed-Integer Linear Programming benefits from the mature solver technology developed initially for LP. Contemporarily, the state-of-the-art commercial solvers also support integer decision variables and selected SOC expressions in constraints and the objective function. Solvers also allow for lazy constraints generated on demand through callbacks provided by the customer. Such constraints can be added to the model at different stages of the optimisation, i.e., whenever a feasible solution reported by the solver violates them. The feature is handy for implementing a cutting plane method and for problem formulations that require a large number of constraints that are seldom violated.

Regardless of the company that implemented the solver, commercial-grade products approach MIP in the same way. The original MIP formulation is replaced by its relaxation in which integer and binary variables are substituted by continuous variables (a.k.a. integrality constraints are relaxed). Such a formulation can be solved in polynomial time by a simplex method or an interior point method. The simplex method can be used only if the relaxed formulation is an LP. Models which contain quadratic expressions are solved by an interior point method. It is the reason for the limited support for SOC expressions that may appear in the model. If possible, the solver runs both the simplex method and the interior point method in parallel and accepts the solution of the first method, which converged. Such a solution is a lower (upper) bound on the solution of the mixed-integer minimisation (maximisation) problem.

The optimal solution of the relaxed formulation may not be feasible to the original problem due to the violation of some integrality constraints (i.e., an integer variable assumes a fractional value in the relaxation). In response to this situation, the solver may generate additional inequality constraints, known as cuts, that are satisfied by all feasible solutions to the original mixed-integer formulation but may be violated after the integrality constraints are relaxed. Cuts may eliminate fractional values and improve the quality of the bound. Many families of cuts could be either problem-specific (a.k.a. valid inequalities) or general-purpose (i.e., Gomory Cuts, Mixed-Integer Rounding Cuts, etc.). They also vary in the computational effort required to find them. Ultimately, cuts are additional constraints added to the relaxed formulation. Therefore, they increase the size of the model and the time needed to solve it. For that reason, the solver has to control the trade-off between the time spent on generating cuts and the effective improvement they provide.

Some variables with relaxed integrality constraints may remain fractional despite cuts. The solver deals with such situations by considering two complementary scenarios in which a given variable $x$ set to some fractional value $v$ is either forced to assume value lower or equal to $\lfloor v \rfloor$ or greater or equal to $\lceil v \rceil$. This process is known as branching because both constraints are mutually exclusive and lead to different solutions. Subsequently, the solver solves the relaxed formulation again to obtain new solutions and update bounds for both scenarios. This solution approach is called branch-and-bound. Ultimately, it leads to finding a solution for the original problem or proving the infeasibility of the problem instance. The search process continues after a feasible solution has been identified. The optimisation is terminated after the relative gap between the bound obtained by solving the relaxed formulation to optimality and the best solution known (a.k.a. the incumbent solution) is less than some preconfigured threshold. If the gap limit is set to a very small value, then the final solution can be considered optimal. For that reason, MIP solvers are known as exact methods.

Theoretically, solving an arbitrary MIP is intractable. The practical effectiveness of the solution approach depends on the size of the branch-and-bound tree to explore. Understandably, the solver uses the bounds to guide the exploration. Branches for which the relaxed problem formulation becomes infeasible are abandoned (a.k.a. pruned). The same holds for branches whose bounds are worse than the incumbent solution. On the other hand, if the bounds computed using the relaxed formulation are weak, i.e., always below the cost of the incumbent solution for a minimisation problem, there is no pruning effectively, and the search process becomes an exhaustive enumeration. As a result, in such circumstances, only small problem instances can be solved.

To conclude, the size of the problem instance and the quality of the bound derived from the relaxed formulation are the key factors for deciding whether the mixed-integer approach is appropriate for solving the problem at hand.

### Adversarial Approach

The adversarial approach is an iterative method for solving a RC without applying the reformulation technique. Instead, the uncertainty set is replaced by a finite set of scenarios obtained by the selection of relevant elements from the uncertainty set. This section demonstrates how the adversarial approach works using an abstract problem with an uncertain inequality constraint modelled using the uncertainty set $\mathcal{U}$.

$$\begin{aligned} \min \mathbf{c}^\top \mathbf{x} \\ \text{s.t.:} \quad \mathbf{a}^\top \mathbf{x} \leq b \quad \forall \mathbf{a} \in \mathcal{U} \\ \mathbf{x} \in \mathcal{X} \end{aligned}$$

The uncertainty set is replaced by a finite set of samples $\mathcal{S}$. Initially, it contains only the nominal scenario. The adversarial approach proceeds by iteratively solving the formulation. A copy of the uncertain constraint is defined for each scenario.

$$\begin{aligned} \min \mathbf{c}^\top \mathbf{x} \\ \text{s.t.:} \quad \mathbf{a}^\top \mathbf{x} \leq b \quad \forall \mathbf{a} \in \mathcal{S} \\ \mathbf{x} \in \mathcal{X} \end{aligned}$$

Let $\mathbf{x}^*$ be the optimal solution computed in a given iteration. By construction, the solution is feasible for all scenarios contained in $\mathcal{S}$. However, the original uncertainty set may contain scenarios for which the solution is infeasible. Consequently, the adversarial method attempts to find at least one such scenario by solving the following optimisation problem to extend the set $\mathcal{S}$ for the next iteration.

$$\beta(\mathbf{x}^*) = \max \mathbf{x}^{*\top} \mathbf{a} - b \tag{2.35}$$

$$\mathbf{a} \in \mathcal{U} \tag{2.36}$$

The $\mathbf{x}^*$ is a part of the input, and the tractability of the problem depends on the uncertainty set. Section 2.2.2 explained uncertainty sets that lead to tractable reformulations. Let $\mathbf{a}^*$ be the optimal solution of the problem. If the optimal objective value is non-negative, then the solution $\mathbf{x}^*$ satisfies the uncertain constraint for the entire uncertainty set $\mathcal{U}$. Otherwise, if the objective is negative, the solution $\mathbf{x}^*$ violates the uncertain constraint for the scenario $\mathbf{a}^*$. The scenario is added to the set $\mathcal{S}$, and a new iteration is started. Eventually, the solution method terminates, which has been proven by [73].

The effectiveness of the adversarial approach compared to the reformulation with ellipsoidal and polyhedral uncertainty sets was analysed by [75]. The authors concluded that for Robust Linear Optimization with ellipsoidal uncertainty sets, reformulations are faster to solve. Nonetheless, if polyhedral uncertainty sets are used or mixed-integer optimisation is considered, then it is not clear which method is more efficient [75]. Ultimately, researchers demonstrated that running both approaches in parallel and accepting the solution of the first method that converged could reduce the computational time by half [75]. The researchers [75] obtained those results using very efficient algorithms for finding the worst-case scenarios that violate a given uncertain constraint. In particular, the authors used a closed-form solution for the ellipsoidal uncertainty set and an algorithm that had $O(nlog(n))$ complexity for the budgeted uncertainty set [75]. Their work should be relevant to practitioners who consider adopting the adversarial approach, as the paper discusses the benefits and limitations of different implementations. An additional merit of the paper [75] is a presentation of a rigorous methodology to derive statistically significant results comparing the computational time of the adversarial approach and the RC reformulation.

### 2.3.2 Oracle-Based Approach

The oracle-based approach attempts to exploit efficient algorithms developed for deterministic DO problems in Robust Mixed-Integer Optimization. A notable example of a solution approach which belongs to this category is solving Combinatorial Optimisation problems with an uncertain cost function modelled using the budgeted uncertainty set. An optimal robust solution for such a problem can be found by solving $n+1$ deterministic problem instances, where $n$ is the number of uncertain coefficients in the cost function [12]. In the similar spirit, given an exact oracle for a deterministic combinatorial optimisation problem, the problem with an uncertain cost function modelled using uncorrelated ellipsoidal uncertainty set admits an approximation scheme that solves the optimisation problem to the desired accuracy in a number of calls to the oracle that is polynomially bounded by the size of the instance and the reciprocal of the accuracy (a.k.a. an oracle FPTAS) [76].

### 2.3.3 Constraint Programming with Local Search and Metaheuristics

Constraint Programming is a solution methodology developed for deterministic DO problems in which all decision variables are integers. Compared to MIP, which has a strong connection to LP, CP adopts a radically different approach. It distinguishes common modelling structures, known as global constraints, which are apparent in many DO problems and employs dedicated algorithms to propagate bounds for integer variables more efficiently than an equivalent representation of the global constraint using a system of inequalities. Notable examples of global constraints include sub-tour elimination, packing, assignment etc. Overall, there are more than 400 global constraints [77], and support for them varies between solvers. The ability to recognise and apply global constraints facilitates building concise formulations which can be solved effectively.

Besides global constraints, CP solvers [78,79] support an interval data type designed to model temporal activities. An interval can be associated with time windows when the activity it represents should commence and a minimum duration it lasts. Naturally, the data type is supported by additional constraints, i.e., defining precedence relationships or enforcing that intervals must not overlap.

Constraint Programming provide means to compute domains for decision variables. However, it is the responsibility of the decision-maker to specify the search strategy the solver should follow, i.e., to decide which variable to assign first and which value to choose, i.e., select an unassigned variable with the smallest domain and assign a random value.

Contrary to the MIP solver technology dominated by commercial products, CP solvers that rank first in benchmarks [80] are available open-source. Consequently, researchers, rather than becoming end-users focusing on the problem formulation, could actively participate in developing the solver technology, i.e., by implementing new global constraints or hybridising CP with LS and metaheuristics. The combination of optimisation techniques became the state-of-the-art approach for solving large instances of the VRP which are beyond capabilities of exact methods [81].

Local Search is an optimisation technique that works by generating a pool of solutions by applying some dedicated transformation algorithms to the incumbent solution. A solution from the pool is accepted as the new incumbent if it is feasible and has a better objective value than the previous incumbent solution. Understandably, the transformation algorithms are application specific and depend on the encoding of a solution. In general, they tend to be simple operations, i.e., swapping values of two variables, because LS is effective if it generates feasible solutions quickly. For example LS operators for the VRP see [82]. Overall, due to its speed and simplicity, LS is a common component of more advanced methods. However, the simplicity of LS is also the source of its limitations. The technique requires initialisation by a feasible solution. Understandably, it could be a bottleneck for problem instances that are at the edge of infeasibility. Moreover, by accepting solutions only if they lead to an improvement, LS eventually converges to the optimal solution in the local basin of attraction induced by LS operators. Consequently, to avoid premature convergence, LS has to be combined with metaheuristics.

Metaheuristics are best effort solution techniques developed to search for a globally optimal solution. They typically include a randomisation component and tentatively accept solutions that do not lead to an improvement. Consequently, metaheuristics are capable of moving between local basins of attraction. However, the navigation through the basins of attractions by metaheuristics is devoid of any systematic plan. At most, some ad-hoc mechanisms preventing cycles are implemented, i.e., the rejection list in Tabu Search (TS) [83]. For that reason, metaheuristics lack clearly defined stopping criteria. A notable example is a Multi-Start LS which continuously applies a randomised algorithm to generate an initial guess and then applies LS. Stopping criteria could be the limit on the number of restarts or the limit on the consecutive number of iterations since the last improvement. For a thorough treatment of metaheuristics in the VRP see [82].

Ultimately, there is no widely accepted solution methodology for solving uncertain

DO problems using CP combined with LS and metaheuristics. Contrary to RO where modelling uncertainty affects the problem formulation and the solution process, CP with LS relies on a deterministic formulation of the problem. The robustness of the solution is assessed afterwards [84]. The approaches considered in the literature are dominated by the scenario-based representation of uncertainty [85]. The cost of a given solution is evaluated in a multi-objective way for each scenario independently. However, to avoid dealing with a pool of Pareto-non-dominated solutions, the cost vectors are sorted in descending order and compared element-wise [85]. Such an approach is also known as lexicographic min-max robustness criterion [86]. It is closely related to the concept of Lorenz-dominance studied in the context of the optimisation over scenarios by [87].

## Concluding Remarks

The chapter explained core concepts related to the formulation of optimisation problems affected by uncertainty. It covered widely-acknowledged methodologies and discussed available solution techniques. The presented material should provide the theoretical foundation to develop a scalable solution methodology for large real-world applications of vehicle routing and scheduling problems in the following chapters.

# Chapter 3

# Methodology: Satellite Quantum Key Distribution

## 3.1 Introduction

This chapter develops a methodology for scheduling space to ground data transfers from a spacecraft deployed into the LEO and discusses in detail an example application to the modelling of the SatQKD problem. Lower Earth Orbit is a standard configuration for an Earth Observation Satellite (EOS) and a small, low-budget experimental satellite (a.k.a. CubeSat). Spacecrafts which are orbiting LEO circle around the globe several times per day. The precise number of revolutions depends on the orbital parameters of the spacecraft. Their mission objectives are to collect or to generate information, and to downlink available data to the network of ground stations for post-processing. The formulation of the objective function for the optimisation problem depends on the application. Notable examples are downloading images collected during Earth observations [88, 89] and delivering cryptographic keys in SatQKD [90]. However, the content transferred is not relevant from the perspective of scheduling communication.

The standard objective function for downloading Earth observations data is to maximise the volume of data successfully downloaded to the network of ground stations divided by the amount of data acquired during the mission (a.k.a. the Maximum Percentage Data Transferred (MaxPDT)) [91, 92]. An arbitrary ground station can receive the data transfer, and there are no negative consequences for directing downloads only to a subset of ground stations.

On the other hand, the ground station that receives the data transfers becomes critical in the SatQKD in which a satellite generates private keys and sends them as confidential information to the given ground station for exclusive use. The keys will be employed later to encrypt internal communication with other ground stations. The satellite should distribute keys according to the importance of ground stations in the network. Scenarios in which some ground stations stockpile all keys, and the remaining ground stations have no keys for securing communication should be avoided.

The space to ground communication considered in this chapter is performed using optical devices. The satellite is equipped with a laser that generates a beam pointed at the telescope of the ground station, which receives the signal. Such a setup has numerous benefits over traditional radio communication used for space to ground communication over the last decades. Optical communication systems are suitable for miniaturisation, require less power for transmission, and allow for an order of magnitude faster data transfer [93]. Moreover, since the data transfer is received in a specific geographical location, there is no need for bandwidth licensing [93]. An additional advantage is an opportunity for enhanced security by applying Quantum Communication Protocols [94], i.e., the receiver can detect if the message has been eavesdropped on by a third party while passing through the atmospheric channel.

Despite notable benefits, a severe limitation of optical communication is its sensitivity to weather conditions and atmospheric perturbations which are due to the physical properties of light. In particular, if a laser beam encounters water particles, the transmission rate is adversely affected due to diffraction [95]. For that reason, scheduling space to ground data transfers for optical communication that can be executed with minimal disruption requires accounting for cloud cover conditions. They are available in advance in the form of weather forecasts which are affected by uncertainty. The underlying circumstances motivate the development of scheduling models for handling cloud cover uncertainty, which is the main contribution of this chapter.

The optimisation models are formulated using the SO, RO and DRO methodologies introduced in Chapter 2. Notable reasons for choosing such an approach over a single solution methodology, i.e., DRO, the most recent one, is the opportunity to compare various formulations. The SatQKD is an emerging application, and all of the presented models are new for the given application. It is not self-evident which formulation leads to the most attractive solutions and whether such solutions can be found consistently for different problem instances. That subject will be explored in the subsequent chapter focused on the application part. Presenting different solution methodologies also has educational merit. A novice reader is provided with an opportunity to see examples derived from the same deterministic model. They demonstrate how uncertainty is handled using various solution methodologies, what information about uncertain parameters is required, and how complex the resultant models become.

### Structure of the Chapter

Section 3.2 provides the motivation for selecting the SatQKD as the featured optimisation problem for this chapter and outlines the key modelling assumptions. Section 3.3 introduces common symbols and modelling structures used in the formulations. Section 3.4 presents the MIP model for the deterministic SatQKD and contains a formal proof of the problem complexity. Section 3.5 presents an alternative CP formulation for the deterministic variant of the SatQKD problem. Section 3.6 develops a methodology for representing uncertainty in cloud cover predictions. It contains a suite of alternative models formulated using RO, DRO and SO paradigms. Some concluding remarks summarise the chapter.

The focus on the SatQKD is without loss of generality. The models presented are transferrable to a broad class of problems involving scheduling a spacecraft deployed into a fixed orbit that does not change throughout the scheduling horizon. Moreover, the formulation of the deterministic model should employ time discretisation. Ultimately, it is the most common modelling approach used when satellites' orbits are fixed, see, e.g. [20, 96, 97]. An example problem that belongs to this group is the MaxPDT problem with a fixed network of ground stations studied by [91, 92]. Contrary to the SatQKD problem, the objective in the MaxPDT problem is the maximisation of the total volume of transferred data regardless of the target ground station. Consequently, deterministic formulations of both problems adopt different definitions of constraints and objective function. Nonetheless, obtaining a suitable MaxPDT formulation with the treatment of uncertainty requires applying the analogous sequence of transformations which is demonstrated step-by-step on the SatQKD problem.

## 3.2 Featured Optimisation Problem

Henceforth, the focus is on the SatQKD as the target application. There are several reasons for making it the featured optimisation problem in the thesis.

1. It is a new application that emerged from the interest in enhanced security offered by Quantum Communication Protocols [98–100]. Research in this area has been dominated by the Physics community, whose primary focus was on the scientific demonstrations [90, 101, 102] and the space mission design [103–105]. However, the problem has not been studied from the optimisation perspective, e.g., the complexity of the optimisation problem has not been formally analysed. Furthermore, it is not immediately apparent how the problem should be formulated and solved.

2. The solution approach adopted for the SatQKD will be vastly different from the methodology developed for the VRP optimisation. Nonetheless, the computational results presented in Chapter 4 will strongly corroborate the rationale for the choice made.

3. There are only a handful of articles on scheduling satellite operations under uncertain weather conditions [88,89,106], which are summarised in Chapter 4. Overall, their authors used well-known RO and SO frameworks. In contrast, computational approaches adopted in the thesis are based on more recent DRO and the riskiness index optimisation. Furthermore, a new uncertainty set that has the desirable property of preserving spatial and temporal correlations between the parameters affected by uncertainty is proposed.

4. To the best knowledge of the author, there is no methodology that supports the use of official weather forecasts in scheduling satellite communications. It is surprising because a successful space-to-ground optical communication requires favourable weather conditions while passing through the atmospheric channel due to the physical properties of light. For that reason, the author believes that several models for handling cloud cover uncertainty and demonstrating their utility in the next chapter will be interesting for practitioners.

### Modelling Assumptions

The following assumptions are adopted to formulate the optimisation problem.

1. For the simplicity of the exposition, the communications system consist of one satellite and a fixed set of ground stations. These assumptions do not affect the application of the methodology and the solution method. Overall, the support for a constellation of satellites can be implemented by adding synchronisation constraints to enforce that a given ground station can receive only one data transfer at a given time [97]. Such constraints can be defined using linear inequalities with binary variables and are not affected by uncertainty. In a similar vein, a decision-maker could provide a broad set of candidate ground stations and charge an operational cost for each facility used.

2. The satellite can perform a data transfer to at most one ground station at a time. Again, the purpose of the assumption is to simplify the exposition. If the spacecraft's design specification includes multiple transmitters, they can be added to the formulation to support a constellation of satellites.

3. After a data transfer to a given ground station is completed, a subsequent data transfer to another ground station can be commenced after a fixed delay. The setup time is spent on the calibration of the optical apparatus and the initialisation of the connection to the next ground station.

4. Data transfers to a given ground station can be conducted only during predefined time intervals. They represent periods when the elevation angle between the

satellite and the ground station is within the permissible range, and the spacecraft remains in the complete Earth shadow (a.k.a. umbra). Intuitively, the lower the elevation angle, the longer distance the laser beam needs to travel in the atmosphere. Understandably, below a certain threshold, which depends on the technical specification of the communication equipment, errors are considered too high for performing a successful data transfer. The requirement for the spacecraft to remain in the shadow during the communication is to avoid the distortion of the laser beam caused by the sunlit.

5. The satellite does not perform any manoeuvrers that may change the parameters of the orbit throughout the scheduling horizon. The assumption allows for a pre-calculation of the time windows for communication with a given ground station and the position of the satellite on the sky to include them in the problem definition. It is a common assumption in the literature focused on scheduling satellite data transfers, see, e.g. [96, 97].

6. The transfer rate is variable and depends on the current position of the satellite on the sky and cloud cover conditions. The transfer rate is decreased proportionally to the percentage of the sky occupied by the clouds.

7. The satellite sends data at the transfer rate a ground station can receive. For simplicity, each ground station has the same communication capabilities.

8. Every ground station is assigned a weight that denotes the importance of the node in the communications system. The spacecraft should distribute the cryptographic keys proportionally to that weight.

9. A cryptographic key is 256 bit of randomness. The keys are used by the ground stations to establish secure duplex connections. A duplex connection requires each endpoint to contribute one key. Then, the key is removed after the connection is closed due to the security requirement preventing the reuse of keys.

10. Ground stations buffer keys for later use. The keys acquired by a given ground station during a scheduling horizon become available for use in the next scheduling

horizon. Consequently, the objective function can be evaluated using the number of keys each ground stations holds at the end of the scheduling horizon.

## 3.3 Shared Symbols and Conventions

The SatQKD problem formulations developed in this chapter will be built using the following symbols and modelling structures.

Let $N \coloneqq \{1, \ldots, n\}$ be the set of ground stations. Besides regular ground stations from the problem domain, there is an auxiliary ground station 0. It will be observed by the satellite during the idle period between consecutive data transfers to regular ground stations. The set of all ground stations, including the auxiliary one, is denoted using the symbol $\bar{N}$.

Two indexing schemes illustrated in Figure 3.1 will be used to discretise the scheduling horizon in the deterministic formulation. Both indexing schemes map some selected time intervals from the scheduling horizon to the elements of an ordered set.

The decision to use the time discretisation approach was made after a careful analysis. Notably, the literature indicates that the models with the time discretisation for scheduling satellite operations are easily solvable for realistic problem instances [96,97]. Such models require a binary variable for each combination of a time interval, a ground station and a spacecraft which could be prohibitive for long scheduling horizons or small discretisation steps. However, it was not an issue for problem instances studied in the thesis.

The indexing scheme $T$ partitions the entire scheduling horizon into equally sized time intervals. Their duration depends on the frequency of releasing new weather forecasts. Weather conditions within the time interval are considered constant. For every time interval $t \in T$, let define an indexing scheme $\Pi_t$. It covers some fragments of the time interval $t$ in which the satellite can communicate with at least one ground station. The indexing scheme $\Pi_t$ has a high resolution with the discretisation step in order of seconds. It corresponds to the frequency of updating the satellite's position in the sky, which impacts the transfer rate. Let assume that the throughput of the connection to a given ground station is constant within the time interval $\pi \in \Pi_t$.

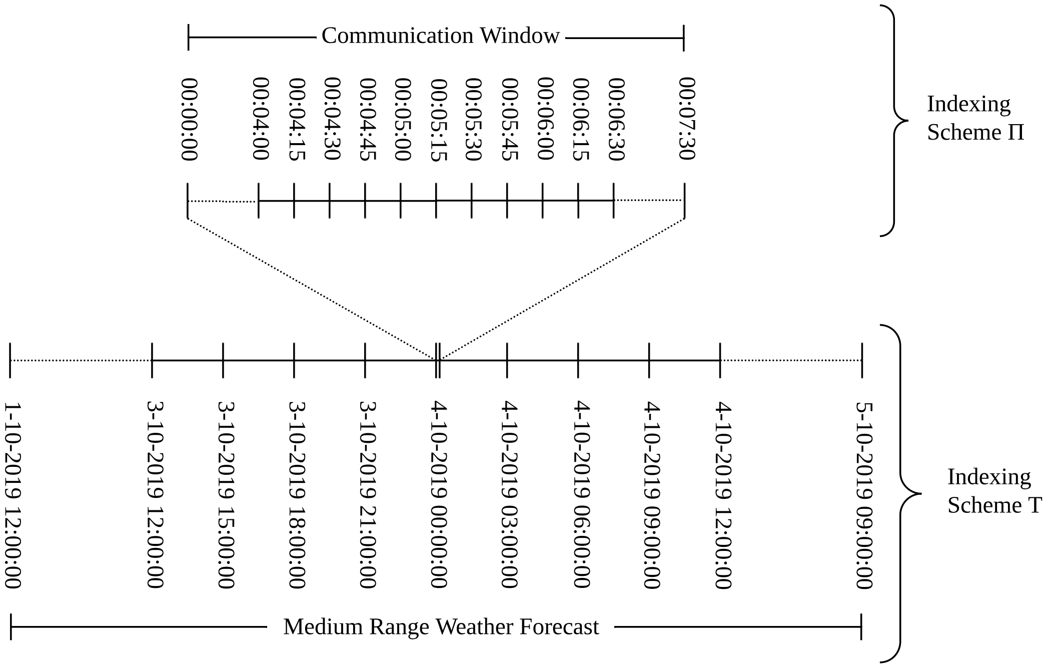


Figure 3.1: Indexing schemes used for the time discretisation. The indexing scheme Π is applied to communication windows between the satellite and ground stations. The discretisation step is small (i.e., in a range of seconds) to accurately reflect the spacecraft's position in the sky. The indexing scheme does not map time intervals in which no communication with any ground station is possible. Contrary to the indexing scheme Π, the time discretisation using the indexing scheme $T$ covers the entire scheduling horizon. The discretisation step is measured in hours and corresponds to the frequency of announcing official weather forecasts.

Furthermore, to simplify the notation, the high resolution indexing scheme is aggregated $\Pi := \bigcup_{i=1}^{T} \Pi_i$.

Given the set of the time intervals $\Pi$, let define the following filtering operations. The subset $\Pi^n$ contains the time intervals in which the satellite can communicate with the ground station $n \in N$. Moreover, the subset of intervals can be further restricted to those which are contained in a given time period $t \in T$, i.e., $\Pi^{nt} := \Pi^n \cap \Pi_t$.

The vector $\mathbf{c} \in [0, 1]^{|N| \times |T|}$ represents cloud cover for a given pair of a ground station $n \in N$ and a time interval $t \in T$. In the deterministic model, the cloud cover vector is a parameter provided as the input. On the other hand, the models of uncertainty built using RO and DRO frameworks represent the cloud cover vector as a variable. The vector of constants $\mathbf{k} \in {\mathbb{R}_{\geq 0}}^{|N| \times |\Pi|}$ stores the number of cryptographic keys that can be transferred to a given ground station $n \in N$ at a given time interval $\pi \in \Pi$ assuming clear sky (a.k.a. cloud-free line of sight). In the presence of clouds, the transfer rate is adjusted proportionally to the area of visible sky, i.e., the effective number of keys delivered to the ground station $n \in N$ during the period $\pi \in \Pi$ is assumed to be $(1 - c_{nt})k_{n\pi}$. The same formula was applied to account for cloud cover conditions in the optimisation of the ground station infrastructure for optical free-space communication [92] and in the analysis of solar irradiance [107]. The importance of the ground station $n$ in the communication system is proportional to the constant $\omega_n$. Finally, the parameter $\beta_n$ contains the number of spare keys a ground stations holds at the beginning of the scheduling horizon.

### 3.3.1 Feasible Schedule

A schedule of data transfers is encoded using a vector of binary variables $\mathbf{x} \in \mathbb{B}^{|\bar{N}| \times |\Pi|}$ which denote whether a given ground station is receiving a data transfer at the selected time interval.

A valid schedule must obey a group of feasibility constraints that are included in every formulation discussed in this chapter. For brevity, let define the set of feasible solutions $\mathcal{X}$ and use the compact notation $\mathbf{x} \in \mathcal{X}$ instead of declaring the constraints individually in every formulation.

$$\mathcal{X} = \left\{ \mathbf{x} \in \mathbb{B}^{|\bar{N}| \times |\Pi|} \;\middle|\; \begin{array}{l} \sum_{n \in \bar{N}} x_{n\pi} \leq 1 \quad \forall \pi \in \Pi \\ x_{n\pi} \leq x_{n\pi-1} + x_{0\pi-1} \quad \forall \pi \in \Pi \setminus \{1\}, \quad \forall n \in N \end{array} \right\}$$

The constraint $\sum_{n \in \bar{N}} x_{n\pi} \leq 1$ states that for every time interval $\pi \in \Pi$ the satellite is either transferring data to at most one ground station or remains idle observing the auxiliary ground station. The constraint $x_{n\pi} \leq x_{n\pi-1} + x_{0\pi-1}$ for every regular ground station $n \in N$ and every pair of subsequent time intervals $(\pi - 1, \pi)$ enforces that the satellite either continues a data transfer to the same ground station or is undergoing the transition phase preceding a data transfer to a different ground station. Consequently, the setup time before commencing a data transfer to a new ground station is respected.

### 3.3.2 Distribution According to the Importance

The satellite is supposed to distribute keys according to the importance weights assigned to ground stations. Depending on the formulation, this requirement will be defined either in the objective function or as a constraint. Let introduce the following expression, which will be used in models presented later.

$$\Lambda(\mathbf{x}, \mathbf{c}) = \min_{n \in N} \left\{ \frac{1}{\omega_n} \left( \beta_n + \sum_{t \in T} \sum_{\pi \in \Pi^{nt}} (1 - c_{nt})\, k_{n\pi} x_{n\pi} \right) \right\} = \min_{n \in N} \{\lambda_n(\mathbf{x}, \mathbf{c})\}$$

Given a feasible schedule $\mathbf{x}$ and cloud cover conditions $\mathbf{c}$, the expression $\Lambda(\mathbf{x}, \mathbf{c})$ returns a non-negative real number which is denoted using the symbol $\lambda$. Intuitively, for each ground station $n \in N$, the expression $\omega_n \lambda$ is a lower bound on the total number of keys the ground station $n$ would store at the end of the scheduling horizon if no keys were consumed in the meantime. The growth of $\lambda$ is equivalent to an increase in the number of keys for all ground stations proportionally to their importance weights. For that reason, henceforth, $\lambda$ is called a scaling factor. Due to the properties mentioned above, maximisation of $\Lambda(\mathbf{x}, \mathbf{c})$ prevents stockpiling of keys by a subset of ground stations. Therefore, it is a suitable objective function for the SatQKD problem.

## 3.4 Deterministic Model

The deterministic formulation considers cloud cover predictions $\mathbf{c}$ as exact values and provides no treatment of uncertainty. Supported by the modelling structures introduced in the previous section, the SatQKD problem is formulated as follows.

$$\max_{\mathbf{x} \in \mathcal{X}} \Lambda(\mathbf{x}, \mathbf{c}) \tag{3.1}$$

The deterministic model presented above was first proposed in [20]. Besides the reference number, Formulation 3.1 is called Deterministic Single Forecast Model (DSF) in the sequel to facilitate cross-referencing and comparisons with alternative models.

A schedule with no data transfers is a valid solution for the DSF problem. The objective value for that schedule, i.e., $\min_{n \in N}\{\beta_n/\omega_n\}$, is a lower bound for the objective of the optimal solution. On the other hand, an upper bound can be computed for the relaxed problem in which the satellite performs data transfers to all available ground stations simultaneously, i.e., $\min_{n \in N} \lambda_n(\mathbf{1}, \mathbf{c})$, where $\mathbf{1}$ is a vector of all-ones.

The exposition of the deterministic problem is concluded by the discussion of its computational complexity stated in Theorem 1.

**Theorem 1.** *The deterministic SatQKD problem is NP-hard.*

*Proof.* Suppose the original SatQKD problem is restricted to the variant with two ground stations of equal importance. The ground stations can receive the same number of keys for every time interval. Furthermore, setup times are ignored and no cloud cover is assumed. The specialisation of the problem formulated to simplify the proof belongs to the same complexity class as the initial problem.

Let consider a reduction to the problem of scheduling jobs on parallel processors to maximise the earliest completion time [108]. Following the notation established in the scheduling community, the problem is known as $P||C_{min}$. Its applications include election manipulation [109, 110] and scheduling maintenance operations to maximise the lifetime of a system [111]. The restriction of the problem with two machines is

NP-hard which can be proven by reduction to the partition problem [108].

Formally, let $\Pi$ be the set of the time intervals in which the ground stations $N$ can receive data transfers. The number of keys possible to deliver in the time interval $\pi$ is positive and denoted $k_{\pi}$. The vector of binary decision variables $\mathbf{x}$ indicates whether the satellite sends keys to a given ground station during some time intervals. Finally, let $\lambda$ be the scaling factor which is subject to optimisation.

Let $M$ be the set of processors and $J$ the set of jobs. The processing time stored in the vector $\mathbf{t}$ is positive and independent of the processor. The vector of binary decision variables $\mathbf{y}$ indicates whether a given processor is scheduled to execute a given job. The time to complete all jobs on the faster processor is denoted by $\gamma$.

Each processor $m \in M$ is uniquely mapped to some ground station $n \in N$. The analogous relation is defined between jobs $J$ and time intervals $\Pi$. Taken together, both mappings indicate the equivalent decision variables between two problems $y_{m,j} = x_{n,\pi}$ for each pair $(\pi, j)$ and $(n, m)$. In a similar way, the processing time of some job $j$ equals the number of keys which can be sent during the corresponding time interval $\pi$, i.e. $t_j = k_{\pi}$.

To complete the proof, let consider the optimal solution of the SatQKD problem. By construction, the optimal value of the scaling factor $\lambda^*$ is equivalent to the maximum time the first processor completes the execution of its jobs. The allocation of the jobs to processors can be retrieved from decision variables $\mathbf{x}$ using mappings ground stations to processors and time intervals to jobs. The proof in the other direction follows the same line of argument. □

Ultimately, one could prove that SatQKD is strongly NP-hard by reduction from the 3-Partition problem [112]. The result demonstrates that a FPTAS (i.e., an approximation scheme polynomial both in the input size and in the reciprocal of the approximation accuracy $\epsilon$) does not exist for the SatQKD problem. Therefore, solving the formulation using a MIP solver is justified. An alternative proof developed by the author was presented in the thesis to highlight the similarity between SatQKD and the scheduling problem $P||C_{min}$.

The deterministic model seems sufficient for the analysis of optimal schedules computed using accurate weather information, which could serve as an indicator of the best possible performance a given specification of the communication system could achieve. Supported by historical weather observations, the deterministic model in [20] was used to evaluate how many keys one satellite could deliver to the predefined network of ground stations weekly and to find the parameters of the satellite orbit that maximise the number of keys possible to distribute.

The utility of the deterministic model for scheduling data transfers "in the future" for real-world operations depends on the ability to predict cloud cover conditions accurately. Understandably, the results vary between the geographical locations considered and the providers of the weather forecasts. The analysis of medium-range weather forecasts published by [113] (i.e., forecasting horizon of the next five days, weather conditions updated every 3 hours) for the selected cities in the UK suggests that prediction errors for cloud cover grow considerably with the length of the forecasting horizon.

Figure 3.2 illustrates an example cloud cover forecast for Glasgow compared to real cloud cover conditions observed later. In the presented example, the cloud cover predictions maintain a reasonable accuracy only for a short time in the future. The standard deviation of the cloud cover prediction error for the next 120 hours exceeds 40%.

Conceivably, schedules built based on inaccurate cloud cover predictions are likely to exhibit unsatisfactory performance in the real world. A pragmatic approach to improve the accuracy of weather forecasts is to create a schedule using a multistage optimisation with a rolling (a.k.a., folding) horizon. During the execution of the schedule, one could observe real weather conditions and collect updated weather forecasts for the remaining period. At every stage, a new schedule is computed for the future using the updated and hence a more accurate weather forecast. Furthermore, cloud cover conditions observed already enable precise evaluation of the scheduling decisions regarding the past. The variant of Formulation 3.1 with the rolling horizon is referred to in the sequel as the Deterministic Multistage Model with Folding Horizon (DRH).

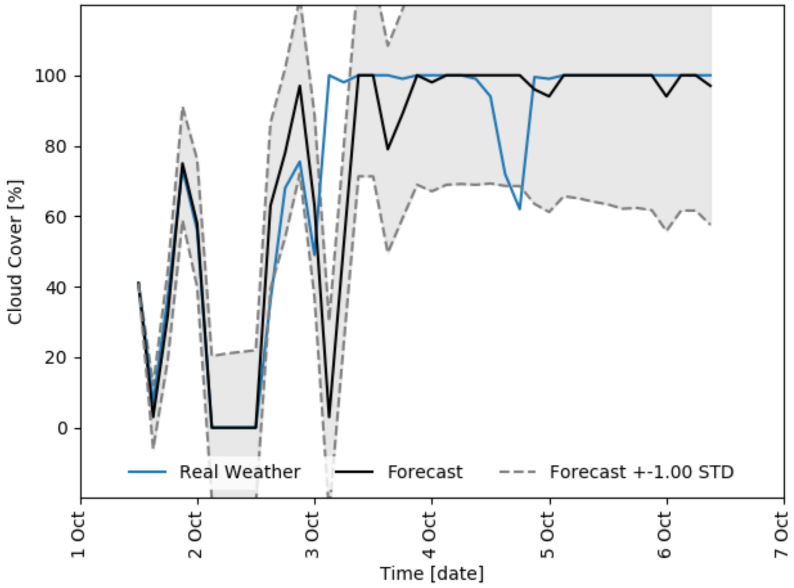


Figure 3.2: Medium-range cloud cover forecast for Glasgow announced on the 1st of October 2019 at noon. The marked area between the dashed lines denotes the range of one standard deviation for the prediction error. It was estimated using a sample of marginal differences between the cloud cover forecast and the conditions observed in the real world between June 2019 and February 2020.

## 3.5 Constraint Programming Model

Endpoints of the time intervals in Formulation 3.1 cannot assume arbitrary values but are restricted to multiples of the discretisation step that was adopted to generate the indexing scheme $\Pi$. One may consider the advantages of using a more flexible formulation that allows endpoints of the time intervals to assume arbitrary values. Arguably, a sensible choice of a paradigm for such a formulation is CP due to the native support and dedicated algorithms for constraints involving time intervals.

This section presents a CP model for the SatQKD problem in which endpoints of the time intervals could assume arbitrary values. The details of the formulations follow the symbols introduced below.

Recall $N$ is the set of ground stations and $\lambda_n$ is the scaling factor for the ground station $n$. Let $\Psi$ be the set of time intervals. Every interval $\psi \in \Psi$ is linked with two variables $\psi_b$ and $\psi_e$. They denote the beginning and the end of the time interval, respectively. Their value range is $[0, \tau]$, which corresponds to the scheduling horizon. The set of time intervals $\Psi$ is partitioned into $|N|$ subsets, i.e., $\Psi_1, \ldots, \Psi_{|N|}$. Every subset contains the time intervals allotted to observations of a particular ground station. The function $\kappa_n(\cdot)$ given some interval $\psi \in \Psi_n$, returns the number of keys delivered to the ground station $n$. The function hides the complexity of applying the reduction in number of keys due to the cloud cover conditions observed during communication. Moreover, the function adheres to the requirement that the satellite has to wait before commencing a new data transfer for the period $\underline{l}$.

$$\max \min_{n \in N} \lambda_n \tag{3.2}$$

$$\lambda_n \leq \frac{1}{\omega_n} \sum_{\psi \in \Psi_n} \kappa_n(\psi) \qquad \forall n \in N \tag{3.3}$$

$$\text{all\_disjunctive}\,(\Psi) \tag{3.4}$$

$$\psi_b \leq \psi_e \qquad \forall \psi \in \Psi \tag{3.5}$$

$$\psi_e - \psi_b > \underline{l} \ \vee \ \psi_b = \tau \qquad \forall \psi \in \Psi \tag{3.6}$$

$$\psi_b, \psi_e \in [0, \tau] \qquad \forall \psi \in \Psi \tag{3.7}$$

$$\boldsymbol{\lambda} \in \mathbb{R}_{\geq 0}^{|N|} \tag{3.8}$$

Similarly to Formulation 3.1, Objective Function 3.2 maximises the minimum scaling factor for all ground stations. Constraint 3.3 bounds the scaling factor from above by the total number of keys transferred to the ground station within the scheduling horizon divided by the weight of the ground station. Constraint 3.4 utilises a global constraint available in CP enforcing the intervals to span across non-overlapping time segments. The time intervals must be well defined, i.e., the start of the interval must happen before it ends, due to Constraint 3.5. Furthermore, every non-empty time interval must not be shorter than $\underline{l}$ according to Constraint 3.6. Some intervals in a given subset $\Psi_n$ may not be allocated to the observation of the ground station $n$. They remain empty and are shifted to the right-hand side of the scheduling horizon $[0, \tau]$ to break the symmetry. Finally, Constraints 3.7-3.8 define domains of decision variables.

The CP formulation presented above was implemented for testing and benchmarking purposes. The preliminary experiments consistently indicated that the performance of the CP formulation was significantly worse compared to Formulation 3.1 regarding the objective value of the final solution and computation time. Consequently, the CP formulation was discarded, and the focus was shifted towards uncertainty modelling in the formulation with time discretisation.

## 3.6 Modelling Cloud Cover Uncertainty

Besides the folding horizon approach, the improvement of the real-world schedules' performance can be accomplished by including the model of cloud cover uncertainty in the problem formulations. This section outlines several such formulations developed using RO, DRO and SO frameworks.

### 3.6.1 Robust Optimisation

#### Box Uncertainty

The range of possible values a parameter affected by uncertainty could assume is essential for building a formulation in the RO framework. Historically, this description of uncertainty was pioneered by [59] and became known either as the interval or box uncertainty. In the model [59], every parameter affected by uncertainty could assume an arbitrary value within the prescribed range independently.

Let introduce the vectors of constants $\bar{\mathbf{c}}, \underline{\mathbf{c}} \in [0,1]^{|N|\times|T|}$ which define respectively the upper and the lower bound for cloud cover conditions for every pair of the ground station $n \in N$ and the time interval $t \in T$. The box uncertainty set $\mathcal{B}$ is defined using vectors.

$$\mathcal{B} = \left\{ \mathbf{c} \in [0,1]^{|N|\times|T|} \mid \underline{c}_{nt} \leq c_{nt} \leq \bar{c}_{nt}, \forall n \in N, \forall t \in T \right\} \tag{3.9}$$

The uncertainty set contains all realisations of uncertainty considered in the following formulation.

$$\max_{\mathbf{x}\in\mathcal{X}} \min_{\mathbf{c}\in\mathcal{B}} \ \Lambda(\mathbf{x},\mathbf{c}) \tag{3.10}$$

Due to non-negativity of variables $\mathbf{x}$, the worst case realisation of uncertainty for the expression $\Lambda(\mathbf{x},\mathbf{c}) = \min_{n\in N}\left\{\frac{1}{\omega_n}\left(\beta_n + \sum_{t\in T}\sum_{\pi\in\Pi^{nt}}(1-c_{nt})\,k_{n\pi}x_{n\pi}\right)\right\}$ is when cloud cover assumes the upper bound. Consequently, the BOX model can be transformed to the following formulation.

$$\max_{\mathbf{x} \in \mathcal{X}} \ \Lambda(\mathbf{x}, \overline{\mathbf{c}})$$

The formulation above is equivalent to the DSF model in which upper bounds replace corresponding cloud cover predictions. Predictably, the model leads to overly conservative solutions.

### Vector AutoRegression

The box uncertainty set assumed the highest possible cloud cover for all ground stations and throughout the entire duration of the scheduling horizon, which is unlikely to happen in the real world. Regarding modelling uncertainty of weather predictions, the previous model was overly simplistic. For instance, it ignored correlations on geospatial and temporal dimensions. Naturally, similar weather conditions are expected within a single geographical region. The same applies to weather conditions registered at a given location within a few hours of time difference.

Geospatial and temporal information could be incorporated by modelling cloud cover conditions using the VAR process [114]. It is a multivariate time series model in which the current state is a linear combination of some predefined number of past states and a white noise process. The number of previous states which influence the present state is known as the lag or the order.

Formally, the VAR process with the lag $h$ is defined as follows.

$$\mathbf{c}_t = \mathbf{A}_1 \mathbf{c}_{t-1} + \mathbf{A}_2 \mathbf{c}_{t-2} + \cdots + \mathbf{A}_h \mathbf{c}_{t-h} + \mathbf{a}_0 + \epsilon$$

For every time interval $t \in T$, the vector $\mathbf{c}_t \in \mathbb{R}^{|N|}$ contains time series variables which represent cloud cover observed at the given time for all ground stations. Matrices $\mathbf{A}_1, \ldots, \mathbf{A}_h \in \mathbb{R}^{|N| \times |N|}$ and the vector $\mathbf{a}_0 \in \mathbb{R}^{|N|}$ store coefficients of the model. The symbol $\epsilon$ denotes the white noise process.

The parameters of the time series model for the given lag are defined to follow the pattern observed in the training data set. For instance, the time series library [115], which was used to implement the model, fits the coefficients of the model using ordinary least squares. The lag controls the number of parameters in the model. Its value should be large enough to represent the data pattern accurately. At the same time, a bigger lag increases the risk of over-fitting. Consequently, the best practice is to build a parsimonious model in which the lag is determined by optimising some decision criterion that attempts to find a balance between the size of the model and the accuracy of the fit, i.e., by minimising the Bayesian Information Criterion (BIC) (a.k.a. Schwarz Information Criterion) [116].

The order selection procedure by minimisation of the BIC score performed on sample data and described in Appendix A.3 provides the rationale for using the lag set to one. The results presented in the literature give additional justification for selecting that setting. The VAR model of the same order was applied to simulate weather conditions, such as temperature and precipitation [117]. Regarding cloud cover prediction, other researchers [107] applied Markov processes which assume that the current state is derived from the previous state and the transition matrix. The assumption is analogous to using the lag one in the VAR model.

For the lag set to one, the time series parameters are reduced to the matrix $A_1$ and the vector $a_0$. Thus, hereafter, the subscripts are stripped to simplify the notation.

The uncertainty set $\mathcal{V}$ proposed in the thesis combines the VAR model and the box uncertainty set $\mathcal{B}$. Although the application of the VAR process to model weather conditions is not new, see, e.g., [117], to the best knowledge of the author, it is the first use of a time series model to construct an uncertainty set.

$$\mathcal{V} = \left\{ \mathbf{c} \in \mathbb{R}^{|N| \times |T|} \;\middle|\; \begin{array}{l} \underset{\sim}{c}_{nt} \leq c_{nt} \leq \bar{c}_{nt} \quad \forall n \in N \quad \forall t \in T \\ \mathbf{c}_t - \mathbf{e} \leq \mathbf{a}_0 + A\mathbf{c}_{t-1} \leq \mathbf{c}_t + \mathbf{e} \quad \forall t \in T \backslash \{1\} \end{array} \right\} \tag{3.11}$$

The white noise process $\epsilon$ from the VAR definition is replaced by the vector of residuals $\mathbf{e} \in \mathbb{R}^{|N|}_{\geq 0}$ which are designed to control the conservativeness of the uncertainty set. The parameter $\Gamma$, which bounds from above the sum of absolute deviations of uncertain parameters from their nominal values in the budgeted uncertainty set [8, 12], has a similar role. However, in the latter uncertainty set, the arrangement of deviations among the parameters affected by uncertainty could be arbitrary, regardless of the correlations. On the other hand, the values assumed by uncertain parameters in the proposed uncertainty set, must follow the VAR model that enforces spatial and temporal correlations. This feature makes the uncertainty set compatible with the behaviour of weather in the real world. The residuals are derived by analysing how frequently the uncertainty set covers weather patterns observed in the training set.

One should proceed with caution while setting the vector of residuals to high values. They cause the constraints that enforce spatial and temporal dependence between uncertain parameters derived from the VAR model to become loose and effectively not binding. Suppose in the extreme case that residuals are arbitrarily large. Then, the first constraint from Definition 3.11 will solely enforce the structure of the uncertainty set, turning $\mathcal{V}$ into the box uncertainty set $\mathcal{B}$.

The SatQKD problem with the $\mathcal{V}$ uncertainty set is defined as follows.

$$\max_{\mathbf{x}\in\mathcal{X}} \min_{\mathbf{c}\in\mathcal{V}} \ \Lambda(\mathbf{x},\mathbf{c}) \tag{3.12}$$

Formulation 3.12 is referred to in the sequel as the Box Time Series Uncertainty Set Model (BTS) model. The uncertainty set $\mathcal{V}$ is a polyhedron. Consequently, the BTS model can be reformulated as a MIP. The resultant formulation is presented in Proposition 1.

**Proposition 1.** *The following MIP is equivalent to the BTS formulation.*

*Proof.* The inner minimisation term in the 3.12 model can be transformed into a semi-infinite constraint.

$$\begin{aligned} \max \quad & \Theta \\ \text{s.t.:} \quad & \Theta \leq \lambda_n(\mathbf{x}, \mathbf{c}) \qquad \forall \mathbf{c} \in \mathcal{V} \quad \forall n \in N \\ & \mathbf{x} \in \mathcal{X} \end{aligned}$$

Without the loss of generality, one could select some ground station $n' \in N$ and proceed to the reformulation of the semi-infinite constraint for that station. In the first step, the expression $\lambda_n$ is replaced by its definition. The resultant expression is written below.

$$\begin{aligned} \sup \left\{ \omega_{n'}\Theta - \beta_{n'} - \sum_{t \in T} \sum_{\pi \in \Pi^{nt}} (1 - c_{n't})\, k_{n'\pi} x_{n'\pi} \right\} \leq 0 & \\ a + \sum_{n^* \in N} A_{n^*t-1} c_{n't-1} \leq c_{n't} + e & \qquad \forall t \in T \mid t > 1 \\ a + \sum_{n^* \in N} A_{n^*t-1} c_{n't-1} \geq c_{n't} - e & \qquad \forall t \in T \mid t > 1 \\ c_{n't} \leq \bar{c}_{n't} & \qquad \forall t \in T \\ c_{n't} \geq \underline{c}_{n't} & \qquad \forall t \in T \end{aligned}$$

Then, let introduce vectors of dual multipliers $\bar{\mathbf{v}}, \underline{\mathbf{v}}, \bar{\mathbf{z}}, \underline{\mathbf{z}} \in \mathbb{R}_{\geq 0}{}^{|N| \times |T|}$ which contain only non-negative entries. They are used to formulate the Lagrangian by dualising constraints from the definition of the uncertainty set $\mathcal{V}$.

$$
\begin{aligned}
L^{n'}(\mathbf{x}, \Theta, \bar{\mathbf{v}}, \underline{\mathbf{v}}, \bar{\mathbf{z}}, \underline{\mathbf{z}}) = \omega_{n'}\Theta - \beta_{n'} - \sum_{t\in T}\sum_{\pi\in\Pi^{nt}} k_{n'\pi}x_{n'\pi} \\
+\sum_{t\in T}\bar{c}_{n't}\bar{z}_{n't} - \sum_{t\in T}\underline{c}_{n't}\underline{z}_{n't} - \sum_{t\in T\ \mid\ t>1}(a-e)\bar{v}_{n't} + \sum_{t\in T\ \mid\ t>1}(a+e)\underline{v}_{n't} \\
+\tilde{c}_{n'1}(-\bar{z}_{n'1} + \underline{z}_{n'1} + \sum_{\pi\in\Pi^{n,1}} k_{n'\pi}x_{n'\pi} - \bar{v}_{n'2}\sum_{n^*\in N} A_{n^*1} + \underline{v}_{n2}\sum_{n^*\in N} A_{n^*1}) \\
+ \sum_{t\in T\ \mid\ t\in[2,|T|-1]} \tilde{c}_{n't}(-\bar{z}_{n't} + \underline{z}_{n't} + \sum_{\pi\in\Pi^{n't}} k_{n'\pi}x_{n\pi} \\
+\bar{v}_{n't} - \underline{v}_{n't} - \bar{v}_{n't+1}\sum_{n^*\in N} A_{n^*t} + \underline{v}_{n't+1}\sum_{n^*\in N} A_{n^*t}) \\
+\tilde{c}_{n'|T|}(-\bar{z}_{n'|T|} + \underline{z}_{n'|T|} + \sum_{\pi\in\Pi^{n'|T|}} k_{n',\pi}x_{n'\pi} + \bar{v}_{n'|T|} - \underline{v}_{n'|T|})
\end{aligned}
$$

The Lagrangian yields a meaningful upper bound whenever the following set of constraints is satisfied.

$$
\begin{aligned}
-\bar{z}_{n'1} + \underline{z}_{n'1} + \sum_{\pi\in\Pi^{n1}} k_{n'\pi}x_{n'\pi} - \bar{v}_{n'2}\sum_{n^*\in N} A_{n^*1} + \underline{v}_{n'2}\sum_{n^*\in N} A_{n^*1} = 0 \\
-\bar{z}_{n't} + \underline{z}_{n't} + \sum_{\pi\in\Pi^{n't}} k_{n'\pi}x_{n\pi} + \bar{v}_{n't} - \underline{v}_{n't} \\
-\bar{v}_{n't+1}\sum_{n^*\in N} A_{n^*t} + \underline{v}_{n't+1}\sum_{n^*\in N} A_{n^*t} = 0 \quad \forall t \in [2, |T|-1] \\
-\bar{z}_{n'|T|} + \underline{z}_{n'|T|} + \sum_{\pi\in\Pi^{n'|T|}} k_{n'\pi}x_{n'\pi} + \bar{v}_{n'|T|} - \underline{v}_{n'|T|} = 0
\end{aligned}
$$

To conclude, let construct the complete reformulation for the BTS model.

$$
\begin{aligned}
& \max \Theta \\
\text{s.t.:} \quad & \omega_n\Theta - \beta_n - \sum_{t\in T}\sum_{\pi\in\Pi^{nt}} k_{n\pi}x_{n\pi} + \sum_{t\in T}\bar{c}_{nt}\bar{z}_{nt} \\
& -\sum_{t\in T}\underline{c}_{nt}\underline{z}_{nt} - \sum_{t\in T|t>1}(a-e)\bar{v}_{nt} + \sum_{t\in T|t>1}(a+e)\underline{v}_{nt} \leq 0 && \forall n \in N \\
& -\bar{z}_{n1} + \underline{z}_{n1} + \sum_{\pi\in\Pi^{n1}} k_{n\pi}x_{n\pi} \\
& -\sum_{n^*\in N} A_{n^*1}\bar{v}_{n2} + \sum_{n^*\in N} A_{n^*1}\underline{v}_{n2} = 0 && \forall n \in N \\
& -\bar{z}_{nt} + \underline{z}_{nt} + \sum_{\pi\in\Pi^{nt}} k_{n\pi}x_{n\pi} + \bar{v}_{nt} - \underline{v}_{nt} \\
& -\sum_{n^*\in N} A_{n^*t}\bar{v}_{nt+1} + \sum_{n^*\in N} A_{n^*t}\underline{v}_{nt+1} = 0 && \forall n \in N \quad \forall t \in [2, |T|-1] \\
& -\bar{z}_{n|T|} + \underline{z}_{n|T|} + \sum_{\pi\in\Pi^{n|T|}} k_{n\pi}x_{n\pi} + \bar{v}_{n|T|} - \underline{v}_{n|T|} = 0 && \forall n \in N \\
& \Theta \in \mathbb{R}_{\geq 0} \\
& \mathbf{x} \in \mathcal{X} \\
& \bar{\mathbf{v}}, \underline{\mathbf{v}}, \bar{\mathbf{z}}, \underline{\mathbf{z}} \in {\mathbb{R}_{\geq 0}}^{|N|\times|T|}
\end{aligned}
$$

□

### 3.6.2 Distributionally Robust Optimisation

The RO models are defined using constraints on geometric properties. Apart from the specification of the support set, they do not introduce any assumptions regarding the probability distribution, which describes the behaviour of uncertain parameters. Nevertheless, if a data sample is available, some properties of the probability distribution could be estimated, leading to less conservative models of uncertainty. The next model employs the SOC representable ambiguity sets proposed by [118] and defines constraints on marginal moments. Specifically, it contains an equality constraint on the mean and an inequality constraint to bound variance from above.

To define the support set and the ambiguity set for every ground station $n \in N$

and the time interval $t \in T$, let $m_{nt}$ be the mean value of the uncertain cloud cover parameter $c_{nt}$, which is now considered a random variable. Furthermore, let $\mathbf{u} \in \mathbb{R}^{N \times T}$ be the vector of the auxiliary variables. Finally, the vector $\mathbf{d} \in \mathbb{R}_{\geq 0}{}^{|N| \times |T|}$ stores upper bounds of the variance derived from the sample data.

Let now define the support set $\mathcal{W}$ and the ambiguity set $\mathbb{F}$.

$$\mathcal{W} = \left\{ (\mathbf{c}, \mathbf{u}) \in \mathbb{R}^{N \times T} \times \mathbb{R}^{N \times T} \;\middle|\; \begin{array}{l} \underset{\sim}{\mathbf{c}} \leq \mathbf{c} \leq \bar{\mathbf{c}} \\ (c_{nt} - m_{nt})^2 \leq u_{nt} \quad \forall n \in N \quad \forall t \in T \end{array} \right\}$$

$$\mathbb{F} = \left\{ \mathbb{P} \in \mathcal{P}_0(\mathbb{R}^{N \times T} \times \mathbb{R}^{N \times T}) \;\middle|\; \begin{array}{l} (\tilde{\mathbf{c}}, \tilde{\mathbf{u}}) \sim \mathbb{P} \\ \mathbb{E}_{\mathbb{P}}(\tilde{\mathbf{c}}) = \mathbf{m} \\ \mathbb{E}_{\mathbb{P}}(\tilde{\mathbf{u}}) \leq \mathbf{d} \\ \mathbb{P}((\tilde{\mathbf{c}}, \tilde{\mathbf{u}}) \in \mathcal{W}) = 1 \end{array} \right\}$$

Due to the auxiliary variables $\tilde{\mathbf{u}}$, the ambiguity set $\mathbb{F}$ contains only affine expressions of random variables.

The SatQKD problem with the ambiguity set $\mathbb{F}$ and the support set $\mathcal{W}$ is defined below.

$$\max_{\mathbf{x} \in \mathcal{X}} \inf_{\mathbb{P} \in \mathbb{F}} \mathbb{E}_{\mathbb{P}}(\Lambda(\mathbf{x}, \tilde{\mathbf{c}})) \tag{3.13}$$

Henceforth, Formulation 3.13 is called Mean Standard Deviation Model (MSTD) due to the constraints present in the definition of the ambiguity set $\mathbb{F}$.

The MSTD formulation could be treated as a two-stage optimisation problem in which the decisions $\mathbf{x} \in \mathcal{X}$ are made in the first stage, and the scaling factor $\lambda$ is computed in the second stage. Uncertain parameters $\tilde{\mathbf{c}}$ are observed between stages.

According to the multistage nomenclature, Formulation 3.13 has the property of complete recourse. It means the second stage problem is feasible for all possible assignments of decisions variables in the first stage and all admissible realisations of uncertainty. In the context of the MSTD formulation, the complete recourse indicates the expression $\Lambda(\mathbf{x}, \mathbf{c})$ could be evaluated for all possible values the input arguments $\mathbf{x}$ and $\mathbf{c}$ could assume. For more information about complete recourse, the implications of this property on the complexity of multistage stochastic programs, and weaker variants of complete recourse, see [119].

Given the complete recourse holds, which is an essential prerequisite for Theorem 1 [118], the DRO formulation of the MSTD model could be transformed into the following RO formulation.

$$\begin{aligned} & R(\mathbf{x}) = \max f + \mathbf{g}^\top \mathbf{m} - \mathbf{h}^\top \mathbf{d} \\ \text{s.t.:} \quad & f + \mathbf{g}^\top \mathbf{c} - \mathbf{h}^\top \mathbf{u} \leq \Lambda(\mathbf{x}, \mathbf{c}) \qquad \forall (\mathbf{c}, \mathbf{u}) \in \mathcal{W} \\ & f \in \mathbb{R}, \quad \mathbf{g} \in \mathbb{R}^{|N| \times |T|}, \quad \mathbf{h} \in \mathbb{R}_{\geq 0}{}^{|N| \times |T|} \end{aligned}$$

The expectation operators in the ambiguity set $\mathbb{F}$ are replaced by the vectors of dual multipliers $\mathbf{g}$ and $\mathbf{h}$. In the next step, the expression $\Lambda(\mathbf{x}, \mathbf{c})$ is substituted by its definition.

$$R(\mathbf{x}) = \max f + \mathbf{g}^\top \mathbf{m} - \mathbf{h}^\top \mathbf{d} \tag{3.14}$$

$$\text{s.t.:} \quad f + \mathbf{g}^\top \mathbf{c} - \mathbf{h}^\top \mathbf{u} \leq \lambda_n(\mathbf{x}, \mathbf{c}) \qquad \forall (\mathbf{c}, \mathbf{u}) \in \mathcal{W}, \quad \forall n \in N \tag{3.15}$$

$$f \in \mathbb{R}, \quad \mathbf{g} \in \mathbb{R}^{|N| \times |T|}, \quad \mathbf{h} \in \mathbb{R}_{\geq 0}{}^{|N| \times |T|}$$

The formulation above is a RO model. Nonetheless, the uncertainty is not constraint-wise which is a common assumption in RO [57]. Specifically, consider Constraint (3.15), which contains cloud cover parameters for all ground stations. A separate instance of the constraint is defined for every ground station. Consequently, the same uncertain parameters are included in the definition of several constraints, which violates the constraint-wise assumption.

Although a transformation into constraint-wise uncertainty is always possible, the formulation is solved directly using the adversarial approach explained in the literature review, see Section 2.3.1. The adversarial method solves iteratively the relaxed RO formulation of the MSTD model presented below.

$$\max f + \mathbf{g}^\top \mathbf{m} - \mathbf{h}^\top \mathbf{d} \tag{3.16}$$

$$\text{s.t.:} \quad f + \mathbf{g}^\top \hat{\mathbf{c}} - \mathbf{h}^\top \hat{\mathbf{u}} \leq \lambda_n(\mathbf{x}, \hat{\mathbf{c}}) \qquad \forall (\hat{\mathbf{c}}, \hat{\mathbf{u}}) \in \hat{\mathcal{W}}, \quad \forall n \in N \tag{3.17}$$

$$\mathbf{x} \in \mathcal{X}$$

$$f \in \mathbb{R}, \quad \mathbf{g} \in \mathbb{R}^{|N| \times |T|}, \quad \mathbf{h} \in \mathbb{R}_{\geq 0}{}^{|N| \times |T|}$$

Constraint 3.15 from the original formulation defined for all elements of the set $\mathcal{W}$ is approximated by Constraint 3.17 restricted to the finite set of samples $\hat{\mathcal{W}}$ such that $\hat{\mathcal{W}} \subset \mathcal{W}$.

Initially, the set $\hat{\mathcal{W}}$ is empty. The adversarial method iteratively solves the relaxed formulation to optimality obtaining an incumbent solution with some assignment of the decision variables $(\hat{\mathbf{x}}, \hat{f}, \hat{\mathbf{g}}, \hat{\mathbf{h}})$. Then, given the assignment of the decision variables, the method checks whether the set $\mathcal{W}$ contains an element that violates Constraint (3.15). Such an element can be found by solving the following optimisation problem for each ground station.

$$\max_{(\mathbf{c}, \mathbf{u}) \in \mathcal{W}} \hat{f} + \hat{\mathbf{g}}^\top \mathbf{c} - \hat{\mathbf{h}}^\top \mathbf{u} - \lambda_n(\hat{\mathbf{x}}, \mathbf{c}) \tag{3.18}$$

If the optimal objective value of the problem above is positive for some ground station, then the optimal solution $(\mathbf{c}^*, \mathbf{u}^*)$ is added to the set $\hat{\mathcal{W}}$. Subsequently, the adversarial method commences the next iteration. Otherwise, if no solution $(\hat{\mathbf{c}}, \hat{\mathbf{u}})$ can attain a positive objective value for any of ground stations, then then the incumbent solution $(\hat{\mathbf{x}}, \hat{f}, \hat{\mathbf{g}}, \hat{\mathbf{h}})$ found by the adversarial method in the current iteration is feasible for the initial formulation, i.e., it satisfies Constraint 3.15. It is the stopping criterion that terminates the computations.

### 3.6.3 Stochastic Optimisation

The remaining formulations are examples of SO over scenarios. They are adaptations of the formulations presented in [43] based on the observation that the SatQKD problem is equivalent to the shortest path problem with a deadline defined as the target scaling factor that should be accumulated. As a result, traversing an arc in the vehicle routing domain is equivalent to transferring keys to a given ground station in the SatQKD setting.

Suppose the target scaling factor the solution should achieve by the end of the scheduling horizon is known and denote it using the symbol $\lambda'$. Furthermore, let $S$ represent the set of alternative weather forecasts.

#### Conditional Value at Risk

The application of the SO framework requires the assumption that cloud cover is a random variable and its probability distribution is known, i.e., $\tilde{\mathbf{c}} \sim \mathbb{P}$. Consequently, the decision-maker could be interested in finding a schedule that meets the target traffic index with some probabilistic guarantee, i.e., $\mathbb{P}(\Lambda(\mathbf{x}, \tilde{\mathbf{c}}) \geq \lambda') \geq 1 - \epsilon$.

Understandably, the expression $\mathbb{P}(\Lambda(\mathbf{x}, \tilde{\mathbf{c}}) \geq \lambda')$ is problematic to evaluate for an arbitrary distribution. Therefore, the true distribution is approximated using a random sample of scenarios, i.e., $\mathbf{c}^1, \ldots, \mathbf{c}^S$, where the vector $\mathbf{c}^s \in \mathbb{R}^{|N| \times |T|}$ contains the cloud cover predictions for a given scenario $s \in S$.

Taken together, let construct the following formulation.

$$
\begin{aligned}
& & \max \Lambda(\mathbf{x}, \mathbf{c}) & \qquad \text{(CHAN)} \\
& \text{s.t.:} & \mathbb{P}\left(\Lambda(\mathbf{x}, \tilde{\mathbf{c}}) \geq \lambda'\right) \geq 1 - \epsilon & \qquad (3.19) \\
& & \mathbf{x} \in \mathcal{X} &
\end{aligned}
$$

The objective function is the scaling factor defined using the cloud cover predictions from the official weather forecast. Intuitively, such cloud cover predictions should be the most likely among the possible scenarios. Constraint (3.19) ensures that the target scaling factor is satisfied in the stochastic sense with probability $1-\epsilon$ for some $\epsilon \in (0,1)$ and assuming the uniform probability distribution over the set of scenarios.

The CV@R is applied to approximate the CC (3.19).

$$\begin{aligned} & \max \Lambda(\mathbf{x}, \mathbf{c}) \\ \text{s.t.:} \qquad & \mathrm{CVaR}_{1-\epsilon}\left(\lambda' - \Lambda(\mathbf{x}, \tilde{\mathbf{c}})\right) \leq 0 \\ & \mathbf{x} \in \mathcal{X} \end{aligned} \tag{3.20}$$

Proposition 2 presents an equivalent MIP formulation for the optimisation problem above.

**Proposition 2.** *Assuming a uniform probability distribution over scenarios, the SatQKD optimisation problem with a CC approximated using the CV@R has the following reformulation.*

$$\begin{aligned} & \max \Lambda(\mathbf{x}, \mathbf{c}) \\ \textit{s.t.:} \qquad & \gamma + \frac{1}{\epsilon|S|}\sum_{s\in S} y_s \leq 0 \\ & y_s \geq \lambda' - \Lambda(\mathbf{x}, \mathbf{c}^s) - \gamma && \forall s \in S \\ & y_s \geq 0 && \forall s \in S \\ & \gamma \in \mathbb{R}, \quad y_s \in \mathbb{R} && \forall s \in S \\ & \mathbf{x} \in \mathcal{X} \end{aligned}$$

One can prove the proposition by substituting the CV@R by its definition and transforming the expectation of a discrete uniform probability distribution into a set of linear inequalities. Section 6 in [43] provides an example of such a transformation for the TSP with a Deadline.

### Essential Riskiness Index

Formulations employing CC's require satisfaction of probabilistic constraints with some prescribed guarantee. As long as the probability of the constraint violation is less than a given $\epsilon$, the solution is considered feasible, and the magnitude of the constraint violations is not relevant. On the other hand, the decision-maker may want to make more extensive violations less likely. The indices of riskiness summarised in Section 2 of the literature review are the modelling tools designed for that purpose.

Among the available riskiness indices, the ERI is selected because it can be defined using MILP. The model below presents the SatQKD problem formulated as the riskiness index optimisation.

$$\underset{\mathbf{x}\in\mathcal{X}}{\mathrm{ERI}} \left(\Lambda(\mathbf{x}, \mathbf{c}) - \lambda'\right) \tag{3.21}$$

Proposition 3 presents an equivalent MIP model.

**Proposition 3.** *Assuming the uniform probability distribution over scenarios, the SatQKD problem with the ERI decision criterion can be formulated as a MIP.*

$$\begin{aligned}
& & \min \alpha & & \\
\textit{s.t.:} & & \sum_{s\in S} y_s \leq 0 & & \\
& & y_s \geq -\alpha & & \forall s \in S \\
& & y_s \geq \lambda' - \Lambda(\mathbf{x}, \mathbf{c}^s) & & \forall s \in S \\
& & \alpha \in \mathbb{R}_{\geq 0} \quad y_s \in \mathbb{R} & & \forall s \in S \\
& & \mathbf{x} \in \mathcal{X} & &
\end{aligned}$$

One can derive the proof by performing analogous steps as detailed in Proposition 2, and Corollary 1 in [43].

## Concluding Remarks

The chapter described a methodology for scheduling data transfers from a satellite deployed into a LEO to the network of ground stations. Out of the possible optimisation problems, the featured problem was the SatQKD, which is an emerging application in Quantum Communication. Despite the targeted application, the modelling techniques presented in the chapter are transferable to optimisation problems involving scheduling satellites circulating along orbits whose parameters remain constant.

To the best knowledge of the author, it is the first study of the SatQKD problem from the optimisation perspective. The part dedicated to the deterministic problem contains its formulation and the complexity analysis proving that the SatQKD problem is NP-hard. Therefore, solving it using a MIP solver is well justified.

Subsequently, a suite of alternative SatQKD models for representing cloud cover uncertainty was developed utilising RO, DRO and SO frameworks. All formulations by design can be parametrised using cloud cover predictions from official weather forecasts and are solvable by MIP solvers. Despite the relevance of the problem and the inherent source of uncertainty apparent in its domain, i.e., optical communication is sensitive to cloud cover conditions whose forecasts are naturally affected by uncertainty, the author is not aware of other research in this area.

Regarding methodological contributions, a new uncertainty set built based on the VAR time series model was proposed. The formulation preserves geospatial and temporal correlations, which are naturally observed in meteorological phenomena. Apart from the desirable property of preserving correlations, the model can be reformulated as a MILP. Hence, it is easily solvable.

The next chapter presents computational results that demonstrate that some formulations for modelling uncertainty derived here considerably improve upon the deterministic model with the folding horizon.

# Chapter 4

# Application: Satellite Quantum Key Distribution

## 4.1 Introduction

The chapter validates the methodology developed for scheduling satellite space-to-ground communication. The content is split into two parts: the deterministic model and the formulations with the treatment of uncertainty. The deterministic model is used to compute optimal schedules given historical weather observations. The analysis of such solutions facilitates performance evaluation and tuning. For instance, one could select the orbital parameters of the satellite to maximise the volume of data delivered to the network. Although the deterministic model is suitable for the performance assessment of a given design of the communication system, the approach is inferior for scheduling space-to-ground data transfers in the future due to uncertainty in cloud cover predictions. Therefore, the second part of the chapter compares various models for handling uncertainty by solving a sequence of scheduling problems using official weather forecasts.

### Structure of the Chapter

The chapter is structured as follows. Section 4.2 covers the literature on the applications of deterministic scheduling in satellite operations and scheduling under uncertainty. Section 4.3 explains the architecture of the SatQKD communication system. Section 4.4 describes the implementation details. Section 4.5 presents computational results obtained using the deterministic model: the selection of orbital parameters for the satellite, the impact of the spacecraft's illumination on communication windows, and the performance evaluation of the system using historical weather observations. Section 4.6 compares the models for the treatment of uncertainty. Some concluding remarks close the chapter.

## 4.2 Literature Review

The literature review on scheduling satellite operations focuses on space-to-ground communication from a spacecraft deployed into the LEO. It covers deterministic scheduling and scheduling under uncertainty. The discussion of the deterministic models does not differentiate between radio and optical communication as the problem formulations and the solution methods are independent of the type of the communications system. On the other hand, scheduling communication under weather uncertainty is well-motivated only for optical communication because its radio frequency counterpart is resilient to conditions in the atmospheric channel.

### 4.2.1 Deterministic Scheduling

The need for optimisation in scheduling satellite operations is well justified, i.e., the volume of data produced by EOS exceeds the amount which can be effectively downloaded using the existing ground station infrastructure [97]. Therefore, optimisation ensures efficient use of available resources. Moreover, besides the limited capacity of the ground station segment, scheduling satellite communication in the real world involves human operators, see, e.g., [120]. However, researchers warn that the current model of operations relying heavily on human intervention is not scalable to service large constellations of satellites in the future [121].

The Operations Research community considers scheduling satellite operations as an application of Machine Scheduling [122] or Multiprocessor Scheduling [123]. The former group is relevant when a satellite is modelled as a machine. The latter class is suitable for scheduling a constellation of satellites in which a spacecraft is represented as a resource [121]. Example operations executed by the satellites are orbital manoeuvres, data collection tasks, command uplinks, payload downlinks, and health checks. Besides individual details of a given job, every operation has the following properties: release date, due date, estimated duration, and time windows in which the task can be executed [124]. If multiple clients share access to the satellite or the system is oversubscribed, the operations could be assigned priorities or rewards to be collected for their execution. If the priorities are defined, then the scheduling problem can be classified as the Resource-Constrained Project Scheduling [125], i.e., scheduling problems that involve downlink of satellite images subject to deadlines and priorities were studied by [126].

Regarding scheduling satellite communication, the computational approach seems to depend on the scientific community. Hybrid methods are popular in the aerospace engineering community, which studied genetics algorithms [127], heuristics and metaheuristics [124, 126] often combined with LS and constraint propagation. On the other hand, the OR community postulates modelling the scheduling problem using time discretisation [96, 97]. Such formulations were shown to have a strong bound of its LP relaxation. Consequently, formulations for realistic problem sizes can be solved to provable optimality using commercial MIP solvers, which is a considerable advantage over heuristic approaches.

Scheduling space-to-ground communication from one EOS to a fixed network of ground stations was studied by [96]. The model included data acquisition as well as monitoring of energy generation and consumption. The researchers formulated the problem as a Mixed-Integer Programming (MIP) with time discretisation and demonstrated that realistic problem instances (i.e., which are not made artificially difficult) are solvable to optimality even without the exploration of the branch-and-bound tree. The formulation was further extended to support a constellation of satellites by [97] who demonstrated that the modified problem retains the appealing computational complexity. Understandably, besides the case studies, the models presented in the literature are general purpose. Hence, they may require some adaptation to represent the design of the spacecraft faithfully. For example, a satellite may have multiple antennas which can operate in several configurations that offer different ratio between power consumption and bitrate. Nonetheless, the researchers [121] demonstrated that the model with time discretisation could be easily extended to support additional constraints.

Suppose the satellite has a fixed orbit (i.e., the spacecraft does not perform attitude and orbit control manoeuvres). In that case, the model with time discretisation can be solved directly, without reformulation techniques and decompositions [96, 97]. In contrast, a spacecraft that can change its orientation by performing attitude control manoeuvres is called an agile satellite. Spacecrafts with such capability are typically considered in scheduling EO because they can collect several images of the same target during a single visibility window [128]. Agile satellites are significantly harder to schedule. For instance, researchers [128] abandon solving the formulation directly in favour of a heuristic based on the column generation scheme. The authors generated columns only at the root node and then explored the branch-and-bound tree without adding new columns to the pool.

In the real world, satellites belong to different organisations that share access to a ground station infrastructure. Typically, the number of requests to handle satellite communication considerably exceeds the capacity of the ground station segment. A well-known example in the literature is the United States Airforce Satellite Control Network (AFSCN) in which a system of nine ground stations hosting 16 antennas supports more than one hundred satellites. The optimisation problem of allocating antennas to spacecraft for a requested duration within a communication window is known as Satellite Range Scheduling [129,130]. The ground station segment is oversubscribed, and conflicting requests for readiness to accept incoming satellite communication are prevalent. The objective function for the problem is defined either as minimisation of the number of declined requests or minimisation of the total time when an antenna is assigned to multiple conflicting requests. The second formulation of the objective is more convenient for scheduling long communication requests. The resolution of conflicting demands (a.k.a., deconfliction) is the responsibility of human operators [130] who carry out negotiations with the involved parties. Currently, the operators of the AFSCN use an iterative algorithm that schedules the most restricted requests first, avoiding the depletion of scarce resources [131]. Declined requests are then processed using a collection of business rules that tentatively relax some operational constraints to find free capacity and resolve conflicts. From a theoretical perspective, the complexity of the deconfliction was studied by [132] using a simplistic example with multiple satellites and one ground station. In a general case, with the setup cost and no task preemption, the problem is NP-hard [132]. The complexity remains an open problem when the task preemption is allowed, and no setup cost is incurred.

The network of ground stations can either be given as input or be the subject of optimisation. The former is a common assumption for radio frequency communication as such ground stations have a monolithic structure, they are already operational, and their locations are well known (i.e., Estrack). On the other hand, the problem of finding a set of suitable ground stations sites is open for optical ground stations. The approach adopted in the literature resembles the Warehouse Selection Problem [133], i.e., to find a subset of ground stations from a pool of shortlisted candidates subject to budget restrictions [91] with the MaxPDT as the objective function [91, 92]. For a given set of ground stations, the objective can be computed using dynamic programming [92, 134]. If the number of candidates in the pool is small, then the problem is solved through direct enumeration. Otherwise, one could apply LS. The case studies on the selection of ground stations for optical communication were conducted using historical weather observations over many years and focused on different areas (i.e., Germany [134], Europe [134, 135], and the entire world [91, 134]). Nonetheless, researchers did not account for two phenomena. Firstly, the illumination of the spacecraft by the Sun, which becomes vital for ground stations located at high latitudes. Furthermore, the transfer rate between the ground station and the satellite was constant during the communication window. It is an oversimplification as in the real world, the elevation angle between the ground station and the spacecraft affects the length of the atmospheric channel the laser beam goes through. The longer the distance, the more information is lost due to errors. To fill these gaps, the formulations introduced in the previous chapter model both the illumination of the spacecraft and the variable transmission rate, which depends on the elevation angle.

### 4.2.2 Scheduling Under Uncertainty

To the best of the author's knowledge, scheduling satellite communication subject to weather uncertainty has not been studied in the literature. Given the fact that the primary type of space-to-ground communication, radio frequency communication, is insensitive to weather conditions, the lack of interest in modelling cloud cover uncertainty seems justified. However, the state of the field is likely to change with the increasing adoption of optical communication.

Contrary to scheduling satellite communication, there are a few articles [88,89,106] regarding the modelling of cloud cover conditions in the related application of scheduling image acquisition using EOS. Predictably, if the target is hidden underneath the clouds, then repeating the image acquisition may be required. It causes disruptions and delays, which could have been avoided if operations that were more likely to render an acceptable image were scheduled instead. A pragmatic approach proposed by [106] applies SO with Bernoulli probability distribution, i.e., an image is either accepted or rejected due to weather conditions adversely affecting the quality of the picture. The objective was defined as the maximisation of the expected reward for successful observations and the number of image acquisitions accomplished. On the other hand, instead of the objective, a minimum reward could be subject to a CC [88]. Chance Constraints are presented in Section 2.1 of Chapter 2. Given the Bernoulli probability distribution, a CC can be approximated using the SAA that is explained in Section 2.2.1 of Chapter 2. In the most recent paper [89], uncertain rewards for images acquired by a constellation of agile EOS are modelled using the budgeted uncertainty set, see Section 2.2.2 in Chapter 2. The authors recommend solving the robust reformulation using column generation, in which a column corresponds to a sequence of image acquisitions for the given orbit. Columns were generated by searching for a minimum cost path in a directed acyclic graph which encodes sequences of image acquisitions.

To conclude, the models for handling cloud cover uncertainty considered in the literature are limited to the Bernoulli probability distribution and the budgeted uncertainty set, which are well-established frameworks. In contrast, the methodology developed in Chapter 3 applies more advanced modelling approaches (i.e., DRO, riskiness index optimisation). Furthermore, it proposes a new uncertainty set that preserves correlations between uncertain parameters over time and space. The computational results discussed in the chapter were obtained using official weather forecasts and validated in real weather conditions. Results presented in the literature, the author is aware of, lack such comparison.

Alongside the efforts to make schedules more resilient by modelling weather uncertainty, researchers attempt to increase the resiliency of optical communication, i.e., by using a geostationary satellite as a relay, by employing radio communication as a backup in case of adverse weather conditions or by deploying a high altitude platform above the level of clouds. For more information on these topics, an interested reader is referred to [136].

## 4.3 Communication System

Let consider a potential quantum-secured communication system that consists of one satellite and five optical ground stations spread across the UK. The ground stations are located in cities with a large population (i.e., at least 600,000). They are also geographically dispersed: Birmingham, Bristol, Glasgow, London, and Manchester. The rationale for selecting a ground station in a given city is provided in [20].

Figure 4.1 presents an artistic impression of the communication system.

The system operates in the downlink configuration, i.e., the satellite is the transmitter. Therefore, the spacecraft is equipped with a quantum source such as a laser, an optical assembly for precise pointing, and a quantum random number generator [103]. Given a reasonable assumption that a potential attacker will not compromise the satellite located in space, let adopt the so-called trusted-node architecture in which the satellite generates keys and sends them to ground stations sequentially. After a pair of ground stations receive their private keys, the satellite broadcasts the XOR hash of both keys over a public channel. The ground stations know their private keys and can

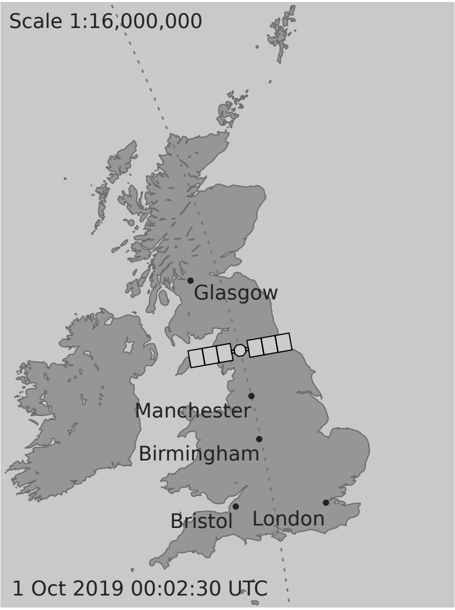


Figure 4.1: Visualisation of the prototype of the national SatQKD communication system. It consists of five optical ground stations spread across the UK. Communication between ground stations is encrypted with symmetric key pairs distributed via a satellite placed into a Sun-Synchronous Orbit. The satellite passes over the UK mainland territory towards the North direction around local midnight. The dashed arc denotes the ground track the spacecraft follows.

decrypt information used to create the XOR hash. Henceforth, the ground stations own a shared key intended for symmetric encryption [137]. A SatQKD system can also operate in a different context, i.e., with the satellite as the receiver (a.k.a., the uplink configuration), and by adopting the so-called untrusted-node architecture. The pros and cons of other operational arrangements are briefly discussed in [20].

Ground stations receive keys in a raw format which requires post-processing for privacy amplification after a sufficient number of raw keys is received [138]. Consequently, the readiness of keys delivered to the ground station is delayed. Let assume that the operational requirements of the communication system for a given scheduling horizon are satisfied using keys available before the start of the scheduling horizon. Hence, it is sufficient to evaluate the objective function once at the end of the scheduling horizon. Besides the enhanced security and simplified evaluation of the objective, the additional advantage of buffering keys is increased resiliency against adverse weather conditions, which negatively impact the number of keys delivered.

Table 4.1 presents the weights assigned to ground stations in the communication system. Recall that weights assigned to ground stations were an indispensable element of the SatQKD problem definition as a ground station should receive the volume of keys that is proportional to the assigned weight. For a ground station $n$, its weight was denoted using $\omega_n$. The weights in the table are proportional to the number of premises in a given location with broadband access of 300 $MBit/s$ download speed or higher. The statistics were collected by the governmental body, which controls the telecommunication sector in the UK (a.k.a., the Office of Communications [139]). The weights are normalised, so their sum equals one and rounded to three decimal places.

Table 4.1: Weights assigned to ground stations. The weights are derived based on the number of premises with access to the High-Speed Broadband (HSBB) located in a given city. The first column contains the name of the city. The second column presents the number of premises with the HSBB. The last column stores the weight assigned to the ground station.

| City | Premises with HSBB | Weight |
|---|---:|---:|
| Birmingham | 599,540 | 0.196 |
| Bristol | 297,599 | 0.097 |
| Glasgow | 381,494 | 0.125 |
| London | 1,421,422 | 0.465 |
| Manchester | 353,884 | 0.116 |

## 4.4 Implementation Details

Optimisation models were implemented in C++ using Gurobi MIP solver. Software for creating problem instances and the analysis of computational results was developed in Python. The VAR models were trained using the statsmodels library [115]. The software developed is available open-source [21] to facilitate the reproduction of results discussed in the chapter.

The optimisation problems were solved on a single workstation with the AMD Ryzen 7 processor and 32 GB of RAM. Weather forecasts were updated every 3 hours, and the communication windows were discretised in 15 seconds steps. The optimisation was stopped after four hours of computational time or reaching the optimality gap of $10^{-4}$.

The astrodynamics model to propagate the position of the satellite was adopted from the paper [20]. The same source explains the formula used to compute the elevation angle between the spacecraft and the ground station. The spacecraft could communicate with the ground station when the elevation angle is above $15^{\circ}$. The periods when the satellite remains in the complete Earth shadow were computed using the algorithm explained in [140]. The model of the atmospheric channel to estimate the key transfer rate when no clouds are present (a.k.a., could free line of sight) was described in [20].

## 4.5 Deterministic Scheduling using Historical Observations

Results presented in this section were obtained by solving the Deterministic Single Forecast Model (DSF) model, referred to as Formulation 3.1. It is parametrised by a vector of weights $\omega$, a multidimensional vector storing the number of keys $\mathbf{k}$ that can be delivered to a selected ground station at a given time interval, and a multidimensional vector of cloud cover conditions $\mathbf{c}$. Weights $\omega$ are set the values presented in Table 4.1. The vector $\mathbf{k}$ is computed using the key transfer rate model [20] with the initial orbital parameters for the satellite passed as input. Cloud cover conditions are derived from the data set of historical weather observations compiled by [113]. It contains weather conditions registered in the cities considered in the study for a continuous period of six years between 2013 and 2019. Weather conditions were collected using a network of meteorological stations and updated every hour. Cloud cover is represented as an integer value in the range [0,100].

The analysis of the historical schedules provides information about the possible performance a given design of the communication system could achieve. Such optimisation could lead to making informed decisions regarding the selection of ground stations and finding the most suitable orbital parameters for the spacecraft. The weather conditions observed in the past are known precisely. Hence there is no need for the treatment of uncertainty, and one could apply Formulation 3.1 without negative consequences.

Finding a schedule for the time frame over multiple years is computationally challenging due to the size of the formulation. Instead, schedules could be computed sequentially using the folding horizon approach. The position of the satellite and the number of keys ground stations store in their buffers are synchronised during the transition between subsequent stages.

Scheduling using historical weather information is not confined to the forecasting horizon adopted in official weather forecasts. For that reason, the scheduling horizon was extended to the period of one year. It enabled the solver to access information about changes in the illumination of the spacecraft by the Sun and weather seasonality patterns to make scheduling decisions that bring long-term profit.

### 4.5.1 Orbital Parameters

The spacecraft is placed into a circular orbit 566.897 $km$ above Earth's surface. At the given altitude, the satellite makes 15 complete revolutions around the globe, i.e., one pass every 96 minutes. The inclination is set to 97.658° to compensate the drift in the Right Ascension of the Ascending Node (RAAN) caused by the Earth's nodal precession. Consequently, the total drift in RAAN accumulated throughout a year is 360°. As a result, the time and the duration of communication windows between the ground station and the satellite observed a given night of the year is repeated every year. An orbit with such a feature is called a Sun-Synchronous Orbit (SSO). The longer of the two visibility windows is centred around midnight by setting the argument of the latitude to 46° given the initial epoch at 00:00:00 UTC, the 1st of January 2013. Considerate planning of communication windows is important during the summer months. In that period, communication with ground stations located at high latitudes, i.e., ground stations based in the UK, is severely restricted due to the illumination of the spacecraft by the Sun. This topic is explored in detail in the next section.

The settings for orbital parameters proposed above have a rational justification and can be derived without optimisation. On the other hand, it is not self-evident which initial RAAN value to choose to maximise the number of keys distributed to the communication system. Such a problem could be systematically addressed by computing a Service Level (SL) parametrised by some threshold value, e.g., 99% for a shortlisted set of candidate values the initial RAAN could assume. The SL is defined as the maximum number of keys a ground station could use within some prescribed recurring period, e.g., a week, without exceeding the capacity of the buffer for the number of periods in the given time frame proportional to the threshold value. The SL is estimated for all nodes in the communication system simultaneously assuming the use of keys proportionally to the weight assigned to the ground station. Therefore, regardless of the target ground station, the SL provides the same insight into the performance of the communication system.

The initial RAAN value is narrowed to the range $[90.5°, 115.5°]$. The search for the most suitable configuration performs an exhaustive enumeration of all possible settings incrementally with the step of 1°. The vector $\mathbf{k}$ that stores the number of keys possible to deliver is computed according to the model [20] for a given initial RAAN configuration. Subsequently, Formulation 3.1 is solved to obtain schedules of data transfers for the time frame the dataset of historical weather observation covers. Finally, the SL is assessed for the resultant schedules.

Table 4.2 presents the maximum weekly key consumption in London for the given RAAN value which can be maintained at 99% SL over six years.

The example illustrates the importance of a careful analysis for the selection of the RAAN parameter. In the best configuration, i.e., the RAAN set to 109.5°, the spacecraft delivered over twice as many keys as in the least effective configuration, i.e., the RAAN set to 97.5°. Computational results presented in the sequel were obtained by using the RAAN setting that yields the best performance for the London ground station. Since keys are delivered proportionally to the weights expressing the importance of the ground stations, the same RAAN setting maximises the number of keys delivered to the entire network.

Table 4.2: Maximum number of keys which can be consumed by the London ground station weekly at the 99% SL for the given RAAN configuration.

| | RAAN | London 99% SL |
|---|---|---|
| 1 | 90.5 | 6263 |
| 2 | 91.5 | 5821 |
| 3 | 92.5 | 5435 |
| 4 | 93.5 | 4998 |
| 5 | 94.5 | 4626 |
| 6 | 95.5 | 4310 |
| 7 | 96.5 | 4133 |
| 8 | 97.5 | 4059 |
| 9 | 98.5 | 4096 |
| 10 | 99.5 | 4278 |
| 11 | 100.5 | 4547 |
| 12 | 101.5 | 4980 |
| 13 | 102.5 | 5524 |
| 14 | 103.5 | 6128 |
| 15 | 104.5 | 6784 |
| 16 | 105.5 | 7309 |
| 17 | 106.5 | 8072 |
| 18 | 107.5 | 8607 |
| 19 | 108.5 | 9039 |
| 20 | 109.5 | 9295 |
| 21 | 110.5 | 9076 |
| 22 | 111.5 | 8904 |
| 23 | 112.5 | 8356 |
| 24 | 113.5 | 7742 |
| 25 | 114.5 | 7040 |
| 26 | 115.5 | 6430 |

### 4.5.2 Spacecraft Illumination

The Earth's movement around the Sun, except for seasonality changes in weather patterns and daytime, has an impact on the illumination of spacecraft placed into the LEO. Specifically, a satellite passing over some territory after dusk could still be in sunlit. According to the assumptions outlined above, such conditions prohibit space-to-ground communication. This phenomenon is pronounced during the summer months for ground stations at high latitude locations, such as the mainland UK territory.

Figure 4.2 displays the total duration of communication windows for all ground stations. Comunication windows were computed by propagating the satellite movement in the sky from the initial orbital parameters derived in Section 4.5.1 according to the model of the orbital dynamics outlined in [20]. The same article explains the procedure to compute the elevation angle between the ground station and the satellite. The communication can be established if the elevation angle is not smaller than 15°. Furthermore, the satellite must remain in the complete shadow (a.k.a., umbra). Such periods were computed using the conical shadow model [140]. The total duration of communication windows was aggregated daily for the entire time range the data set of historical weather observation covers.

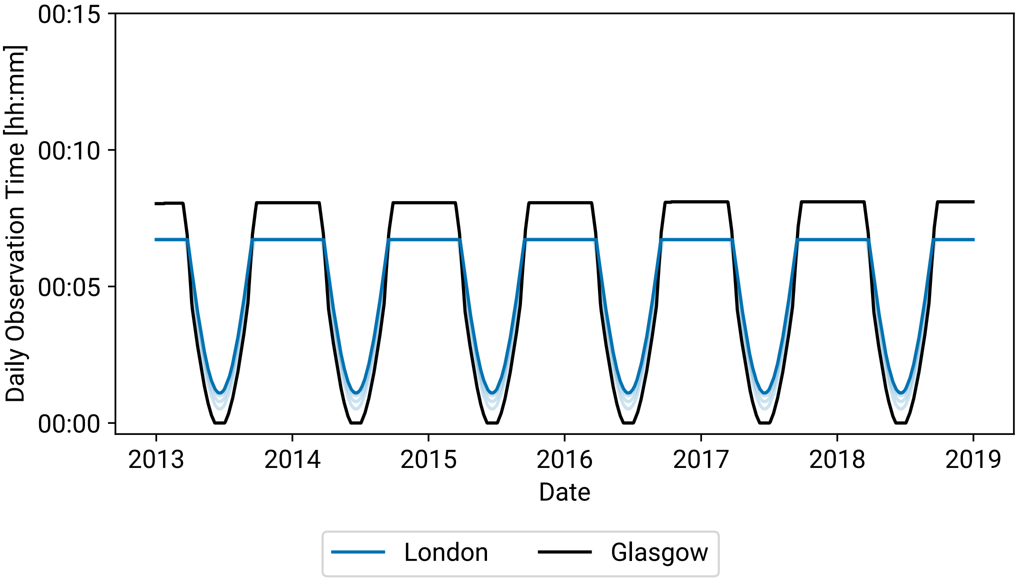


Figure 4.2: Total duration of communication windows with the satellite observed daily for ground stations located in London and Glasgow. Time series painted in light blue correspond to other ground stations: Bristol, Birmingham and Manchester ordered from top to bottom.

The impact of the satellite's illumination on the duration of communication windows depends on the latitude of the ground station. The phenomenon is not apparent between late autumn and early spring, i.e., from the 27th of September to the 14th of March for Glasgow and from the 15th of September until the 26th of March for London. During that period, the total duration of communication windows is the longest throughout a year and constant. All ground stations observe a communication window around midnight. Furthermore, Glasgow, which is located at the highest latitude among the ground stations, has an additional communication window around 10:30 PM. Therefore, it has the longest total duration of communication windows in the winter months. On the other hand, the higher the latitude of the ground station, the more affected it is due to the satellite's illumination for the remaining period of the year. For instance, the total duration of communication windows for Glasgow gradually starts decreasing from 8 minutes and 4 seconds on the 14th of March to a complete blackout on the 7th of June. The lack of ability to communicate with the satellite persists until the 5th of July. No other ground station loses entirely the ability to communicate. Nonetheless, the duration of communication windows is severely limited. London, which is the least affected ground station, can communicate with the satellite only for 1 minute and 6 seconds during the satellite illumination peak. In contrast, its total communication time was 6 minutes and 43 seconds during the winter season.

Figure 4.3 illustrates periods when the London ground station observes communication windows with the satellite. The Y-axis displays UTC time to avoid discontinuities due to daylight saving time shifts.

The figure presents the occurrence and the duration of communication windows repeated with yearly seasonality. Furthermore, communication windows always start at the same time, which is convenient from the operations perspective. These behaviours are a consequence of the careful selection of orbital parameters. The reproducibility of communication windows every year was accomplished by adjusting the orbit inclination to compensate for the drift in the RAAN caused by Earth's precession. It is impossible to eliminate the variability in the duration of communication windows between March and September caused by the illumination of the satellite by the Sun. Even though the communication windows are positioned in the middle of the night, they remain severely affected.

### 4.5.3 Weekly Scheduling

Figure 4.4 is a visualisation of the SatQKD schedule computed for the first week of January 2018. The schedule was obtained by solving Formulation 3.1, with the vector

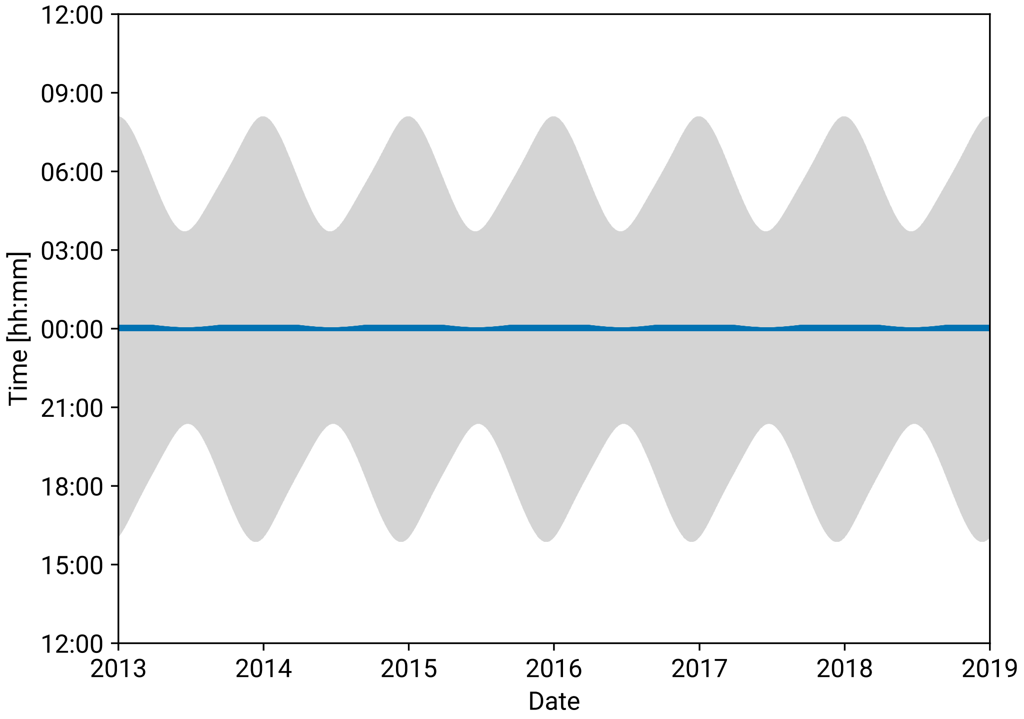


Figure 4.3: Occurrence and duration of communication windows for the London ground station. Grey colour indicates the period between dusk and dawn.

of weights $\omega$ set according to Table 4.1, the vector of keys possible to deliver $\mathbf{k}$ computed using the model outlined in [20], and the vector of cloud cover defined based on the dataset of historical weather observations described in the opening of Section 4.5. The initial orbital parameters of the satellite were explained in Section 4.5.1. The time frame is narrowed down to one week for readability as the primary motivation for presenting the plot is giving insights into the nature of the scheduling problem, i.e., the number of communication windows, their duration, and the curve of the key transfer rate. The selected period occurs between late autumn and early spring to eliminate the effect of the spacecraft's illumination by the Sun. Therefore, communication windows are the same every week within that time frame. The transfer rate also has a similar shape, but its magnitude is adjusted to the observed cloud cover conditions.

Depending on the location, at most two communication windows are observed every night. The first one is available only for Glasgow around 10:30 PM and lasts less than 1.5 minutes. The second communication window is observed around midnight for all

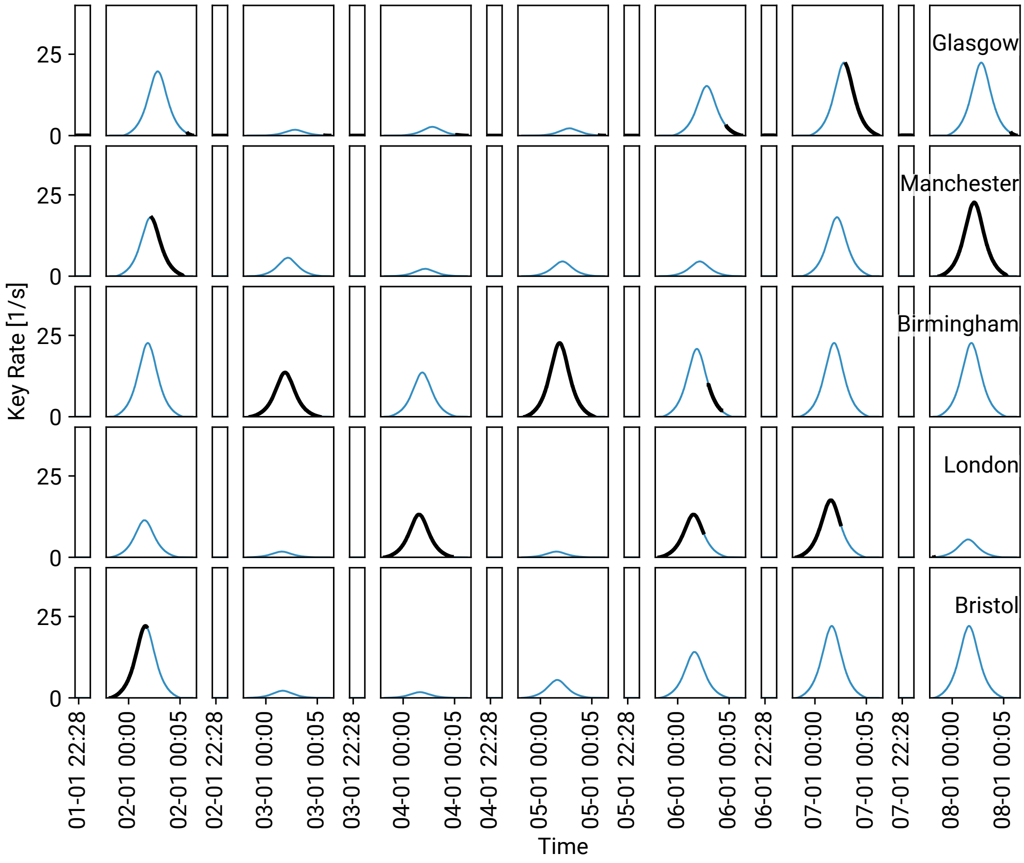


Figure 4.4: Weekly schedule for the first week of January 2018. Every column of the plot represents a communication window. Blue curves illustrate the effective transfer rate to the given ground station after accounting for the cloud cover conditions. Black segments represent periods allocated for communication with a given ground station.

ground stations, and its duration is 6 minutes. The transfer rate is symmetric, with a peak in the middle of the communication window. Moreover, longer communication windows achieve a higher transfer rate in the same weather conditions.

### 4.5.4 Long-Term Scheduling

The long-term schedule is computed for the years between 2013 and 2019 using Formulation 3.1 with the folding horizon approach. Each stage lasted one year. The vector of weights $\omega$ is set according to the values presented in Table 4.1. The vector of keys possible to deliver in the clear line-of-sight is computed using the model [20] given the initial orbital parameters of the spacecraft outlined in Section 4.5.1. Cloud cover conditions represented by the vector **c** are defined in the dataset introduced at the beginning of Section 4.5. The final schedule is a concatenation of partial schedules solved to optimality for every stage. The number of the keys distributed to the given ground station throughout a week is illustrated in Figure 4.5.

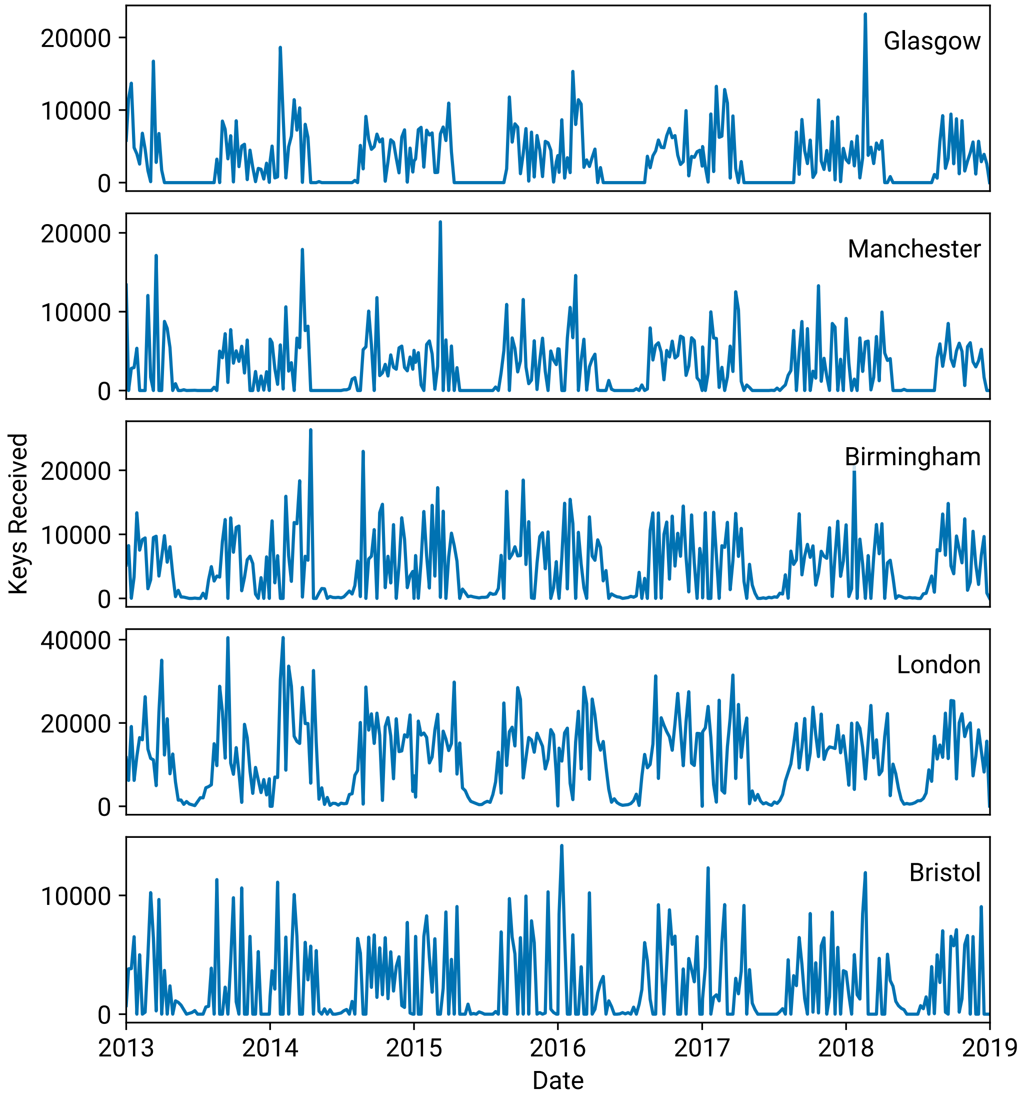


Figure 4.5: Long-term SatQKD schedule. The number of keys delivered weekly to every ground station between the years 2013 and 2019.The scale of the Y-axis is adjusted accordingly for every ground station.

Roughly, the time series representing the keys delivered to a given ground station is repeated every year. Considerably fewer keys are distributed to all ground stations during the summer months due to the illumination of the spacecraft by the Sun. Specifically, this phenomenon causes Glasgow to lose the ability to receive keys for certain weeks during the summer, i.e., between the 7th of June and the 3rd of July every year. Consequently, to maintain the connectivity with the remaining nodes, the Glasgow ground station node must accumulate a sufficient number of keys in advance.

Predictably, London is the ground station with the highest importance in the network. Consequently, it receives a large proportion of new keys every week. Birmingham acquires keys similarly, but in smaller quantities, as the number of keys is proportional to the weight of the ground station. Glasgow and Manchester have comparable weights and display similar performance characteristics. The key delivery curve of Bristol, which is the least significant ground station in the communication system considered, alternates swiftly. It is also the most southern location among the ground stations and the first one a satellite can send keys to during the pass at midnight.

The key delivery profiles presented in Figure 4.5 are rapidly changing, which complicates their analysis. Hence, the maximum weekly key consumption rate that can be steadily maintained without depleting the capacity of the buffer is computed instead. Figure 4.6 illustrates the relation between the maximum key consumption rate for London and the SL. The plot looks similar for all ground stations.

London could consume up to 9295 keys weekly without depleting its buffer capacity for 99% of weeks. If the key consumption rate exceeds that threshold, then the service level threshold will further decrease due to unmet demand for some additional weeks. Table 4.3 reports the maximum number of keys available for weekly consumption at the 99% SL for all ground stations.

Table 4.3: Maximum weekly key consumption maintainable at 99% Service Level by a given ground station.

| City | 99% SL |
|---|---:|
| Glasgow | 2498 |
| Manchester | 2318 |
| Birmingham | 3918 |
| London | 9295 |
| Bristol | 1939 |

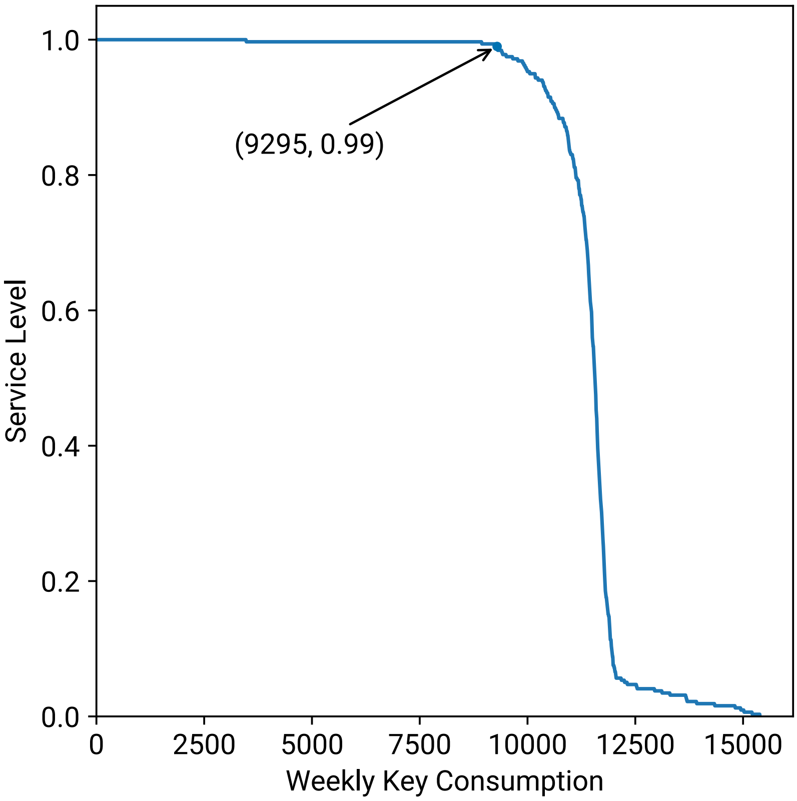


Figure 4.6: Maximum number of keys that can be consumed weekly by London ground station enforcing the SL guarantee. If the SL threshold $\alpha$ is lower than one, then the buffer capacity can be depleted for $(1 - \alpha)$ relative number of weeks. Plots for other ground stations are similar.

Naturally, the key consumption rate closely follows the importance of the ground station in the network. Specifically, Manchester and Glasgow have similar weights and consume a comparable number of keys.

## 4.6 Scheduling under Uncertainty using Official Weather Forecasts

The section compares the effectiveness and the computational effort of using the suite of the SatQKD formulations with the treatment of uncertainty developed in Chapter 3. It presented five alternative formulations. For the ease of tracking and referencing they are enumerated below along with the acronyms and the reference numbers assigned to them: Box Uncertainty Set Model (i.e., BOX, Formulation 3.10), Box Time Series Uncertainty Set Model (i.e., BTS, Formulation 3.12), Mean Standard Deviation Model (i.e., MSTD, Formulation 3.13), CV@R Model (i.e., CV@R, Formulation 3.20), and Essential Riskiness Index Model (i.e., ERI, Formulation 3.21). The suite of models is further extended by Formulation 3.1 in two variants: Deterministic Single Forecast Model (DSF) and Deterministic Multistage Model with Folding Horizon (DRH).

All formulations share some common parameters. The vector of weights $\omega$ was set according to Table 4.1. The number of keys possible to deliver in the cloud-free line of sight to a given ground station at a given time interval was computed using the model [20] assuming the initial orbital parameters of the spacecraft outlined in Section 4.5.1. The vector of cloud cover conditions $\mathbf{c}$ was derived from official weather forecasts published between October 2019 and February 2020 by [113]. Every forecast starts at noon and contains cloud cover predictions for the next 40 hours. The period of late autumn and winter months was selected because during that time frame communication windows between the satellite and ground stations depend only on the elevation angle and are not impacted by the illumination of the satellite.

The data set contains twenty-six problem instances. Each instance corresponds to a single weather forecast which predicts cloud cover conditions for the next five days. Overall, cloud cover predictions are not available for more extended periods, which seems to be related to the low accuracy of such forecasts. The performance of the optimal solution computed based on weather forecasts is assessed using weather conditions observed in the real world. Even though optimisation problems together constitute a continuous period, they are solved for every instance independently. Therefore, keys delivered to a given ground station throughout the scheduling horizon are not carried to the next scheduling horizon. As a result, a poor performance of a given model for one problem instance does not affect the solution for another problem instance.

Models built using the SO framework (i.e., Formulations 3.21 and 3.20) assume the discrete uniform distribution over a collection of 256 scenarios generated using the following procedure. For every historical weather forecast, a sequence of prediction errors is obtained by subtracting cloud cover forecasts from the weather conditions observed in the real world. Subsequent forecasts are separated by 12 hours to avoid temporal correlation between scenarios. Overall, prediction errors in the data set have zero mean and are not correlated. Then, the cloud cover conditions from the weather forecast for a given problem instance are added to every instance of the vector of prediction errors to create a set of alternative weather scenarios. Finally, predictions whose values are outside the interval $[0, 1]$ are rounded to the closest value, which remains in that range.

Chance Constraint in the CV@R model must be satisfied with probability at least 95%, i.e., the parameter $\epsilon$ is set to 0.05. Furthermore, the ERI and CV@R models require specification of the target scaling factor $\lambda'$, which depends on the problem instance. Intuitively, $\lambda'$ should be set to the largest value for which a given problem instance has a feasible solution. The target scaling factor is computed using the bisection algorithm in which the lower and the upper bound of the scaling factor are successively restricted, i.e., $\lambda' \in [\underline{\lambda}', \bar{\lambda}')$. In the first step, the deterministic formulation is solved to compute the initial scaling factor $\lambda^i$. Suppose the given problem instance is not feasible for the initial scaling factor. In that case, the upper bound is increased $\bar{\lambda}' = \lambda^i$. Otherwise, the lower bound is initialised $\underline{\lambda}' = \lambda^i$. After the first bound is set, the other is obtained by multiplying (dividing) $\lambda^i$ by 2 in the case of the upper (lower) bound. The initial bounds are passed to the bisection algorithm, which terminates after the distance between bounds reaches 1. Testing for feasibility in the bisection algorithm requires solving one problem instance per iteration for the given value of the target scaling factor until the first solution is found. Nonetheless, it could be a computationally challenging process, particularly when the problem instance is at the edge of infeasibility. For that reason, the computations are accelerated by increasing the discretisation step of communication windows to 30 seconds and by setting the computational time limit to 30 minutes per problem instance. If the time limit is not sufficient to find a solution to the problem instance, it is considered it infeasible.

Robust Optimisation and Distributionally Robust Optimisation formulations (i.e., BOX, BTS, MSTD) developed in Chapter 3 contain parameters which allow for controlling the conservativeness of the model. Such parameters were set to maximise the schedule performance assessed in real weather conditions using the subset of 20 problem instances selected in the chronological order.

Figure 1 illustrates the pseudocode of the algorithm for selecting the combination of parameters that maximise the performance of the formulation for a given suite of problem instances.

```
Parameters: Problem Formulation $\mathcal{F}$, Parameters ($\underline{\boldsymbol{p}}$, $\bar{\boldsymbol{p}}$, $\boldsymbol{s}$),
            Problem Instances $\mathcal{I}$
Result: Parameters with the best performance $\mathbf{p}$
$\mathbf{g} \leftarrow \mathrm{RANGE}(\underline{\mathrm{p}}_1, \bar{p}_1, s_1) \times \cdots \times \mathrm{RANGE}(\underline{\mathrm{p}}_n, \bar{p}_n, s_n)$
$\boldsymbol{r} \leftarrow []$
for $[p_1, \ldots, p_n] \in \boldsymbol{g}$ do
    $\lambda \leftarrow 0$
    for $I \in \mathcal{I}$ do
        $\lambda \leftarrow \lambda + \mathcal{F}([p_1, \ldots, p_n], I)$
    end
    $\boldsymbol{r} \leftarrow \mathbf{r} \mid (\lambda, [p_1, \ldots, p_n])$
end
$(\lambda, \boldsymbol{p}) \leftarrow max_{\lambda}(\mathbf{r})$
return $\boldsymbol{p}$
```

**Algorithm 1:** Hyperparameter tuning for models with the treatment of uncertainty.

The algorithm takes as the input the optimisation model ($\mathcal{F}$), vectors containing minimum, maximum values and the discretisation step for each configuration parameter in the model ($\underline{\boldsymbol{p}}$, $\bar{\boldsymbol{p}}$, $\boldsymbol{s}$), and the suite of problem instances $\mathcal{I}$.

The following operators are defined. The operator RANGE$(a, b, c)$ creates a vector containing all values between $a$ and $b$ generated with the incremental step $c$. The boundary values are included in the output vector. The operator $\boldsymbol{a}|\boldsymbol{b}$ appends the vector $\boldsymbol{b}$ to the vector $\boldsymbol{a}$ by stacking it vertically. The operator $\boldsymbol{a}\times\boldsymbol{b}$ creates a Cartesian product from the pair of input vectors. Finally, the operator $\max_b(\boldsymbol{a})$ given a multidimensional vector $\boldsymbol{a}$ and a parameter $b$ present in every row of the vector $\boldsymbol{a}$ returns the first element for which the parameter $b$ assumed the largest value.

The algorithm creates a grid $\mathbf{g}$, which stores every possible combination of configuration parameters. Then, for every point in the grid, all problem instances from the set $\mathcal{I}$ are solved using the formulation $\mathcal{F}$ initialised according to the configuration of parameters considered. Resultant traffic indices are summed to assess the overall performance on the dataset obtained using the given configuration. The algorithm returns the coordinates of the first point for which the sum of traffic indices attained the largest value. Overall, the grid $\mathbf{g}$ stores $\sum_{i=1}^{n} \left\lceil \frac{1}{s_i}(\overline{p}_i - \underline{p}_i) + 1 \right\rceil$ candidate configurations. Suppose $\boldsymbol{o}$ is the computational complexity of solving the formulation $\mathcal{F}$ for a problem instance. Hence, the computational complexity of the hyperparameter tuning procedure is $\sum_{i=1}^{n} \left\lceil \frac{1}{s_i}(\overline{p}_i - \underline{p}_i) + 1 \right\rceil |\mathcal{I}|\boldsymbol{o}$.

The size of the uncertainty set/support set is measured in standard deviations from the nominal value, i.e., $\tilde{\boldsymbol{u}} \in [\boldsymbol{u} - r \cdot \boldsymbol{\sigma}, \boldsymbol{u} + r \cdot \boldsymbol{\sigma}]$ given the nominal value $\boldsymbol{u}$, the multiplicative factor $r$, and the standard deviation $\boldsymbol{\sigma}$. Candidate values for the radius $r$ are sampled from the range between 0.05 until 4.0 with the step of 0.05 using Algorithm 1. For the uncertainty set $\mathcal{B}$ and the support set $\mathcal{W}$, the best performance was observed by setting the radius to 0.4. For the uncertainty set $\mathcal{V}$, the radius was reduced further to 0.3. On the other hand, the parameter $r$ in the DRO model was increased to 0.45. Residuals $\boldsymbol{e}$ in the BTS model were set to a value high enough to represent 99% of cloud cover prediction changes announced in weather forecasts. If no realisation of uncertainty fits the BTS model, the residual terms were multiplied by the smallest coefficient, which makes the uncertainty set non-empty.

### 4.6.1 Computational Results

Problem instances from the dataset were solved for every formulation introduced in Chapter 3. Table 4.4 presents the value of the absolute scaling factor $\lambda$. It was computed using weather conditions observed at the time the schedule found for the given problem instance would be executed. This information was not revealed to the solver, which relied only on weather forecasts. The advantage of using real weather conditions for performance evaluation is the lack of additional assumptions. Furthermore, the performance of the models recorded in the real world should be useful for practitioners as it closely imitates the intended application context.

Understandably, problem instances differ in the accuracy of weather forecasts and the cloud cover conditions observed in the real world. Such effects could be captured by reporting the Mean Cloud Cover (MCC) according to the initial weather forecast for every problem instance and the Mean Absolute Percentage Error (MAPE) between the cloud cover forecast and the conditions observed in the real world during communication windows. Naturally, the difference in the performance of various models of uncertainty will depend on weather conditions. For instance, during adverse weather conditions such as overcast, there will be little difference in the performance of the formulations due to limited opportunity for communication. The same applies to the discrepancies between weather forecasts and actual cloud cover conditions, as considerable prediction errors in weather forecasts increase the risk of making sub-optimal decisions by optimisation models.

The BTS formulation reported solutions with the largest accumulated scaling factor across all models considered. The MSTD model recorded similar performance for all problem instances. The deterministic model with the rolling horizon was ranked third before the ERI and CV@R formulations. Nonetheless, the last two models obtained better results for individual problem instances in which forecasts were inaccurate or when weather conditions were adverse (e.g., MAPE above 20 or MCC over 80%). Predictably, the BOX model was the most conservative across the formulations with the treatment of uncertainty. The deterministic model with a single weather forecast recorded the worst performance. The result is not surprising because all other models implemented a rolling horizon framework and therefore had the opportunity to reconsider decisions regarding the future using the most up to date weather forecasts.

Discussing the computational results expressed using the absolute scaling factor could give an advantage to models which perform well in good weather conditions, such as the clear sky. For a broader perspective, let normalise scaling factors by dividing them by the best scaling factor one could find for every problem instance. The schedule with such a scaling factor could be computed by solving the deterministic model given weather conditions observed in the real world. Henceforth, such a solution is called a perfect schedule.

Table 4.5 presents scaling factors for every problem instance normalised by dividing the result by the scaling factor of the perfect schedule.

The visualisation of the selected models in Figure 4.7 complements information

Table 4.4: Scaling factors for solutions computed using optimisation models developed in Chapter 3. Rows correspond to problem instances sorted in chronological order. Column MCC contains mean cloud cover according to the initial weather forecast. Column MAPE stores the mean absolute prediction error between cloud cover forecast and the real weather conditions observed during communication windows. The remaining columns are labelled using acronyms that represent optimisation models introduced in Chapter 3: Formulation 3.1 in two variants, Deterministic Single Forecast Model (DSF) and Deterministic Multistage Model with Folding Horizon (DRH), Box Uncertainty Set Model (i.e., BOX, Formulation 3.10), Box Time Series Uncertainty Set Model (i.e., BTS, 3.12), Mean Standard Deviation Model (i.e., MSTD, Formulation 3.13), CV@R Model (i.e., CV@R, Formulation 3.20), and Essential Riskiness Index Model (i.e., ERI, Formulation 3.21). The columns contain the scaling factors for a given problem instance estimated using cloud cover conditions observed in the real world. Scaling factors normalised by dividing by the best scaling factor for a given problem instance are presented in Table 4.5.

| | MCC | MAPE | DSF | DRH | BOX | BTS | MSTD | CV@R | ERI |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 76 | 17 | 552 | 4,522 | 8,475 | 10,896 | 7,600 | 4,719 | 9,396 |
| 2 | 68 | 25 | 4,817 | 13,161 | 7,495 | 13,296 | 13,710 | 11,691 | 12,821 |
| 3 | 73 | 20 | 1,136 | 4,592 | 2,890 | 4,045 | 3,347 | 1,940 | 4,861 |
| 4 | 46 | 22 | 2,885 | 8,297 | 11,499 | 7,890 | 8,228 | 15,781 | 9,296 |
| 5 | 73 | 19 | 5,864 | 11,856 | 6,413 | 13,616 | 14,151 | 5,429 | 13,674 |
| 6 | 21 | 48 | 660 | 696 | 696 | 689 | 696 | 696 | 696 |
| 7 | 85 | 19 | 1,993 | 2,630 | 3,368 | 2,630 | 2,630 | 3,181 | 2,630 |
| 8 | 53 | 23 | 12,486 | 5,458 | 12,508 | 8,508 | 8,794 | 8,891 | 3,036 |
| 9 | 84 | 20 | 3,509 | 3,613 | 8,011 | 3,613 | 4,411 | 4,857 | 5,522 |
| 10 | 75 | 17 | 6,288 | 8,089 | 8,043 | 8,049 | 8,110 | 5,954 | 8,297 |
| 11 | 93 | 8 | 327 | 327 | 327 | 327 | 327 | 327 | 327 |
| 12 | 73 | 10 | 10,544 | 12,793 | 9,056 | 12,717 | 12,756 | 8,762 | 10,929 |
| 13 | 32 | 16 | 22,451 | 29,178 | 13,832 | 28,729 | 28,860 | 24,417 | 19,206 |
| 14 | 83 | 19 | 3,245 | 7,195 | 7,208 | 8,212 | 7,200 | 6,360 | 6,504 |
| 15 | 67 | 19 | 1,573 | 5,204 | 6,095 | 7,499 | 6,160 | 7,239 | 6,908 |
| 16 | 73 | 15 | 1,899 | 408 | 1,064 | 563 | 550 | 1,729 | 1,057 |
| 17 | 61 | 29 | 3,019 | 8,808 | 8,153 | 8,199 | 7,903 | 9,037 | 6,030 |
| 18 | 92 | 16 | 331 | 3,129 | 1,515 | 3,938 | 3,559 | 3,563 | 1,423 |
| 19 | 65 | 24 | 1,127 | 1,715 | 2,189 | 1,513 | 1,921 | 1,173 | 1,766 |
| 20 | 83 | 18 | 2,684 | 5,048 | 2,955 | 4,755 | 5,888 | 4,714 | 9,052 |
| 21 | 80 | 8 | 3,373 | 4,402 | 1,854 | 4,059 | 4,166 | 3,321 | 5,343 |
| 22 | 59 | 21 | 4,022 | 3,716 | 5,276 | 3,715 | 4,004 | 6,342 | 5,965 |
| 23 | 65 | 12 | 15,231 | 14,254 | 6,263 | 13,015 | 13,433 | 5,502 | 8,225 |
| 24 | 82 | 16 | 1,615 | 6,490 | 6,330 | 6,000 | 6,018 | 7,440 | 5,802 |
| 25 | 87 | 9 | 589 | 1,674 | 1,631 | 925 | 1,301 | 615 | 2,158 |
| 26 | 54 | 15 | 12,258 | 17,466 | 10,628 | 18,290 | 17,419 | 17,638 | 12,494 |
| Total | - | - | 124,478 | 184,722 | 153,773 | 195,689 | 193,142 | 171,319 | 173,418 |

Table 4.5: Scaling factors for every problem instance computed using formulations derived in Chapter 3 and divided by the best performance a schedule could have for normalisation. Table 4.4 presents the results before normalisation.

| | MCC | MAPE | DSF | DRH | BOX | BTS | MSTD | CV@R | ERI |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 76.00 | 17.00 | 0.04 | 0.29 | 0.55 | 0.71 | 0.49 | 0.31 | 0.61 |
| 2 | 68.00 | 25.00 | 0.27 | 0.74 | 0.42 | 0.75 | 0.77 | 0.66 | 0.72 |
| 3 | 73.00 | 20.00 | 0.13 | 0.52 | 0.33 | 0.46 | 0.38 | 0.22 | 0.55 |
| 4 | 46.00 | 22.00 | 0.11 | 0.31 | 0.43 | 0.29 | 0.31 | 0.59 | 0.34 |
| 5 | 73.00 | 19.00 | 0.35 | 0.71 | 0.38 | 0.82 | 0.85 | 0.33 | 0.82 |
| 6 | 21.00 | 48.00 | 0.07 | 0.08 | 0.08 | 0.08 | 0.08 | 0.08 | 0.08 |
| 7 | 85.00 | 19.00 | 0.23 | 0.30 | 0.38 | 0.30 | 0.30 | 0.36 | 0.30 |
| 8 | 53.00 | 23.00 | 0.84 | 0.37 | 0.84 | 0.57 | 0.59 | 0.60 | 0.20 |
| 9 | 84.00 | 20.00 | 0.31 | 0.32 | 0.70 | 0.32 | 0.39 | 0.42 | 0.48 |
| 10 | 75.00 | 17.00 | 0.59 | 0.76 | 0.76 | 0.76 | 0.76 | 0.56 | 0.78 |
| 11 | 93.00 | 8.00 | 0.07 | 0.07 | 0.07 | 0.07 | 0.07 | 0.07 | 0.07 |
| 12 | 73.00 | 10.00 | 0.75 | 0.91 | 0.64 | 0.91 | 0.91 | 0.62 | 0.78 |
| 13 | 32.00 | 16.00 | 0.74 | 0.96 | 0.46 | 0.95 | 0.95 | 0.80 | 0.63 |
| 14 | 83.00 | 19.00 | 0.31 | 0.68 | 0.68 | 0.78 | 0.68 | 0.60 | 0.62 |
| 15 | 67.00 | 19.00 | 0.15 | 0.50 | 0.59 | 0.72 | 0.59 | 0.70 | 0.67 |
| 16 | 73.00 | 15.00 | 0.59 | 0.13 | 0.33 | 0.17 | 0.17 | 0.53 | 0.33 |
| 17 | 61.00 | 29.00 | 0.21 | 0.60 | 0.56 | 0.56 | 0.54 | 0.62 | 0.41 |
| 18 | 92.00 | 16.00 | 0.04 | 0.40 | 0.19 | 0.50 | 0.45 | 0.45 | 0.18 |
| 19 | 65.00 | 24.00 | 0.07 | 0.10 | 0.13 | 0.09 | 0.11 | 0.07 | 0.10 |
| 20 | 83.00 | 18.00 | 0.20 | 0.37 | 0.22 | 0.35 | 0.44 | 0.35 | 0.67 |
| 21 | 80.00 | 8.00 | 0.50 | 0.65 | 0.27 | 0.60 | 0.62 | 0.49 | 0.79 |
| 22 | 59.00 | 21.00 | 0.33 | 0.31 | 0.44 | 0.31 | 0.33 | 0.53 | 0.50 |
| 23 | 65.00 | 12.00 | 0.78 | 0.73 | 0.32 | 0.67 | 0.69 | 0.28 | 0.42 |
| 24 | 82.00 | 16.00 | 0.14 | 0.55 | 0.54 | 0.51 | 0.51 | 0.63 | 0.49 |
| 25 | 87.00 | 9.00 | 0.21 | 0.59 | 0.58 | 0.33 | 0.46 | 0.22 | 0.76 |
| 26 | 54.00 | 15.00 | 0.56 | 0.80 | 0.49 | 0.84 | 0.80 | 0.81 | 0.58 |
| Total | - | - | 8.58 | 12.76 | 11.37 | 13.40 | 13.24 | 11.90 | 12.89 |

presented in the table by grouping problem instances with similar weather conditions and comparable accuracy of weather forecasts.

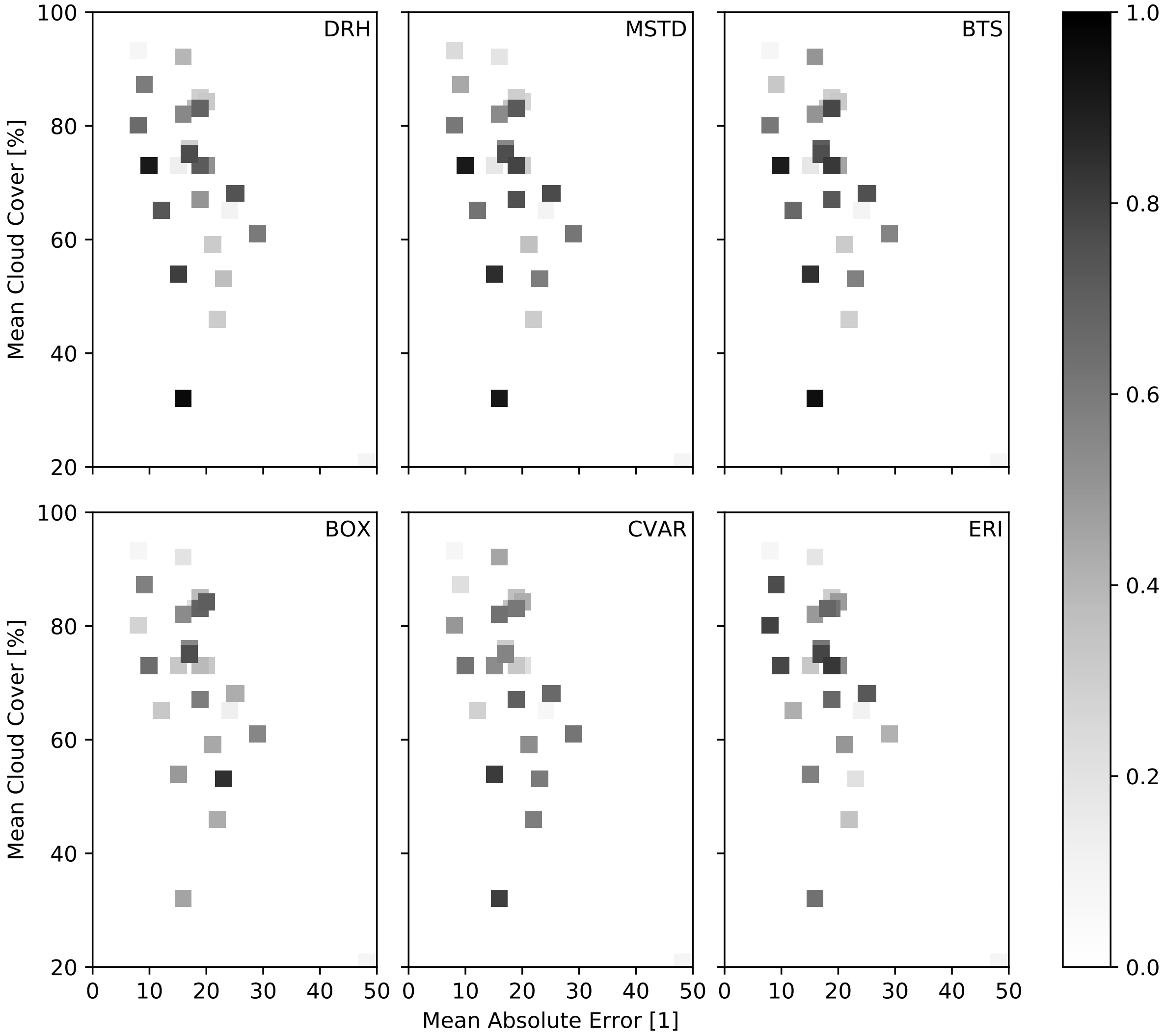


Figure 4.7: Performance of the selected formulations observed in the real-world and divided by the best performance a schedule could have. Exact values are provided in Table 4.5.

Similarly to the analysis of the absolute scaling factor, the best results were recorded for the BTS and the MSTD models. Interestingly, the ranking of the models in the next three positions differs from the previous comparison. The ERI formulation outperformed the deterministic model with a rolling horizon and the CV@R model. The BOX formulation closes the ranking of models which provide the treatment of uncertainty.

The deterministic single forecast model is the worst formulation in the comparison. Overall, the best score possible to achieve equals the number of problem instances in the data set. Nonetheless, the best model reached only half of that score. The situation seems to be the culprit of the low accuracy of weather forecasts. For instance, consider that only for 10 out of 26 instances, the deterministic model with a rolling horizon found a schedule whose scaling factor was at least 60% of the value of the perfect schedule. Only in specific circumstances, i.e., at most moderate MCC (below 80%) combined with small MAPE (below 20), DRH, BTS and MSTD models achieved the normalised scaling factor above 80%.

Results discussed previously suggest that different problem formulations exhibit Pareto efficient performance profiles. Therefore, the next aim is to find the circumstances that justify using a specific model in given weather conditions. Figure 4.8 illustrates models with the treatment of uncertainty which obtained the best scaling factor for a given problem instance. The latter are represented as points in 2D with MCC and MAPE as the dimensions. The colours of the markers distinguish between models.

Overall, in 10 out of 17 cases when weather forecast had high accuracy (i.e., the MAPE below 20%), the MSTD and the BTS were the best models with the treatment of uncertainty. Conversely, when the MAPE was over 20, schedules computed by either the BOX or the CV@R model performed best in 6 out of 7 problem instances. Although the MAPE is handy for explaining when given formulations yield satisfactory results, using the parameter for the model selection is impractical because the estimate of the prediction error is not available until real weather conditions are revealed. On the other hand, the policy definition can exploit information about predicted weather conditions, e.g., when the MCC is above 70%, then the ERI is the best model (i.e., in 6 out of 15 cases).

Let conclude the analysis above by constructing an ensemble model which combines the ERI and the MSTD formulations in the following manner. If the MCC of the initial weather forecast is less than 70%, then the schedule is computed using the MSTD model. Otherwise, the ERI model is selected. The study of Tables 4.4 and 4.5 indicates

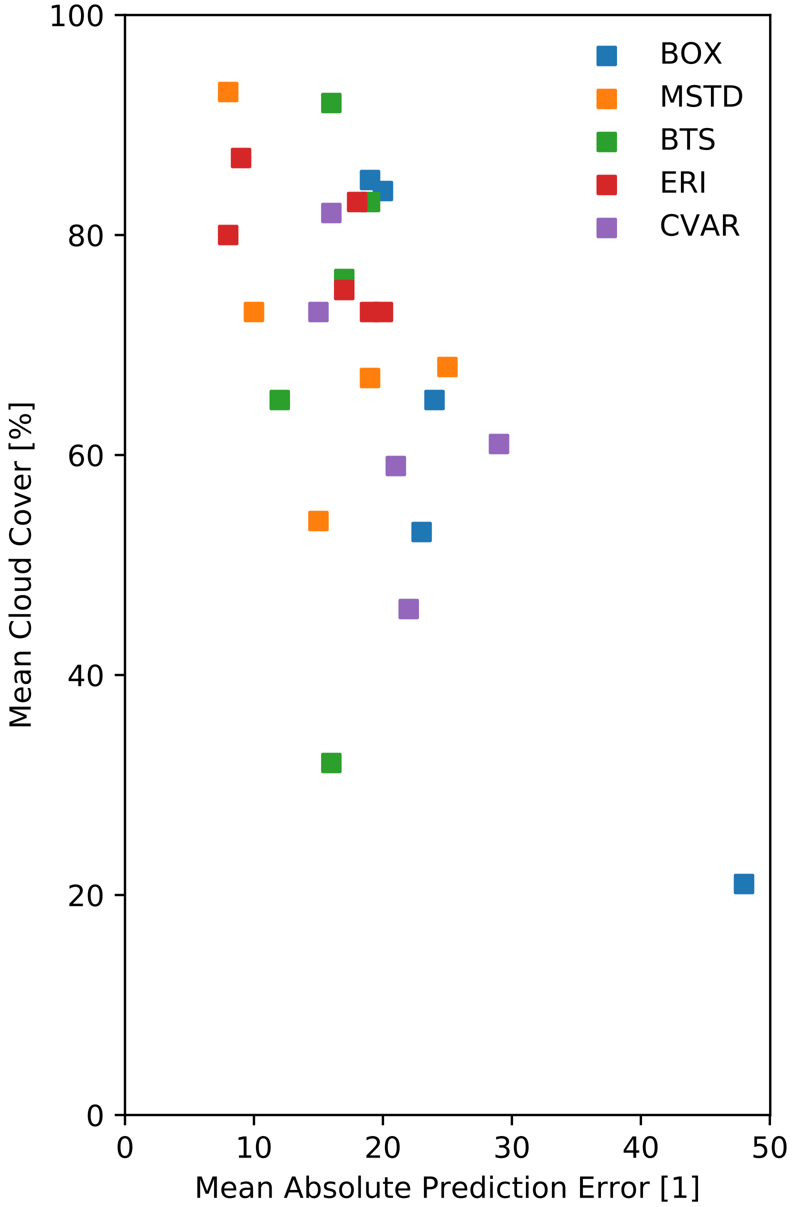


Figure 4.8: Models of uncertainty which found the best solution for a given problem instance. Data points are plotted on a 2D plane with the MCC of the initial weather forecast and the MAPE between the initial cloud cover forecast and the real cloud cover during communication windows. Table 4.5 presents complete results.

that the ensemble model would accumulate the absolute scaling factor of 198,317 and the normalised scaling factor of 14.06. These performance indicators outperform all stand-alone models in both benchmarks. Overall, the example demonstrated the merit of studying different models for handling uncertainty and led to the construction of an ensemble model that outperformed all stand-alone models.

### 4.6.2 Computational Time

Table 4.6 presents the average and the worst-case computational time for solving a problem instance for all models introduced in Chapter 3.

Table 4.6: Average and the worst-case computational time for solving a problem instance for models introduced in Chapter 3: Formulation 3.1 in two variants, Deterministic Single Forecast Model (DSF) and Deterministic Multistage Model with Folding Horizon (DRH), Box Uncertainty Set Model (i.e., BOX, Formulation 3.10), Box Time Series Uncertainty Set Model (i.e., BTS, 3.12), Mean Standard Deviation Model (i.e., MSTD, Formulation 3.13), CV@R Model (i.e., CV@R, Formulation 3.20), and Essential Riskiness Index Model (i.e., ERI, Formulation 3.21). Computational times for CV@R and ERI models do not include the estimation of the target scaling factor $\lambda'$.

| Method | Mean [s] | Max [s] |
|---|---:|---:|
| DSF | 0.40 | 2.80 |
| DRH | 0.90 | 3.10 |
| BOX | 0.60 | 3.20 |
| BTS | 4.00 | 18.90 |
| MSTD | 373.80 | 3,439.50 |
| CV@R | 2,979.20 | 19,096.30 |
| ERI | 119.40 | 563.90 |

Deterministic formulations and the model with the BOX uncertainty set required less than one second on average to solve a problem instance and up to 3.2 seconds in the worst case. The BTS model needed approximately six times more computational time. The DRO formulation solved using the adversarial approach and the SO models performing the optimisation over scenarios required an order of magnitude more time for computations. The most computationally challenging model in the comparison, the CV@R, needed 49 minutes on average to solve a problem instance and more than 4 hours in the worst case.

It is not surprising that models without the treatment of uncertainty completed in a fraction of the time required by more advanced formulations. Fortunately, the additional computational time is not excessive for some models with the treatment of uncertainty and is compensated by the performance improvement of the resultant schedules. In particular, the BTS model, which is the most effective stand-alone formulation considered in the study, requires computational time in the range of seconds. Taking into account that the BTS model has comparable performance to the MSTD formulation, the former seems a better choice considering its order of magnitude shorter computational time. On the other hand, the excessive computational time and inferior performance compared to alternative formulations make the CV@R model impractical. Due to its overall effectiveness, the ERI does not seem to be the first choice for a stand-alone formulation. However, its computational time is below 6 minutes in the worst case, and the ERI remains still viable as a secondary formulation if an ensemble model is considered.

### Model Scalability

Computational time depends on the size of the formulation. It is controlled by the length of the discretisation step for communication windows, i.e., the smaller the discretisation step, the more binary variables in the formulation. The discretisation step is adjusted for the BTS model, which turned out to return the most effective schedules for the dataset of problem instances considered. The aim of the analysis is to study the relation between the size of the model and the computational effort required to solve it.

Table 4.7 displays descriptive statistics of computational time for the given length of the discretisation step.

Computational time increases with the resolution of the discretisation scheme for communication windows. Nonetheless, the model remains practically solvable for all problem instances, even for the discretisation step of one second, i.e., the longest computational time is 8 minutes. Regarding the computational time and the size of the discretisation scheme, similar observations were confirmed by [97]. The same author

Table 4.7: Average and the worst-case computational time required by the BTS model to prove optimality given the time discretisation of communication windows.

| Scheme [s] | Mean [s] | Max [s] |
|---:|---:|---:|
| 30 | 1.30 | 2.90 |
| 15 | 3.20 | 12.80 |
| 10 | 5.30 | 23.20 |
| 5 | 13.60 | 62.80 |
| 1 | 155.80 | 516.30 |

showed that the computational time grows linearly with the number of satellites when communication windows are generated randomly.

## Concluding Remarks

Computational results discussed in this chapter demonstrated the validity of the methodology developed for scheduling space-to-ground data transfers. The scheduling problems were solved using exact methods in attractive computational times.

The case study demonstrated the importance of using optimisation in designing SatQKD communication systems. The primary contribution is the evaluation of the number of keys that can be delivered to the network of ground stations. Firstly, the configuration that maximised the number of keys delivered to the communication system was found by computing optimal schedules for various orbital parameters of the satellite. Then, for each ground station, the number of keys that can be consumed weekly without exceeding the capacity of the buffer subject to probabilistic service level guarantees was computed. To the best of the author's knowledge, such a study has not been published before. However, it is conceivable that a similar upfront performance analysis would be indispensable for building commercial-grade Quantum Communication systems in the future.

Besides estimating the performance of the communication system, the influence of weather seasonality and spacecraft's illumination by the Sun on the number of keys delivered to the network of ground stations was also studied. A vital observation is that the opportunity for communication between the satellite and ground stations located at high latitudes is severely restricted for several weeks during the summer.

Furthermore, regarding scheduling satellite data transfers, it was the first study that used official weather forecasts to build optimisation models with the treatment of uncertainty and evaluated the performance of the formulations in real weather conditions. Such an approach was adopted deliberately to accurately reflect the application context in which the optimisation models are going to be used.

The study demonstrated that several models with the treatment of uncertainty and a rolling horizon significantly outperform the deterministic model. Out of the formulations developed, the BTS and MSTD models recorded the best performance for the whole suite of problem instances.

The BTS model is a novel formulation proposed first in Chapter 3. Its uncertainty set describes cloud cover conditions preserving temporal and spatial correlations, which is a desirable property. The model can be reformulated into a MILP and effectively solved using an off-the-shelf commercial MIP solver.

Overall, no single model with the treatment of uncertainty equivocally dominated other formulations. By analysing the results for individual problem instances, it was shown that some models, i.e., ERI, recorded a better performance in adverse weather conditions. This observation led to the creation of an ensemble model which combined the MSTD and the ERI formulations. The new model outperformed all stand-alone formulations on the set of problem instances considered. A similar technique has not been applied in the literature on optimisation under uncertainty. Consequently, the research questions such as "which models of uncertainty complement each other" has not yet been explored.

The next two chapters move from solving pure scheduling problems to combined routing and scheduling optimisation. The exact methods employed here for scheduling satellite communication are not applicable for solving large problem instances in the vehicle routing domain (i.e., more than 150 visits). Consequently, a new solution methodology that is based on CP, LS and metaheuristics was developed.

# Chapter 5

# Methodology: Vehicle Routing Problem with Side Constraints

## 5.1 Introduction

Fast transport of goods and people is critical for the modern economy. Besides technological improvements that increase the speed and throughput of the transportation networks, the cost of transport remains a suitable material for routing optimisation. One of the most famous optimisation problems formulated for this application domain is the Traveling Salesman Problem (TSP) [141]. Its objective is to find the shortest closed tour that visits a set of destinations exactly once and returns to the origin. This chapter considers an extension of the TSP problem, which involves routing a fleet of vehicles to perform visits in geographically distributed locations known as the Vehicle Routing Problem (VRP) [142]. Each visit is mandatory and must be executed without pre-emption by a vehicle that spends some prescribed duration at the visit's location. Vehicles start their routes from depots, where they must return following the last visit. The most common objective is the minimisation of the total travel time for all vehicles.

The VRP is a generic class of problems. Notable representatives of this group are the Capacitated Vehicle Routing Problem (CVRP), the Vehicle Routing Problem with Time Windows (VRPTW), and the Vehicle Routing Problem with Pickup and Delivery (VRPPD). The fleet of vehicles in the CVRP transports goods from a central depot to customers subject to the maximum cargo load limit for each vehicle. Visits in the VRPTW [143] must commence within some prescribed time interval. If a vehicle arrives at the visit's location too early, it must wait until the time window begins. Arrival after the time window ends either is disallowed or subject to a penalty, depending on the problem definition. Finally, the VRPPD coordinates vehicles that transport goods or people between multiple destinations.

This thesis considers a further specialisation of the VRPTW in which a subset of visits requires a simultaneous presence of two vehicles. The problem is known as the Vehicle Routing Problem with Time Windows and Synchronized Visits (VRPTWSync) [144]. The synchronisation constraint is restricted to two vehicles only. Such requirements arise in the delivery applications and the HHC, e.g., to deliver or to pick up a heavy object, to transport a person with reduced mobility, etc. Problems that involve the coordination of more than two vehicles to perform a visit are outside the scope. They are known as Technician Routing and Scheduling (TRS) [145] and Manpower Allocation Problem (MAP) [146].

A synchronisation of visits is a special case of a general temporal dependency. Other common types of temporal dependencies are restrictions on the minimum and the maximum period between consecutive visits. Scheduling visits constrained by arbitrary temporal dependencies is the subject of the Vehicle Routing Problem with Time Windows and Temporal Dependencies (VRPTWTD) formulated by [19, 147].

### Structure of the Chapter

Section 5.2 presents the MIP formulation of the VRPSC problem. The formulation is suitable for solving small problem instances (e.g., 25 visits) to provable optimality. Section 5.3 develops an equivalent CP formulation. The following section presents an extension of the CP formulation performing optimisation accounting for an uncertain duration of visits. Large problem instances (e.g., scheduling 500 visits using 60 vehicles) can be solved using either variant of the CP formulation by the multistage optimisation algorithm described in Section 5.5. Some concluding remarks close the chapter.

## Shared Symbols

Before presenting the MIP and the CP models of the VRPSC problem, the shared symbols and conventions adopted in both formulations are introduced below.

### Visits

The set $V$ contains requested visits. For each visit $v$, its duration is $d_v$ and the time window $[\underline{s}_v, \overline{s}_v]$ defines when the visit should commence.

### Vehicles

The set $C$ represents the fleet of vehicles. Every vehicle $c$ is initially located at the depot $b_c$. Following the last visit, the vehicle $c$ must return to the depot $e_c$. Without loss of generality, multiple depots could be assigned to vehicles on an individual basis. For simplicity, the start and the end depots are treated as the first and the last visit on a route the vehicle follows. The time window associated with both depots overlaps with the availability of the vehicle for work, i.e., $[\underline{s}_{b_c}, \overline{s}_{b_c}] = [\underline{s}_{e_c}, \overline{s}_{e_c}]$. A vehicle is not required to stop at the depot, i.e., the visit duration at both depots is zero.

### Synchronised Visits

Some visits may require two vehicles. The execution of such a visit cannot commence until the requested vehicles arrive. A visit with synchronisation constraints is represented by two elements in the set of visits $V$. Either both elements are assigned different vehicles or together are left unassigned.

The set $V$ is split into disjoint subsets $V = V^1 \cup V^2$. For the given number $i \in \{1, 2\}$, the subset $V^i$ contains elements representing visits that require dispatching exactly $i$ vehicles. Furthermore, for each subset $V^i$, let define the partition $\Pi^i$ composed of subsets which contain the elements that refer to the same visit.

### Contractual Breaks

A vehicle could be entitled to contractual breaks during working hours. The set $B$ contains all breaks regardless of the vehicle. Breaks for a given vehicle $c$ are stored in the subset $B^c$. Each break $b$ has the time window $[\underline{s}_b, \overline{s}_b]$ when it must commence and the duration $d_b$ it lasts. All breaks are mandatory. Similarly to visits, their execution cannot be interrupted, and the vehicle that takes a break must remain at the same location. For conciseness, the time outside of working hours is also modelled as breaks. The break representing the time before working hours has a fixed time window of zero length, i.e., the start and the end of the time window are equal. The time after working hours could have a time window and a flexible duration depending on the overtime allowance for a given vehicle.

### Skill-based Routing

Some vehicles may not be allowed to perform a given visit, i.e., due to skill constraints, client's preferences, etc. Therefore, for each visit $v$, let define the set $C^v$ that contains permitted vehicles.

### Visit Continuity

In circumstances where several visits are performed for a given client (a.k.a. user), it is common to have a policy that fosters dispatching the same vehicles for consecutive visits of a given user. Let $U$ be the set of users who are being visited. The set $V^u$ stores visits of the user $u$. For each user $u$, let define a limit $\overline{f}_u$ that restricts the maximum number of different vehicles which can visit the user during the scheduling horizon.

### Optional Visits

In real-world applications, demand for visits could exceed the capacity of the system, i.e., due to emergency visits being reported or vehicles being temporarily unavailable. This phenomenon is known as oversubscription. If it happens and all visits are mandatory, the problem instance becomes infeasible. A pragmatic way of dealing with this situation is making visits optional subject to the penalty $\rho$ for each declined visit. Understandably, whether a given visit is effectively cancelled, postponed, or outsourced depends on the application. For simplicity, let assume any visit can be declined. The penalty is charged once for each declined visit regardless of the number of vehicles it requires. For the sake of convenience, declined visits are assigned to the special vehicle $c_{\emptyset}$.

### Network Representation

The locations between which vehicles can travel are encoded using a directed graph $G(N, E)$. The set $N$ represents vertices (a.k.a. nodes), and $E$ is the set of edges. If a vehicle can travel from one node to the other, then they are connected by an edge. The set $N$ contains depots and visits. It may also include breaks, which depends on the formulation.

Let introduce the following operators and symbols to implement the navigation in the graph. Given a node $n$, the operator $\delta^{+}(n)$ returns the set of nodes that can be reached directly from the node $n$ by traversing one edge. Conversely, the operator $\delta^{-}(n)$ returns the set of nodes from which any vehicle could arrive at the node $n$ by traversing one edge. The set $N^{b}$ is the subset of $N$ whose elements have at least one outgoing edge. Conversely, the set $N^{e}$ contains nodes that have at least one incoming edge. Understandably, visiting a node and traversing an edge requires time. The function $\delta : N^{b} \times N^{e} \rightarrow \mathbb{Z}_{\geq 0}$ computes the time needed for travel between the first and the second node passed as the input. If an edge does not connect both nodes, then the distance is infinite. Finally, the function $\tau(n_i, n_j) = d_{n_i} + \delta(n_i, n_j)$ returns the time required to visit the node $n_i$ and to arrive at the node $n_j$.

## 5.2 Mixed-Integer Programming Model

The MIP formulation presented in this section is based on the three-index multi-commodity network flow formulation for the VRPTW [148]. The synchronisation constraints for visits that require two vehicles were proposed by [19]. The handling of contractual breaks was inspired by [149] who applied it to model one lunch break per vehicle. It is extended further in the thesis to support an arbitrary number of breaks per vehicle.

### Symbols and Decision Variables

The set of routing nodes $N$ in the MIP formulation besides visits and depots includes breaks, i.e., $N = V \cup \bigcup_{c \in C} \{b_c, e_c\} \cup B$. Since the location of a vehicle does not change after it takes a break, the following rules are adopted to navigate in the graph:

1. A vehicle can move to another visit or the end depot only from a visit or the start depot,
2. A vehicle must return to the visit or the start depot which preceded the break or a sequence of consecutive breaks,
3. Travel time is zero if the source or the destination is a break.

Let introduce the following variables. The three-index formulation derived its name from defining a binary variable for every triple: a vehicle, a source node, and a target node. Intuitively, the binary variable $x_{n_b,n_e,c}$ is set to one if the vehicle $c \in C$ traverses the edge $(n_b, n_e) \in E$, otherwise the variable is set to zero. Furthermore, for each routing node $n \in N$, let define a continuous variable $s_n$. Its interpretation depends on the type of a node. If $n$ is a visit or a break, then the variable $s_n$ stores the time when the execution of the visit or the break commences. If $n$ represents a begin depot, then the variable is the departure time of the vehicle. Otherwise, the variable is the arrival time of the vehicle to the end depot. For each break $b \in B$, let introduce the integer variable $r_b$ that stores the node where the vehicle must return to restore its previous location before the break, or a sequence of consecutive breaks was taken. Furthermore, for each pair of the vehicle $c$ and the user $u$, the binary variable $f_{c,u}$ indicates whether the vehicle performs at least one visit for the user within the scheduling horizon. Finally, for each visit $v$, an auxiliary binary variable $a_v$ indicates whether the given visit is staffed or declined.

### Formulation

The following formulation builds upon the prerequisites introduced above.

$$\min \sum_{c \in C} \sum_{n_i \in N^b} \sum_{n_j \in N^e} \delta(n_i, n_j) x_{n_i,n_j,c} + \rho \sum_{v \in V^1} (1 - a_v) + \frac{\rho}{2} \sum_{v \in V^2} (1 - a_v) \tag{5.1}$$

s.t.:

$$\sum_{n \in \delta^+(b_c) \cap N} x_{b_c,n,c} = 1 \qquad \forall c \in C \tag{5.2}$$

$$\sum_{n \in \delta^-(e_c) \cap N} x_{n,e_c,c} = 1 \qquad \forall c \in C \tag{5.3}$$

$$\sum_{c \in C} \sum_{n_i \in \delta^-(n_j) \cap N} x_{n_i,n_j,c} \leq 1 \qquad \forall n_j \in N^e \tag{5.4}$$

$$\sum_{n \in \delta^-(b)} x_{n,b,c} \leq 1 \qquad \forall b \in B^c \quad \forall c \in C \tag{5.5}$$

$$x_{n,v,c} = 0 \qquad \forall n \in N \cup B^c \quad \forall v \in V \quad \forall c \in C : v \notin V^c \tag{5.6}$$

$$x_{v,n,c} = 0 \qquad \forall n \in N \cup B^c \quad \forall v \in V \quad \forall c \in C : v \notin V^c \tag{5.7}$$

$$\sum_{n_j \in \delta^+(n_i)} x_{n_i,n_j,c} - \sum_{n_j \in \delta^-(n_i)} x_{n_j,n_i,c} = 0 \qquad \forall n_i \in N \cup B^c \quad \forall c \in C \tag{5.8}$$

$$\sum_{c \in C} \sum_{b \in \delta^+(n) \cap B^c} x_{n,b,c} \leq 1 \qquad n \in N^b \tag{5.9}$$

$$\sum_{b \in \delta^+(b_c) \cap B^c} x_{b_c,b,c} = 1 \qquad \forall c \in C \tag{5.10}$$

$$\sum_{b \in \delta^+(e_c) \cap B^c} x_{e_c,b,c} = 1 \qquad \forall c \in C \tag{5.11}$$

$$\sum_{n \in \delta^-(b)} x_{n,b,c} = 1 \qquad \forall b \in B^c \quad \forall c \in C \tag{5.12}$$

$$x_{n_j,b,c} - \sum_{n_i \in \delta^-(n_j) \cap N} x_{n_i,n_j,c} \leq 0 \quad \forall n_j \in N \quad \forall b \in B^c \quad \forall c \in C \tag{5.13}$$

$$x_{b,n_j,c} - \sum_{n_i \in \delta^-(n_j) \cap N} x_{n_i,n_j,c} \leq 0 \quad \forall n_j \in N \quad \forall b \in B^c \quad \forall c \in C \tag{5.14}$$

$$\sum_{b \in \delta^+(n) \cap B^c} x_{n,b,c} - \sum_{b \in \delta^-(n) \cap B^c} x_{b,n,c} = 0 \quad \forall n \in N \quad \forall c \in C \tag{5.15}$$

$$r_b \leq (i+1) x_{n_i,b,c} + M(1 - x_{n_i,b,c}) \quad \forall n_i \in N \quad \forall b \in B^c \quad \forall c \in C \tag{5.16}$$

$$(j+1) x_{b,n_j,c} \leq r_b + M(1 - x_{b,n_j,c}) \quad \forall n_j \in N \quad \forall b \in B^c \quad \forall c \in C \tag{5.17}$$

$$r_{b_j} \leq r_{b_i} + M(1 - x_{b_i,b_j,c}) \quad \forall b_i, b_j \in B^c \quad \forall c \in C \tag{5.18}$$

$$s_{n_i} + \tau(n_i, n_j) \leq s_{n_j} + M(1 - x_{n_i,n_j,c}) \quad \forall n_i, n_j \in N \quad \forall c \in C \tag{5.19}$$

$$s_{b_i} + d_{b_i} \leq s_{b_j} + M(1 - x_{b_i,b_j,c}) \quad \forall b_i, b_j \in B^c \quad \forall c \in C \tag{5.20}$$

$$s_b + d_b \leq s_n + M(1 - x_{b,n,c}) \quad \forall n \in N \quad \forall b \in B^c \quad \forall c \in C \tag{5.21}$$

$$s_{n_i} + \tau(n_i, n_j) \leq s_b + M(2 - x_{n_i,n_j,c} - x_{n_j,b,c}) \quad \forall n_i, n_j \in N \quad \forall b \in B^c \quad \forall c \in C \tag{5.22}$$

$$\sum_{n \in \delta^-(v)} x_{n,v,c} \leq M f_{c,u} \quad \forall v \in V^u \quad \forall u \in U \quad \forall c \in C \tag{5.23}$$

$$\sum_{c \in C} f_{c,u} \leq \overline{f}_u \quad \forall u \in U \tag{5.24}$$

$$a_{v_j} - \sum_{c \in C} \sum_{v_i \in \delta^-(v_j)} x_{v_i,v_j,c} = 0 \quad \forall v_j \in V \tag{5.25}$$

$$s_{v_i} = s_{v_j} \quad \forall (v_i, v_j) \in \Pi^2 \tag{5.26}$$

$$a_{v_i} = a_{v_j} \quad \forall (v_i, v_j) \in \Pi^2 \tag{5.27}$$

$$\sum_{v_h \in V \setminus \{v_i, v_j\}} \left\{ x_{v_h,v_i,c} + x_{v_h,v_j,c} \right\} \leq 1 \quad \forall (v_i, v_j) \in \Pi^2 \quad \forall c \in C \tag{5.28}$$

$$\sum_{v_h \in \delta^-(v_j) \cap V} x_{v_h,v_j,c_y} \leq 1 - \sum_{v_h \in \delta^-(v_i) \cap V} x_{v_h,v_i,c_x} \quad \forall (v_i, v_j) \in \Pi^2 : i < j \quad \forall c_x, c_y \in C : x < y \tag{5.29}$$

$$\underline{s}_v \leq s_v \leq \overline{s}_v \quad \forall v \in V \tag{5.30}$$

$$\underline{b} \leq s_b \leq \overline{b} \quad \forall b \in B^c \quad \forall c \in C \tag{5.31}$$

$$\mathbf{s} \in \mathbb{R}_{\geq 0}^{|V \cup B|}, \quad \mathbf{r} \in \mathbb{R}_{0 \leq |V|}^{|B|}$$
$$\mathbf{x} \in \{0,1\}^{|N \cup B| \times |C|}, \quad \mathbf{a} \in \{0,1\}^{|V|}, \quad \mathbf{f} \in \{0,1\}^{|C| \times |U|} \tag{5.32}$$

The goal of the optimisation problem is to minimise Objective Function (5.1) defined as the sum of the total travel time for all vehicles and penalties for each declined visit.

Besides the constraints explained below, the three-index binary variables are set to zero if any of the following conditions defined for nodes (a.k.a. endpoints) connected by an edge holds:

1. Both endpoints are the same node,
2. The source endpoint is a visit, and the target endpoint is the start depot,
3. The source endpoint is the end depot, and the target endpoint is either a visit or the start depot,
4. Time windows indicate that the target endpoint must be visited before the source endpoint.

Constraints (5.2 - 5.18) define the network flow. Every vehicle must leave the start depot and eventually arrive to the end depot due to Constraints (5.2) and (5.3), respectively. Constraints (5.4) and (5.5) enforce that each node representing a visit or a break is visited by at most one vehicle. Constraints (5.6) and (5.7) implement skill-based routing, i.e., they prevent dispatching a vehicle to a visit if the vehicle is not allowed. Constraint (5.8) enforces the flow conservation.

A vehicle visiting any node can move to at most one break by Constraint (5.9). Before leaving the start depot, the vehicle must take the break that represents the time before working hours due to Constraint (5.10). The analogous rule for the end depot and the time after working hours is implemented by Constraint (5.11). Every break is mandatory and must be taken by its dedicated vehicle due to Constraint (5.12). If a vehicle moves to a break node from a visit, then Constraint (5.13) requires that the vehicle must have arrived at the visit from some other node. Constraint (5.14) enforces a similar condition if the vehicle returns to the visit from a break node. Constraint (5.15) strengthens the flow conservation for break nodes. Constraints (5.16-5.18) employ a big constant customarily denoted as $M$. Collectively, the constraints enforce that a vehicle which moved to a break node from a visit will return to the same visit either immediately or after a sequence of consecutive breaks.

Time propagation is defined by Constraints (5.19-5.22). Specifically, let assume that if a vehicle moves to a break node from a visit, the break node is executed before the visit due to Constraint (5.22). The visit is commenced after the break or a sequence of consecutive breaks completes by Constraint (5.21).

Constraints (5.23) and (5.24) implement the continuity of visits. Constraint (5.23) updates the value of the binary variables that indicate whether a user is visited at least once by a given vehicle. Constraint (5.24) limits the number of different vehicles that may visit the user.

Constraint (5.25) controls the status of variables that indicate whether a given visit is scheduled.

Constraints (5.26) and (5.27) provide support for visits that require two vehicles. If such a visit is scheduled, then two vehicles must be dispatched by Constraint (5.27). Moreover, both vehicles commence the visit at the same time due to Constraint (5.26). Constraints (5.28) and (5.29) are defined to strengthen the formulation in the face of the value symmetry introduced by visits with synchronisation constraints. Constraint (5.28) states that if a vehicle arrives at a node that represents a visit that requires two vehicles, the other node which points to the same visit must be visited by another vehicle. Constraint (5.29) enforces lexicographic ordering in the assignment of vehicles to nodes that refer to the same visit.

Constraints (5.30) and (5.31) define time windows for visits and breaks, respectively. Finally, Constraint (5.32) declares types and domains of decision variables.

Ultimately, the formulation above can be solved to provable optimality using an off-the-shelf MIP solver. Nonetheless, it is well-known that the formulation has a weak lower bound of the LP relaxation. This phenomenon is due to the big-M constraints which define the time propagation. Consequently, only small and medium problem instances, i.e., between 20-30 visits can be solved to optimality. The observation was reproduced in preliminary results obtained using the formulation above. For larger problem instances, one could solve the set-partitioning formulation using column generation [150]. Nonetheless, problem instances with 500 visits arise in the real world, which is well beyond the scalability limits of the exact methods. Consequently, the approach in the thesis follows the path paved by many researchers and adopts a modest but achievable goal to solve large problem instances using heuristic methods. They are applied to the alternative CP formulation explained in the next section.

## 5.3 Constraint Programming Model

The CP formulation presented in this section is built on top of the generic model for the Vehicle Routing Problem with Time Windows and Breaks implemented in the or-tools library [17]. The original formulation is extended in the thesis by adding support for visits with synchronisation constraints, skill-based routing, and the continuity of visits.

### Global Constraints

Let first present the global constraints that build the formulation. All constraints accept as input the vector of integer variables $\mathbf{y}$ which encode routes in the following manner. The vector $\mathbf{y}$ has the length $|C| + |V|$. The first $|C|$ elements of the vector represent the start depots for each vehicle. The remaining elements correspond to visits. The element $y_i$ stores the index of the node succeeding the node $i$ on the route. Since a route terminates at the end depot, there is no need to store its successor. Consequently, assigning a successor which is out of bounds of the vector $\mathbf{y}$ terminates the route. If a given element is assigned its index, e.g., $y_i = i$, then the node is not traversed on any route.

no_cycle($\mathbf{y}$) - Routes encoded by the vector $\mathbf{y}$ contain no cycles except for self-loops.

all_different($\mathbf{y}$) - Every variable in the vector $\mathbf{y}$ is assigned a different value.

at_most($\mathbf{z}, v, n$) - At most $n$ variables in the vector $\mathbf{z}$ are assigned value $v$.

member($z, S$) - Integer variable $z$ is assigned an element from the set $S$.

path_cumul($\mathbf{y}, \mathbf{z}, \delta$) - An arbitrary resource is acquired or consumed by vehicles along the routes they follow according to the function $\delta$. The vector $\mathbf{z}$ stores the amount of the resource held by a vehicle at a given node. The function $\delta : [0, \dots, |\mathbf{y}| - 1] \times [0, \dots, |\mathbf{y}| - 1] \to \mathbb{Z}$ returns the marginal change (a.k.a. delta) in the amount of the resource due to the visit of a given node and travel to its successor.

breaks($\mathbf{y}, \mathbf{z}, \delta, B$) - Vehicles must have enough idle time between travelling and performing visits to schedule the mandatory breaks from the set $B$.

The breaks constraint is a custom global constraint implemented in the vehicle routing library of the or-tools framework [17]. Contrary to the breaks constraint, all the remaining global constraints are widely known and supported by the CP solvers.

### Symbols and Decision Variables

Let extend the notation and symbols introduced in Section 5.2.

The operator bool($\cdot$) applied to an arbitrary logical expression converts it into a binary variable. Furthermore, let define a generic lookup operator $\iota_n(\dots)$ parametrised using a non-negative integer $n$. The operator accepts an arbitrary number of input arguments (a.k.a. indefinite arity). Given an input vector or a list of input arguments, the operator returns the n-th element of the sequence.

The set of nodes $N$ in the CP formulation consists of visits and depots, i.e., $N = V \cup \bigcup_{c \in C} \{b_c, e_c\}$. Contrary to the MIP formulation, it does not include contractual breaks. Moreover, in contrast to the three-index formulation, the routes are represented using a one-dimensional vector of integer variables $\mathbf{y}$ and the encoding scheme explained in the definition of global constraints. The vector of constants $\mathbf{d}$ stores visits' duration and the vector of variables $\mathbf{s}$ keeps track of visits' start times. Furthermore, let introduce a vector of integer variables $\mathbf{w}$ to determine which vehicle is visiting a given node. If a visit $v$ is declined, then $w_v$ is assigned to the value which corresponds to the special vehicle $c_\emptyset$.

### Formulation

Let introduce the CP formulation.

$$\min \sum_{n \in N^b} \delta(n, y_n) + \rho \sum_{v \in V^1} (1 - a_v) + \frac{1}{2}\rho \sum_{v \in V^2} (1 - a_v) \tag{5.33}$$

$$\text{s.t.:} \quad \text{all\_different}(\mathbf{y}) \tag{5.34}$$

$$\text{no\_cycle}(\mathbf{y}) \tag{5.35}$$

$$\text{path\_cumul}(\mathbf{y}, \mathbf{s}, \tau) \tag{5.36}$$

$$\text{path\_cumul}(\mathbf{y}, \mathbf{w}, \iota_2) \tag{5.37}$$

$$\text{breaks}(\mathbf{y}, \mathbf{s}, \mathbf{d}, \delta, b_c, e_c, B^c) \qquad \forall c \in C \tag{5.38}$$

$$a_v = \text{bool}(y_v \in N^e \setminus \{v\}) \qquad \forall v \in V \tag{5.39}$$

$$a_v = \text{bool}(w_v \neq c_\emptyset) \qquad \forall v \in V \tag{5.40}$$

$$y_v = v \Leftrightarrow w_v = c_\emptyset \qquad \forall v \in V \tag{5.41}$$

$$\text{member}(w_v, C^v) \qquad \forall v \in V \tag{5.42}$$

$$a_{v_i} = a_{v_j} \qquad \forall (v_i, v_j) \in \Pi^2 \tag{5.43}$$

$$s_{v_i} = s_{v_j} \qquad \forall (v_i, v_j) \in \Pi^2 \tag{5.44}$$

$$w_{v_i} \leq w_{v_j} \qquad \forall (v_i, v_j) \in \Pi^2 \tag{5.45}$$

$$\sum_{v \in V^u} \text{bool}(w_v = c) \leq |V^u| f_{c,u} \qquad \forall c \in C \quad \forall u \in U \tag{5.46}$$

$$\sum_{c \in C} f_{c,u} \leq \overline{f}_u \qquad \forall u \in U \tag{5.47}$$

$$\underline{s}_n \leq s_n \leq \overline{s}_n \qquad \forall n \in N \tag{5.48}$$

$$\begin{gathered} \mathbf{a} \in \mathbb{B}^{|V|}, \quad \mathbf{f} \in \mathbb{B}^{|C| \times |U|} \\ \mathbf{y} \in \mathbb{Z}^{|N|}_{0 \leq |N|-1}, \quad \mathbf{s} \in \mathbb{Z}^{|N|}_{\geq 0}, \quad \mathbf{w} \in \mathbb{Z}^{|N|}_{0 \leq |C|-1} \end{gathered} \tag{5.49}$$

Identically to the MIP formulation, the objective is to minimise the sum of the total travel time for all vehicles and penalties for declined visits.

Constraint (5.34) enforces every node to be visited at most once. Routes must not have cycles which connect two or more nodes due to Constraint (5.35). Time is propagated by Constraint (5.36). Interestingly, Constraint (5.37) which tracks visiting vehicles uses the same global constraint definition but with a different vector of variables and the resource acquisition function. Constraint (5.38) is responsible for scheduling mandatory breaks. Constraints (5.39) and (5.40) update the variables which indicate whether a visit is scheduled or declined. Constraint (5.41) enforces declined visits to be assigned to the special vehicle $c_{\emptyset}$.

The skill-based routing is implemented using Constraint (5.42). Support for visits which require two vehicles is offered by Constraints (5.43-5.45). Constraint (5.43) enforces that such a visit is staffed by two vehicles. Both of them must be available simultaneously to commence the visit by Constraint (5.44). Constraint (5.45) strengthens the formulation by breaking the value symmetry and enforcing a lexicographic assignment of vehicles. Constraints (5.46) and (5.47) provide the continuity of visits. The binary variable $f_{c,u}$ is set if the user $u$ is visited by the vehicle $c$ by Constraint (5.46). The upper limit on the maximum number of different vehicles visiting a given user is enforced by Constraint (5.47). The start time of the visit is restricted to the prescribed time window due to Constraint (5.48). Finally, the types and domains of decision variables are declared in Constraint (5.49).

The CP formulation above assumes that visits' duration is known precisely. In the following section, the assumption is dropped, and uncertain visits' duration is modelled using a set of scenarios. Fortunately, moving to the more extended formulation will not require changing any of the constraints defined already.

## 5.4 Riskiness Index Optimisation in Constraint Programming

This section describes a CP formulation to minimise the delay in commencing visits after their time windows. Such a situation is forbidden in the deterministic variant of the VRPSC problem where the time windows must be respected in the hard sense. On the other hand, in the riskiness index optimisation considered here, it is sufficient that the expected start time of a visit obeys the time window restrictions. The pivotal property that motivates applying the riskiness index optimisation is enforcing larger delays to occur less likely. Out of the riskiness indices presented in Chapter 2, the ERI was selected as the decision criterion due to an efficient procedure for computing its value which is explained later in the section. Regarding alternative definitions of riskiness indices, a similar algorithm could be applied to calculating the SFI using its equivalent formulation derived by [16][Theorem 1]. On the other hand, it is not self-evident how to develop an evaluation procedure for the RVI without solving an optimisation problem, which is a computational bottleneck.

### Formulation

The set $H$ is the sample of scenarios. Each scenario is a collection of visits' durations recorded someday in the past. Let $\mathcal{F}$ be a feasible set which contains the assignments of variables that satisfy Constraints (5.33 - 5.49). Constraints involving visits' duration, i.e., Constraints 5.36 and 5.38, are defined once for the entire set of scenarios by taking the average duration for each visit. Hence, there is no Constraint 5.48 enforcing time window restrictions on visits' start times in the hard sense for each scenario. Finally, the variable $\rho$ stores the value of the riskiness index.

$$\min \rho \tag{5.50}$$

$$\text{s.t.:} \qquad (\mathbf{y}, \mathbf{s}, \mathbf{w}, \mathbf{a}) \in \mathcal{F} \tag{5.51}$$

$$\rho \geq \mathrm{ERI}(\mathbf{y}, \mathbf{s}, \delta, B, H) \tag{5.52}$$

$$\rho \in \mathbb{R}_{\geq 0} \tag{5.53}$$

The objective of the problem is to minimise the largest riskiness index of a delay for all visits. Constraint (5.51) restricts assignments of the decision variables to the feasible solutions of the deterministic CP formulation presented in Section 5.3. Constraint (5.52) is a custom global constraint implemented in the thesis to compute the largest riskiness index of a delay. If its value is finite, then the expected visits' start times obey time window restrictions. Finally, Constraint (5.53) declares the variable representing the riskiness index in the formulation.

### Essential Riskiness Index Global Constraint

The evaluation of the ERI is split into two steps:

1. Compute the delay in commencing a visit for each scenario,
2. Evaluate the ERI for the vector of delays computed in the previous step.

The following subsections provide a detailed description of each step.

#### Computation of the Delay in Commencing a Visit

The delay in commencing a visit cannot be calculated for each route independently due to synchronisation constraints, e.g., the first vehicle ready to begin a visit which requires two vehicles, has to wait until the other vehicle arrives.

This section presents an efficient algorithm for computing visits' start times that requires processing each visit exactly once. The algorithm represents data using a directed graph whose vertices correspond to performing visits, taking breaks, and visiting depots, whereas edges encode temporal precedence dependencies between nodes. Such graphs are known as Program Evaluation Review Technique (PERT) networks, i.e., $G(N = V \cup B \cup \bigcup_{c=1}^{|C|} \{b_c, e_c\}, E)$. A correctly defined PERT network does not contain cycles. The algorithm relies on the invariant that the start time of a given visit node is computed based on the start time of the preceding node on the same route and the start time of the sibling visit node on another route in the case of a pairwise synchronisation. If the synchronisation constraint is not defined for a given visit node, then the visit's start time cannot change after it has been computed. Otherwise, if the synchronisation constraint is present, then the node's start time is established after both sibling nodes have been processed.

Suppose, $a$, $b$, and $b'$ are visit nodes. The node $a$ directly precedes the node $b$. Furthermore, let assume that visit nodes $b$ and $b'$ must commence simultaneously. The start time at the node $b$ is computed according to the following formula.

$$s_b = max\{s_a + d_a + \delta(a,b), s_{b'}, \underline{s}_b\} \tag{5.54}$$

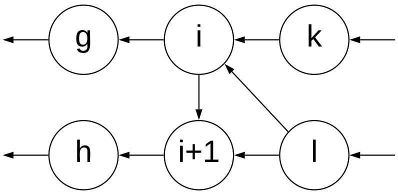
Consequently, the start time of the visit node $b$ is bounded from below by the arrival time at the visit node ($s_a + d_a + \delta(a,b)$), the start time of the sibling node ($s_{b'}$) and the start of its time window ($\underline{s}_b$). After the resultant start time is computed, the start time of the sibling visit node is updated, i.e., $s_{b'} = s_b$. The presentation of the example in which the synchronisation constraint is not defined is omitted because it naturally follows from the case discussed above.

The algorithm must process visit nodes in a specific order to obey its invariants. They are enforced by precedence dependencies defined according to the following approach.

Figure 5.1 presets a fragment of a PERT network which illustrates how precedence dependencies are defined between nodes. Suppose, the sequences of nodes $(g)-(i)-(k)$ and $(h)-(i+1)-(l)$ belong to different routes. Nodes $(i)$ and $(i+1)$ represent a visit with a synchronisation constraint. Besides edges connecting nodes on the same route, modelling the time propagation for a visit with a synchronisation constraint requires two additional precedence dependencies: $(i \rightarrow i+1)$ and $(l \rightarrow i)$. The configuration ensures that the start time for the node $(i+1)$ is computed before the node $(i)$. Moreover, the successor nodes $(l)$ and $(k)$ are not processed until the final start time for their preceding nodes is set. It is not required to have a precedence dependency $(k \rightarrow i+1)$ because the start time of the node $(i+1)$ is computed before the node $(i)$ which precedes the node $(k)$, i.e., $(k \rightarrow i)$.

Given the set of routes encoded as $\mathbf{y}$, the aim is to find a permutation $p$ that stores the nodes (i.e., visits, breaks, and depots) in the order compliant with precedence constraints. For instance, if two elements at positions $i$ and $j$ in the permutation $p$ refer to different visits and $i < j$, then the start time of the node $p_j$ has no impact on the node $p_i$. The permutation $p$ can be found by a topological sort of the graph $G$. Overall, it is a well-known procedure for which several algorithms have been proposed.

Figure 5.1: Handling visits with a synchronisation constraint in a PERT network.



A popular implementation employs traversal of all edges of the graph using depth-first search. It has complexity $O(|N| + |E|)$. Since the topological sort depends only on the set of routes $\mathbf{y}$, it is sufficient to compute it once and use the same order to calculate the start time for all visits and all scenarios. Taken together, given the set of scenarios $H$, the computation of the delay for all visits and all scenarios has the complexity $O(|N| + |E| + |N||H|)$.

**Computation of the Essential Riskiness Index for the Vector of Delays**

The riskiness index optimisation is integrated with a CP solver using a new global constraint. It evaluates the ERI for a given vector of delays for a selected visit. The pseudocode of the global constraint is presented in Algorithm 2.

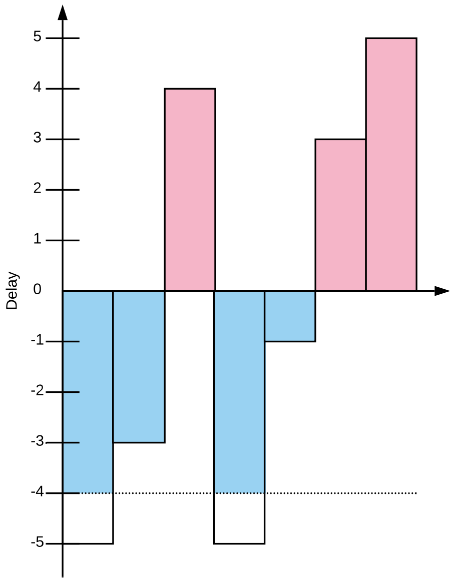
```
Parameters: $\mathbf{y}$, $\mathbf{s}$, $\mathbf{d}$, $\rho$, $\delta$, $C$, $V$, $B$, $H$
foreach $c \in C$ do
    if not path_connected($\mathbf{y}$, $\mathbf{s}$, $\mathbf{d}$, $\delta$, $b_c$, $e_c$) then
        return
    end
end
$\mathbf{e} \leftarrow$ compute_delays($\mathbf{y}, \mathbf{s}, \delta, C, V, B, H$)
$\rho_{\max} \leftarrow 0$
foreach $v \in V$ do
    $\rho_{\max} \leftarrow \max\{\rho, \text{compute\_ERI}(e_v)\}$
end
if lb($\rho$) $\leq \rho$ then
    add_constraint($\rho \geq \rho_{\max}$)
end
```

**Algorithm 2:** Global Constraint for the Essential Riskiness Index evaluation.

The constraint takes as the input: the assignment of decision variables ($\mathbf{y}$ and $\mathbf{s}$), the duration of visits in the nominal scenario ($\mathbf{d}$), the riskiness index variable ($\rho$), the distance function ($\delta$), and the details of the problem instance ($C$, $V$, $B$, $H$). Intuitively, global constraints should handle situations in which some of the variables are left unassigned. Hence, if some of the variables encoding routes are not set, then the constraint is not propagated. Otherwise, the delays are computed in a batch for all visits and all scenarios processing nodes according to the topological sort described in the previous subsection. Then, for each visit, the riskiness index is calculated for the vector of delays. If the riskiness index evaluated for the given visit is greater than the current lower bound of the riskiness index variable, its bound increases accordingly.

The algorithm for computing the ERI was developed by studying its analytical interpretation illustrated in Figure 5.2.

Figure 5.2: Analytical interpretation of the ERI.



The histogram presents example delays in commencing a hypothetical visit observed in the past. Every bar above the horizontal axis denotes a positive delay, i.e., a visit commenced after its time window ended. Conversely, a bar below the horizontal axis represents slack between the start time of the visit and the end of its time window. It is referred to in the sequel as a negative delay. Overall, the total positive delay filled using red colour is 12, i.e., $(4 + 3 + 5)$. The area is balanced out by the portion of the negative delay marked using a blue colour between the horizontal axis and the dotted line with the vertical intercept −4. The ERI is defined as the absolute value of the intercept. If the total negative delay does not compensate for the positive delay, then the ERI is infinite. Supported by the analytical interpretation explained above, let propose Algorithm 3 which computes the ERI for the given delay vector **e**.

```
Parameters: e
sort(e)
if $e_0 \geq 0$ then
    return int_max
end
$p \leftarrow$ length(e) $- 1$
if $e_p \leq 0$ then
    return 0
end
surplus $\leftarrow 0$
for ; $p \geq 0$ and $v_p \geq 0$ ; $p \leftarrow p - 1$ do
    surplus $\leftarrow$ surplus $+ e_p$
end
if $p == -1$ then
    return int_max
end
budget $\leftarrow 0$
for ; $p > 0$ and budget $+(p+1) * e_p +$ surplus $> 0$; $p \leftarrow p - 1$ do
    budget $\leftarrow$ budget $+ e_p$
end
balance $\leftarrow$ budget $+(p+1) * e_p +$ total_delay
if balance $> 0$ then
    return int_max
end
if balance $< 0$ then
    $\rho \leftarrow \min\{0, e_{p+1}\}$
    remaining_balance $\leftarrow$ total_delay $+$ budget $+(p+1) * \rho$
    $\rho \leftarrow \rho - \lceil$remaining_balance $/(p+1)\rceil$
    return $-\rho$
end
return $v_p$
```

**Algorithm 3:** Compute the Essential Riskiness Index.

The algorithm starts by sorting the vector of delays in ascending order and testing for corner cases. If the lowest delay in the vector is positive, then the riskiness index is infinite. Conversely, if the maximum delay is non-positive, the start time of the visit is compliant with the time window for all scenarios, and the riskiness index is zero. Otherwise, the total positive delay is computed, and the pointer $i$ is set to the element of the vector that stores the smallest negative delay. Then, in every iteration, the intercept is set to the peak value of the preceding histogram bar. This process continues until the area between the horizontal axis and the intercept balances the total positive delay. If the intercept is set to the most negative delay, and the area still does not compensate for the total positive delay, then the riskiness index is infinite. Otherwise, if the area between the horizontal axis and the intercept exceeds the accumulated positive delay, then the intercept is increased accordingly to find the balance.

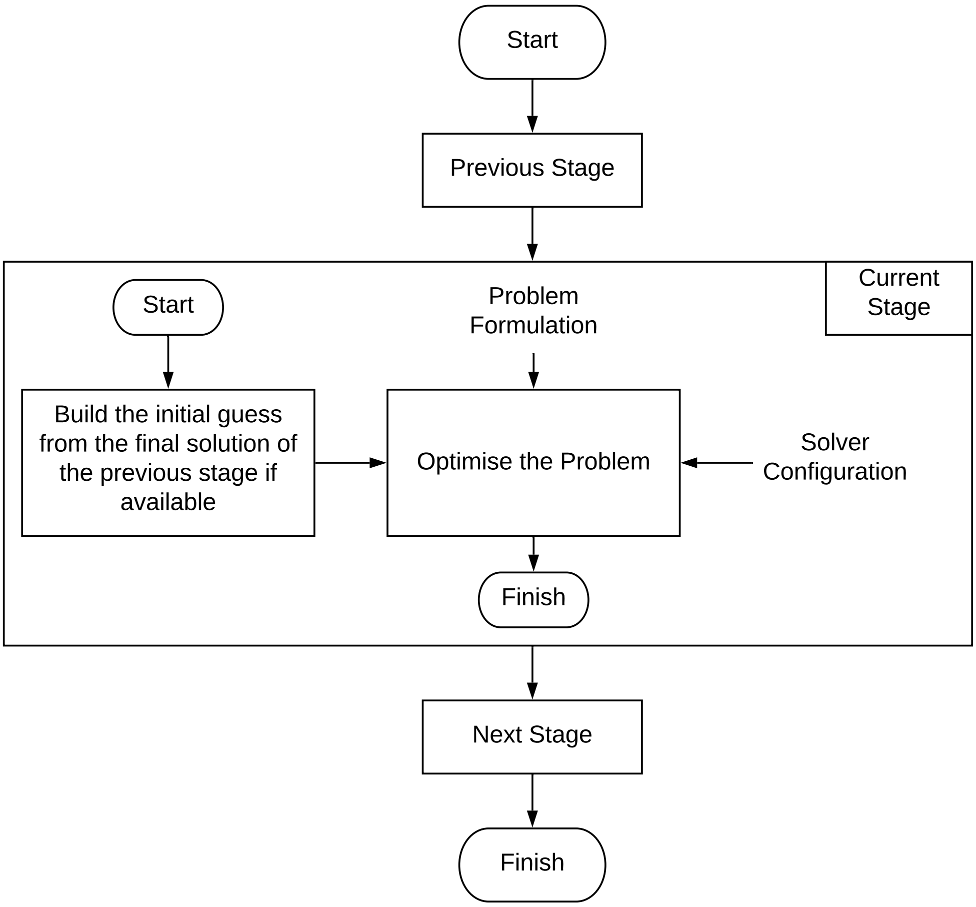
Besides sorting the elements of the vector, which has complexity $O(nlog(n))$ given the input vector of the length $n$, the evaluation of the ERI requires one pass through the content of the vector. Therefore, the presented algorithm has complexity $O(nlog(n))$.

Preliminary computational experiments with solving large problem instances containing visits with pairwise synchronisation (e.g., 450-500 visits in which 100 requires two vehicles) using either CP formulation directly is challenging for the solver. Specifically, the solver struggles to schedule more than 60-70% of all visits. The pairwise synchronisation of visits turned out to be the culprit. The computational issue motivated the development of the multi-stage optimisation algorithm, which is the subject of the next section.

## 5.5 Multi-Stage Optimisation Algorithm

A multistage algorithm is defined in the thesis as a hybrid method composed of at least two stages that sequentially solve an optimisation problem formulated using the same set of variables. Figure 5.3 presents the outline of the computation method. The first stage finds a solution that is passed to the second stage. Such an integration pattern has been widely used. Its typical example is employing a construction heuristic or CP in the first stage and a meta-heuristic in the second stage [151–153]. Furthermore, each stage could use a different formulation of the objective function. For instance, the total travel time minimised in the first stage can become a constraint in the next stage, which aims to maximise the continuity of visits [154]. Besides the problem formulation, the solver configuration may vary between stages. Reasonably, it is useful to restrict the time spent on optimising a solution at an intermediate stage which is likely to change in the following stage. Therefore, initial stages are configured to find a solution quickly, expecting that the final stage will improve it. The term multistage algorithm could be easily confused with a multiphase algorithm. The former term refers to a decomposition scheme that operates on different sets of variables. For instance, vehicles could be assigned to visits before routes with minimal travel time are computed [155, 156].

Figure 5.3: Control flow of the multistage optimisation algorithm.



Although there is no apparent limit on the number of stages, each involves solving an optimisation problem and contributes to the overall computational time. Therefore, the number of stages in the algorithm implemented in the thesis was limited to three, with several alternative versions of the final stage.

Every stage solves the problem instance using a highly configurable hybrid method implemented in the routing library of the or-tools framework [17]. Before explaining its pseudocode and the implementation details of each stage of the multistage optimisation algorithm, let introduce the following symbols and operators.

$C$ - set of vehicles,

$P$ - solution pool,

$S$ - set of search operators,

$V$ - set of visits,

$\mathbf{y}$ - vector of variables encoding a solution,

$\Delta$ - limit on the maximum time without improvement of the incumbent solution,

ADD$(P, \mathbf{y})$ - adds the solution $\mathbf{y}$ to the pool $P$,

ADD$(Q, \mathbf{v}, c)$ - adds the element $\mathbf{v}$ with the cost $c$ to the priority queue $Q$,

BEST$(P)$ - returns the best solution stored in the pool $P$,

ELAPSED-TIME() - returns the time since the start of the optimisation,

FIRST$(n)$ - given a routing node $n$ which represents a visit with a pairwise synchronisation constraint in the first stage, the function returns the first element of the pair of visits whose start time is bounded by the synchronisation constraint. For simplicity, the same function is used for the mapping between vehicles in the first and the second stage.

IS-FEASIBLE$(\mathbf{y})$ - tests the feasibility of the solution $\mathbf{y}$,

NEIGHBOURHOOD$(s, \mathbf{y})$ - returns the neighbourhood of the solution $\mathbf{y}$ induced by the search operator $s$,

NEXT$(P)$ - returns a solution from the pool $P$ to commence search, OBJ($\mathbf{y}$) - evaluates the objective of the solution $\mathbf{y}$,

OPTIMISE($F(V, C), \mathbf{y}, S$) - runs the optimisation procedure for solving the formulation $F(V, C)$ starting from the initial guess $\mathbf{y}$ using the set of search operators $S$. The optimisation method is implemented in the or-tools framework [17] and its pseudocode is explained in Algorithm 4.

POP($Q$) - removes the element at the top from the priority queue $Q$,

SECOND($n$) - given a routing node $n$ in the first stage, the function returns the second element of the pair of visits bounded by a synchronisation constraint. For simplicity, the same function is used for the mapping between vehicles in the first and the second stage.

SORT-DESC($\mathbf{v}, \theta$) - sorts the elements of the container (i.e., a set or a vector) $\mathbf{v}$ according to the key function $\theta(\cdot)$ in descending order,

$\omega(C)$ - returns the total time all vehicles in the set $C$ can work together,

$\sigma(C)$ - returns the number of skills shared by all vehicles passed in the input vector.

Algorithm 4 presents the pseudocode of the optimisation method adapted for solving minimisation problems. The hybrid method consists of a heuristic initialisation, followed by LS and LNS combined with constraint propagation. The LS operators employed by the multi-stage algorithm are explained in the following section.

The hybrid method commences search from an initial feasible solution provided as input. The incumbent solution is iteratively improved by exploring candidate solutions in neighbourhoods induced by search operators. The search process resembles a VNS without perturbation. Candidate solutions generated by a given search operator do not have to be fully initialised, which simplifies the implementation. Unassigned variables are set by the CP solver before testing the feasibility of the solution and evaluating its cost. The variant of the solution method presented above accepts candidate solutions as the new incumbent only if they strictly improve the objective. The or-tools framework [17] allows for replacing the strategy with a meta-heuristic, such as the TS or

```
Data: Initial feasible solution $\mathbf{y}^{(i)}$
Parameters: Problem formulation, Search operators S, Solution pool $P$
ADD($P$, $\mathbf{y}^{(i)}$)
$\mathbf{y}^{(s)}, \mathbf{y}^{(n)} \leftarrow \emptyset, \mathbf{y}^{(i)}$
$t \leftarrow 0$
repeat
    $\mathbf{x}^{(s)} \leftarrow \mathbf{x}^{(n)}$
    for $s \in S$ do
        $Y \leftarrow$ NEIGHBOURHOOD($s, \mathbf{y}^{(s)}$)
        while $Y \neq \emptyset$ do
            for $\mathbf{y}^{(c)} \in Y$ do
                if ELAPSED-TIME() $- t \geq \Delta$ then
                    return BEST($P$)
                end
                if IS-FEASIBLE($\mathbf{y}^{(c)}$) and OBJ($\mathbf{y}^{(c)}$) $<$ OBJ(BEST($P$))) then
                    ADD($P, \mathbf{y}^{(c)}$)
                    $t \leftarrow$ ELAPSED-TIME()
                    $Y \leftarrow$ NEIGHBOURHOOD($s, \mathbf{y}^{(s)}$)
                    break
                end
            end
        end
    end
    $\mathbf{y}^{(n)} \leftarrow$ NEXT($P$)
until $\mathbf{y}^{(s)} = \mathbf{y}^{(n)}$
return BEST($P$)
```

**Algorithm 4:** Optimisation method for solving VRPs implemented in the or-tools framework [17]. The algorithm is customised to solve minimisation problems. The maximum time with no improvement to the incumbent solution is the stopping criterion.

the GLS. Moreover, besides altering the stopping criteria, users can provide a custom implementation of the solution pool, which stores only the current best solution. The algorithm terminates after the stopping criteria are met, i.e., the limit on the number of iterations or when no search operator can improve the incumbent solution. The latter criterion is binding only when no meta-heuristic is used.

The following section explains the LS operators employed by the multi-stage algorithm.

### 5.5.1 Local Search Operators

The multi-stage algorithm employs a dozen of LS operators that were implemented by the authors of the or-tools framework [17]. All LS operators are simple transformations applied to a path or a collection of paths that reduce the total cost of traversing visited nodes. The search operators process variables $\mathbf{y}$ representing paths. All remaining variables, i.e., start times, are inferred by the CP solver.

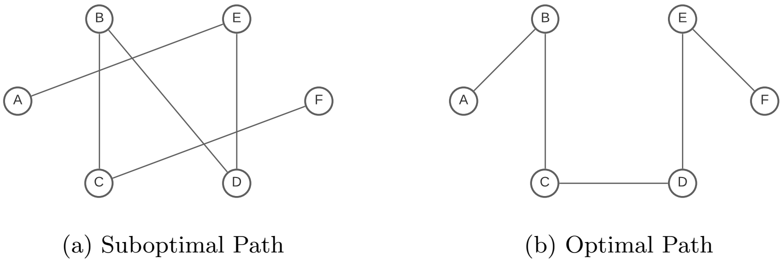
LS operators are presented by describing a single step they apply to reduce the total cost of traversing the set of six nodes: $A$, $B$, $C$, $D$, $E$ and $F$ located in a fully connected graph. The focus here is on the clarity of exposition rather than algorithmic details of each LS implementation. Paths must cover all nodes of the graph. The majority of LS operators are processing a single path in isolation. The path must commence at the node $A$ and terminate at the node $F$ in the presented example. If a LS operator is processing two paths simultaneously, the other path begins and terminates at nodes $B$ and $E$, respectively.

The left-hand side of Figure 5.4 presents an example suboptimal path that covers all nodes of the graph. It is used as the starting port for the LS operators processing a single path. The goal of the LS operators is to remove some edges from the suboptimal path and place new edges, so no path segments are disconnected. Removed edges are painted with red, and newly inserted edges are marked with green colour. The right-hand side of Figure 5.4 displays an optimal path. Even though paths are oriented, the direction is not displayed on edges to limit the number of details presented in the figure and simplify the exposition. The direction of traversal is straightforward as the path always starts from the node $A$.

The cost of traversal is presented in Table 5.1 for every pair of nodes. The matrix is symmetric and was computed according to the Euclidean distance.

#### Local Search Operators for a Single Path

The LS operators presented in this section restrict the admissible transformations to edges located on the same path. Some of the operators are well known due to research on heuristics for building initial tours for solving the TSP problem.



(a) Suboptimal Path

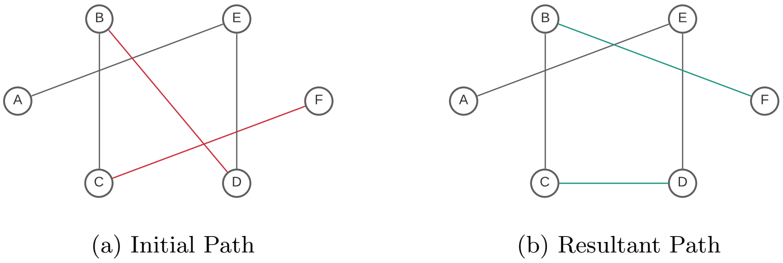
(b) Optimal Path

Figure 5.4: Example Paths

Table 5.1: Distance matrix of the example graph.

| Node | A | B | C | D | E | F |
|---|---|---|---|---|---|---|
| A | - | $\sqrt{2}$ | $\sqrt{2}$ | $\sqrt{10}$ | $\sqrt{10}$ | 4 |
| B | $\sqrt{2}$ | - | 2 | $2\sqrt{2}$ | 2 | $\sqrt{10}$ |
| C | $\sqrt{2}$ | 2 | - | 2 | $2\sqrt{2}$ | $\sqrt{10}$ |
| D | $\sqrt{10}$ | $2\sqrt{2}$ | 2 | - | 2 | $\sqrt{2}$ |
| E | $\sqrt{10}$ | 2 | $2\sqrt{2}$ | 2 | - | $\sqrt{2}$ |
| F | 4 | $\sqrt{10}$ | $\sqrt{10}$ | $\sqrt{2}$ | $\sqrt{2}$ | - |

Figure 5.5 illustrates the Two-Opt operator [157]. It removes two arbitrary edges from a path. Such an action could create up to three disconnected segments of nodes, which must be connected back to form a new path. There is only one way to achieve that, assuming the initial and the final path must be different, which simplifies the complexity of the operator.



(a) Initial Path

(b) Resultant Path

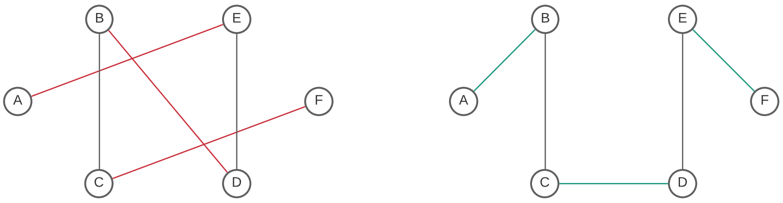
Figure 5.5: Example Move: Two-Opt

Edges $B \to D$, $C \to F$ are removed and edges $B \to F$, $C \to D$ are inserted in the Two-Opt move presented in the figure. The resultant path is suboptimal. One more

Two-Opt step is required to obtain the optimal solution.

Removing some prescribed number of edges and reconnecting path segments into a single path is an algorithm for a generic LS operator. Nevertheless, the number of possibilities the segments could be connected to form a new path grows exponentially with the number of edges involved. Processing up to three edges is widely acknowledged as a sensible trade-off between the computational complexity and the effectiveness of LS operators defined in this matter.

Figure 5.6 presents an example move performed by the Three-Opt operator [158] which removes three arbitrary edges from a path. Seven possible resultant paths could be obtained by inserting edges to reconnect the path segments assuming that repeating the initial combination of edges is not allowed.



(a) Initial Path

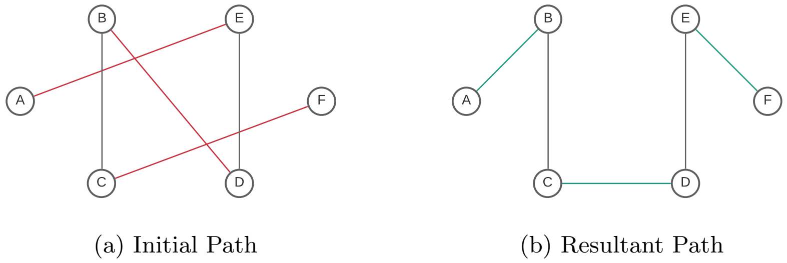
(b) Resultant Path

Figure 5.6: Example Move: Three-Opt

The Three-Opt operator removed the most expensive edges $A \rightarrow E$, $C \rightarrow F$ and $B \rightarrow D$ in the presented figure. Subsequently, cheaper edges $A \rightarrow B$, $C \rightarrow D$ and $E \rightarrow F$ were inserted. The resultant path $A \rightarrow B \rightarrow C \rightarrow D \rightarrow E \rightarrow F$ is optimal.

The need to consider seven alternative paths in a single Three-Opt move poses an implementation hurdle as performance is vital for the LS. Researchers suggested to impose additional restrictions on the Three-Opt that could reduce the computational complexity and retain high-quality paths the operator generates. A successful example of such an attempt is the Or-Opt operator [159]. It requires two of the three edges removed from the path to be located in close proximity controlled by a configuration parameter, i.e., separated by no more than three edges. Consequently, the Or-Opt moves either a node or a path segment up to three edges long to a new position on the path. The node or the path segment are inserted between nodes connected by the third removed edge.

Figure 5.7 presents an example Or-Opt move applied to the initial path.



(a) Initial Path

(b) Resultant Path

Figure 5.7: Example Move: Or-Opt

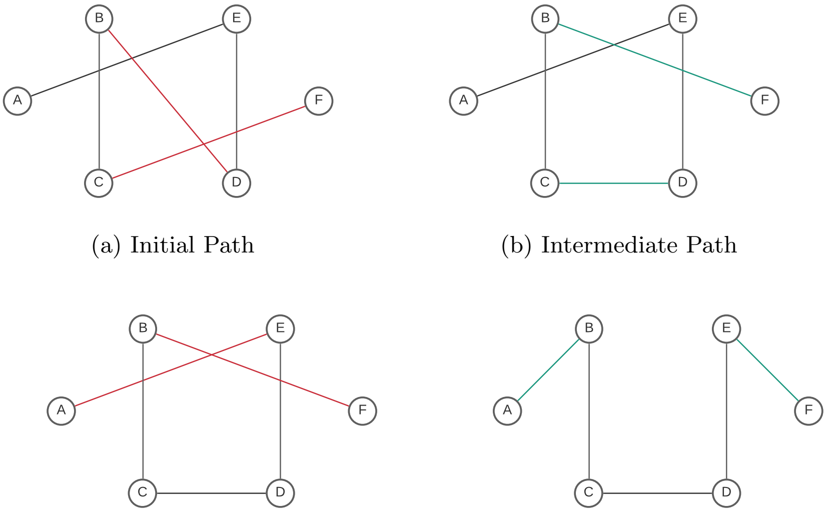
The Or-Opt operator relocates the segment $E \rightarrow D$ between nodes $C$ and $F$. Overall, the set of removed and inserted edges is the same as in the Three-Opt move presented before. The final path is optimal.

The Lin-Kernighan heuristic evaluates alternative paths obtained by a sequence of multiple Two-Opt moves. They are restricted to some pool of promising nodes whose size depends on the position of a move in the sequence. The heuristic is proven empirically to return optimal or near-optimal TSP tours [160] and is the most advanced local search operator employed by the multi-stage optimisation algorithm. Since its initial proposal [161], the Lin-Kernighan operator attracted a substantial interest of the research community. It resulted in notable refinements of the heuristic by [162] and [160]. A careful explanation of the Lin-Kernighan implementation details and its most popular variants are covered in the book [141][Chapter 15].

Figure 5.8 illustrates a sequence of two Two-Opt moves performed by the Lin-Kernighan heuristic, which led to an optimal path.

In the presented example, the Lin-Kernighan operator subsequently removes pairs of intersecting edges. Edges $B \rightarrow D$ and $C \rightarrow F$ are replaced by $B \rightarrow F$ and $D \rightarrow C$ in the first move. Then, edges $B \rightarrow F$ and $A \rightarrow E$ are substituted by edges connecting the closest neighbours $A \rightarrow B$ and $E \rightarrow F$.

Figure 5.9 presents a LS operator that moves a path segment located between the



(a) Initial Path

(b) Intermediate Path

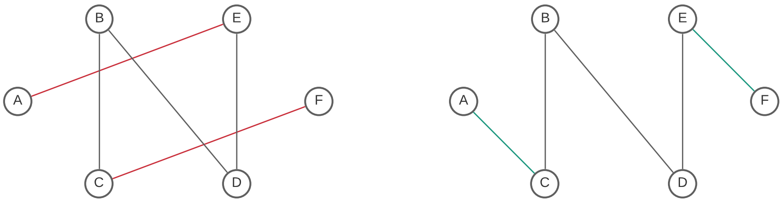
(c) Intermediate Path

(d) Resultant Path

Figure 5.8: Example Move: Lin-Kernighan Heuristic

longest edges.

The operator replaces two longest edges $A \to E$ and $C \to F$ by edges $A \to C$ and $E \to F$. It does not lead to an optimal path, but it is the shortest path obtained by removing two edges.



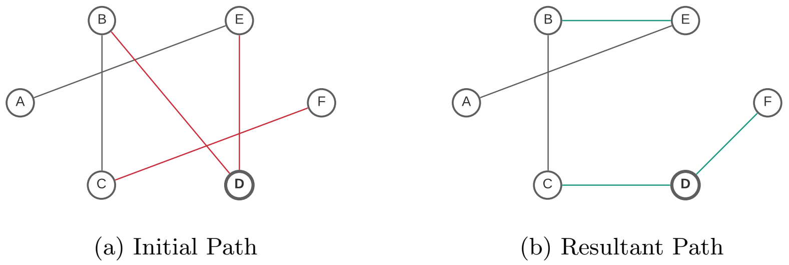
(a) Initial Path

(b) Resultant Path

Figure 5.9: Example Move: Relocate a Segment Between Two Longest Edges

Figure 5.10 presents a move performed by a LS operator that relocates a node into a different position in the path. Such a heuristic is a restricted variant of the Or-Opt operator.

The operator removes the node $D$ and inserts it between nodes $C$ and $F$. The resultant path is suboptimal.

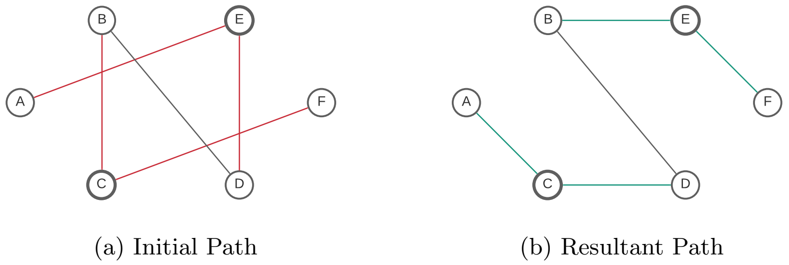


(a) Initial Path

(b) Resultant Path

Figure 5.10: Example Move: Relocate a Node

Figure 5.11 illustrates a LS operator that swaps two nodes located on the same path. Such a move may require removing up to four edges. Despite the number of edges affected, only one possible path is created by such an operation.



(a) Initial Path

(b) Resultant Path

Figure 5.11: Example Move: Swap Nodes in a Path

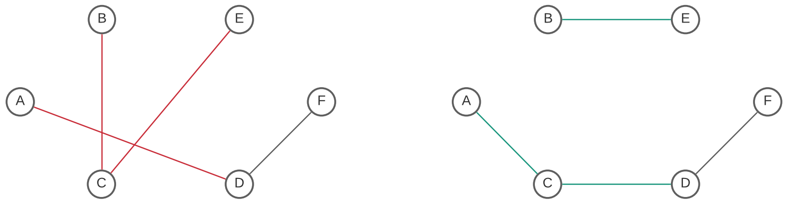
Position of nodes $E$ and $C$ on the path are swapped. Additional moves, i.e., the swap of nodes $B$ and $D$ followed by the swap of nodes $D$ and $E$, would have to be performed to obtain an optimal path.

### Local Search Operators for Multiple Paths

The LS operators discussed so far performed operations on nodes that belong to the same path. Heuristics presented here relax that restriction.

Intuitively one of the most straightforward LS operators for processing multiple paths is swapping nodes between paths. Figure 5.12 presents a LS operator that relocates a node from one path to the other.

The node $C$ is transferred from the path $B \rightarrow C \rightarrow E$ into the path $A \rightarrow D \rightarrow F$.



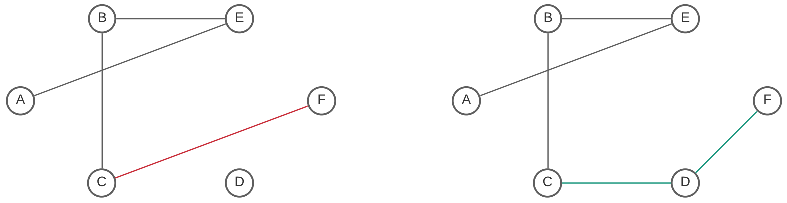
(a) Initial Paths

(b) Resultant Paths

Figure 5.12: Example Move: Relocate Nodes Between Paths

The new node is inserted between nodes $A$ and $D$. The final collection of paths has the lowest overall cost among combinations of two paths that cover the set of nodes.

The multi-stage solution algorithm must handle nodes not covered by any path. A dedicated LS operator is employed for that purpose to insert a disconnected node into a path. Figure 5.13 presents an example move of inserting a node that was not visited into a path.



(a) Initial Path

(b) Resultant Path

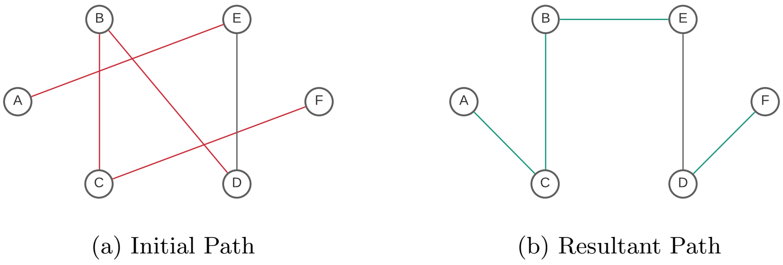
Figure 5.13: Example Move: Insert a Disconnected Node

The node $D$, which was not covered by any path, is inserted between nodes $C$ and $F$.

### Local Search Operators for Large Neighbourhood Search

The final group of LS operators are embedded in the LNS in which the CP solver from the or-tools framework [17] is responsible for the repair operation and constructing efficient paths. Consequently, the design of the LS operators is simplified as it reduces to finding structures in the incumbent solution, i.e., entire paths, path segments or nodes not visited to destroy, so the solver could rebuild them.

Figure 5.14 illustrates a LS operator that destroys two path segments of up to three nodes long. There are no restrictions regarding the location of the segments. In an extreme case, they could belong to the same path or even overlap.



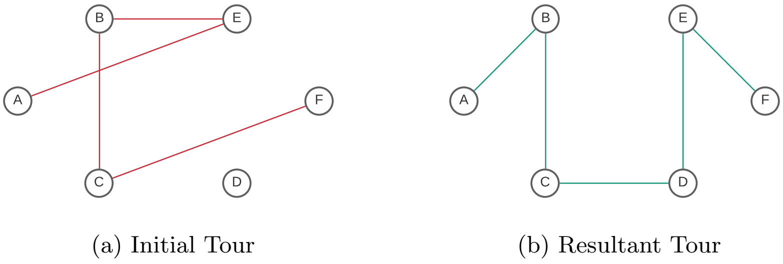
(a) Initial Path

(b) Resultant Path

Figure 5.14: Example Move: Destroy Two Path Segments

Two segments $A \rightarrow E$ and $D \rightarrow B \rightarrow C \rightarrow F$ are removed from the path. The edge $C \rightarrow B$ is retained in the resultant path with the opposite orientation.

Nodes that are not covered by any path are processed by a LS operator that selects a path and destroys it entirely. Subsequently, the solver attempts to form a new path considering nodes that were not covered before. An example of such a move is presented in Figure 5.15.



(a) Initial Tour

(b) Resultant Tour

Figure 5.15: Example Move: Destroy a Path and Nodes Not Covered

The LS operator destroys the path $A \rightarrow E \rightarrow B \rightarrow C \rightarrow F$. Then, the CP solver performs a repair operation to create a new path which now includes the node $D$.

The LS operators explained above are employed in various stages of the multi-stage optimisation algorithm. The following sections explain the details of each stage.

### 5.5.2 First Stage

The first stage computes a schedule that covers only visits with pairwise synchronisation constraints. Other visits are declined. The solution is used to create an initial guess for the second stage optimisation, which has the flexibility to change the assignment between vehicles and visits. Due to the provisional nature of the first stage solution, a heuristic approach that groups vehicles into pairs is applied. The vehicles matched together will operate as a single unit following the same route, commencing visits, and taking breaks.

Vehicles are matched using a greedy strategy that maximises the working hours and skills shared by the paired vehicles. The pseudocode of the heuristic is illustrated in Algorithm 5.

```
Parameters: $C$
$T \leftarrow \emptyset$; $R \leftarrow \emptyset$
foreach $c_i \in$ SORT-DESC$(C, \omega)$ do
    if $c_i \in R$ then
        continue
    end
    $R \leftarrow R \cup \{c_i\}$; $A \leftarrow C \setminus R$
    if $A = \emptyset$ then
        return $T$
    end
    $m \leftarrow c_\emptyset$
    foreach $c_k \in A$ do
        if $\omega(c_i, c_k) \geq \underline{t}$ and $\omega(c_i, c_k) \geq \omega(c_i, m)$ and $\sigma(c_i, c_k) \geq \sigma(c_i, m)$ then
            $m \leftarrow c_k$
        end
    end
    if $m \neq c_\emptyset$ then
        $R \leftarrow R \cup \{m\}$; $T \leftarrow T \cup \{(c_i, m)\}$
    end
end
return $T$
```

**Algorithm 5:** Team Selection

The algorithm takes the set of vehicles $C$ as the input. The matched vehicles are stored in the set $T$. The auxiliary variable $A$ is the set of vehicles currently available for creating a pair. Conversely, vehicles stored in the set $R$ are not eligible because they either have been paired already or no other vehicle is compatible with them. The vehicles can be matched if they can work at least $\underline{t}$ time (e.g., 2 hours and 15 minutes).

The vehicles are processed in descending order according to the total working hours. If a given vehicle remains available, the algorithm attempts to pair it with another available vehicle to maximise shared working hours and skills. If no valid companion vehicle is available, the first vehicle will not be considered for creating any other pair. Assuming that the sorting algorithm of complexity $O(nlog(n))$ is applied, the computational complexity of the pair selection is $O(n^2log(n))$ where $n$ is the number of the vehicles.

The first stage solves the deterministic CP formulation for the problem instance $F(V^{(T)}, C^{(T)})$ restricted to visits with synchronisation constraints ($V^{(T)}$) using the vehicle pairs ($C^{(T)}$).

For every pair of visits with a pairwise synchronisation constraint, one element of the pair is added to the set $V^{(T)}$.

Synchronisation and symmetry-breaking constraints can be removed from the formulation because they are satisfied due to the assumption that vehicles work in pairs.

Algorithm 6 illustrates the pseudocode of the first stage optimisation.

A parallel cheapest insertion heuristic computes the initial guess for the first stage optimisation. Its implementation in the or-tools framework [17] is inspired by the construction heuristic developed by [163]. The algorithm considers every possible opportunity to insert an unstaffed visit into the current set of routes. The possible insertion operations are stored in the min priority queue $Q$ ordered by the marginal increase in the total travel time after the insertion operation is performed. If both vehicles in a team have depots in different locations, mean travel time is used instead. In every iteration, the algorithm performs the first feasible insertion with the lowest cost. The iteration is completed by adding to the priority queue new opportunities for inserting the remaining unstaffed visits, i.e., at two arcs connected to the most recently inserted visit. The algorithm terminates after all feasible insertions have been performed.

The solution created by the parallel cheapest insertion heuristic is passed as the initial guess to Algorithm 4 and optimised. The first stage employs a collection of LS operators that operate on a single path: Two-Opt, Or-Opt, the Lin-Kernighan

```
Parameters: Problem formulation $F(V^{(T)}, C^{(T)})$, Search operators $S^1$
Result: First stage solution $\mathbf{y}^{(1)}$
$\mathbf{y} \leftarrow [1, \ldots, |N^{(T)}|]$
for $c \in C^{(T)}$ do
    $y_{b_c} \leftarrow e_c$
end
$Q \leftarrow \emptyset$  $\triangleright$initialise a priority queue
for $v \in V^{(T)}$ do
    for $c \in C^{(T)}$ do
        ADD$(Q, \delta(b_t, v) + \delta(v, e_t) - \delta(b_t, e_t), [b_t, e_t, v])$ $\triangleright$insertion $b_t - [v] - e_t$
    end
end
while $Q \neq \emptyset$ do
    $a, b, v \leftarrow$ POP$(Q)$  $\triangleright$get the next cheapest insertion
    if $y_a \neq b$ then
        $\triangleright$nodes a and b are no longer directly connected
        continue  $\triangleright$insertion not possible
    end
    $\mathbf{y}^{(c)} \leftarrow \mathbf{y}$  $\triangleright$perform the insertion on a copy
    $y_a^{(c)}, y_v^{(c)} \leftarrow v, b$ $\triangleright$insert $v$ between nodes $a$ and $b$
    if IS-FEASIBLE$(\mathbf{y}^{(c)})$ then
        $\mathbf{y} \leftarrow \mathbf{y}^{(c)}$  $\triangleright$accept the insertion
        for $v' \in V^{(T)}$ do
            if $y_{v'} = v'$ then
                $\triangleright$visit $v'$ is still available for insertion
                ADD$(Q, \delta(a, v') + \delta(v', v) - \delta(a, v), [a, v, v'])$ $\triangleright$insertion $a - [v'] - v$
                ADD$(Q, \delta(v, v') + \delta(v', b) - \delta(v, b), [v, b, v'])$ $\triangleright$insertion $v - [v'] - b$
            end
        end
    end
end
return OPTIMISE$(F(V^{(T)}, C^{(T)}), \mathbf{y}, S^1)$
```

**Algorithm 6:** First stage optimisation.

heuristic [161], swap two nodes, relocate a node, move a path segment between two longest edges. The suite is complemented by the LS operators that attempt to insert a disconnected node into a path and swap segments of nodes at the beginning of two paths. The first stage is terminated after all possible opportunities to improve the incumbent solution using LS have been attempted.

Let discuss the benefits and the limitations of the first stage optimisation. Computational results presented in Chapter 6 demonstrate that a partial solution for visits with synchronisation constraints helps the solver to staff considerably more visits, i.e., the number of the declined visits is reduced approximately from 30% to less than 1% in the problem instances considered. This noteworthy improvement is easy to accomplish, accounting for the difficulty of the first stage problem. The proportion of visits with synchronisation constraints varies typically between 10 and 20% of all visits for problem instances discussed in the literature. Therefore, vehicles grouped into pairs need to staff only a small fraction of visits from the original problem instance. Consequently, the optimisation problem considered at the first stage is easy to solve. Computational results in Chapter 6 corroborate this intuitive reasoning.

The limitations of the heuristic approach may arise while dealing with small problem instances with a handful number of vehicles whose working hours and skills are not compatible. Understandably, vehicles with significantly different working hours can work together for a small fraction of their availability. However, the issue does not affect large problem instances from the real world with dozens of vehicles available as the fleet of vehicles is large enough to match them into pairs with similar working hours and skills.

### 5.5.3 Second Stage

The second stage computes a schedule for the original problem instance $F(V, C)$. It contains all visits, including those without pairwise synchronisation constraints. Algorithm 7 presents the pseudocode of the second stage optimisation.

Before commencing the optimisation, the first stage solution is transformed into an initial guess for the second stage. Teams of two vehicles ($T$) are split into individual vehicles. In a similar vein, every visit with a pairwise synchronisation constraint is replaced by two routing nodes. Each node should be visited by a different vehicle from the same team. Visits for which synchronisation constraints are not defined are left unassigned in the initial guess. The collection of the LS operators used in the first stage ($S^1$) is extended by including two additional LNS operators ($S^2$). The first operator destroys one complete path and the set of uncovered nodes. The second

```
Data: First stage solution $\mathbf{y}^{(1)}$
Parameters: Problem formulation $F(V, C)$, Search operators $S^1 \cup S^2$
Result: Second stage solution $\mathbf{y}^{(2)}$
$\mathbf{y}^{(i)} \leftarrow [1, \ldots, N]$
for $n \in N^{(T)}$ do
    $n_1, n_2 \leftarrow$ FIRST($n$), SECOND($n$)
    $y_{n_1}, y_{n_2} \leftarrow$ FIRST($y_n^{(1)}$), SECOND($y_n^{(1)}$)
end
return OPTIMISE($F(V, C), \mathbf{y}^{(i)}, S^1 \cup S^2$)
```

**Algorithm 7:** Second stage optimisation.

operator destroys two path segments up to three nodes long. The solver uses the deterministic CP formulation, which minimises the sum of the total travel time for all vehicles and the penalty for declined visits. The optimisation is stopped after the LS operators cannot improve the incumbent solution further. Out of the schedules with the fewest declined visits, the solver returns the solution that has the lowest total travel time.

### 5.5.4 Third Stage

The third stage aims to decrease the cost of the schedule or alter the problem formulation and optimise a different objective.

Regardless of the objective function, the improvements that lead to more declined visits are rejected. If the initial guess derived from the final solution of the second stage has no declined visits, then performing all visits becomes mandatory for the third stage. Otherwise, the following constraint is added to prevent the number of declined visits from increasing.

$$\text{at\_most}(\mathbf{w}, c_{\emptyset}, m) \tag{5.55}$$

Recall that $\mathbf{w}$ is the vector of integer variables indicating the visiting vehicle and $c_{\emptyset}$ is the auxiliary vehicle dispatched to declined visits. The constant $m$ is set to the number of declined visits in the final schedule computed by the second stage.

Further reduction of the cost is possible by adopting the GLS meta-heuristic. Its implementation in the or-tools framework [17] includes the term below in the definition of the objective function:

$$\lambda \sum_{n \in N^b} (p_{n,y_n} \delta(n, y_n)) \tag{5.56}$$

The scaling factor $\lambda$ is set to 0.1. The current value of the penalty for using the arc $(a, b)$ is stored in the variable $p_{a,b}$. The formulation is denoted with the GLS variant of the objective function as $F^{(\mathrm{GLS})}(V, C, \mathbf{p})$. The vector $\mathbf{p}$ stores the values of penalties.

Algorithm 8 illustrates the pseudocode of the third stage optimisation.

The third stage is started from the best solution found by the second stage. The method proceeds iteratively until the maximum time without improvement to the incumbent solution is exceeded. Initially, the penalty for using each arc is zero. In every iteration, the solution most recently added to the pool is used as the initial guess to commence optimisation using Algorithm 4. The collection of search operators employed in the first and the second stage ($S^1 \cup S^2$) is further extended by adding a LS operator that moves a sequence of visits either forward or backwards only if the relocation reduces the travel time.

For every arc $(a, b)$ traversed in the solution found in a given iteration, the GLS meta-heuristic computes the expression $\delta(a, b)/(p_{a,b} + 1)$ known as the utility of a feature. Subsequently, arcs are sorted in descending order by their utility and the penalties for arcs that attained the highest value are increased by one. By analysing the formula for the utility of a feature, it is apparent that the penalty for using a costly arc is not set to an excessive value if the arc frequently appears in the solutions.

Overall, three alternative variants of the third stage, which reflect different preferences of the decision-maker regarding the formulation of the objective function, were developed.

```
Data: Second stage solution $\mathbf{y}^{(2)}$
Parameters: Problem formulation $F(V, C)$, Search operators $S^1 \cup S^2 \cup S^3$,
            Solution pool $P$
Result: Third stage solution $\mathbf{y}^{(3)}$
ADD($P, \mathbf{y}^{(3)}$)
$\mathbf{p} \leftarrow [0]^{|N| \times |N|}$  $\triangleright$initialise GLS penalties
$t \leftarrow 0$
while ELAPSED-TIME() $- t \leq \Delta$ do
    $\mathbf{y}^{(i)} \leftarrow$ NEXT($P$)
    $\mathbf{y}^{(c)} \leftarrow$ OPTIMISE($F^{(\mathrm{GLS})}(V, C, \mathbf{p}), \mathbf{y}^{(i)}, S^1 \cup S^2 \cup S^3$)
    if OBJ($\mathbf{y}^{(c)}$) < OBJ(BEST($P$)) then
        $t \leftarrow$ ELAPSED-TIME()
    end
    ADD($P, \mathbf{y}^{(c)}$)
    $\mathbf{d} \leftarrow []$
    for $n \in N$ do
        $\triangleright$compute the utility for each arc in the solution
        if $y_n^{(c)} \neq n$ then
            $\mathbf{d} \leftarrow \mathbf{d} || [\delta(n, y_n)/(p_{n,y_n} + 1), n, y_n]$  $\triangleright$append a row to the 2D vector
        end
    end
    $\triangleright$increment penalties for arcs with the highest utility
    SORT-DESC($\mathbf{d}, \iota_1$)
    $u_{\max} \leftarrow d_{0,0}$
    for $(u, a, b) \in \mathbf{d}$ do
        if $u \neq u_{\max}$ then
            break
        end
        $p_{a,b} \leftarrow p_{a,b} + 1$
    end
end
return BEST($P$)
```

**Algorithm 8:** Third stage optimisation.

### Distance Minimisation

This variant of the third stage makes no changes in the formulation of the objective function. It attempts to decrease further the cost of the schedule defined as the sum of the total travel time for all vehicles and the penalty for declined visits. The opportunity for improvement is due to enabling additional LS operators and the GLS meta-heuristic.

Both were not active in the second stage.

### Vehicle Number Minimisation

It might be desirable to utilise fewer vehicles to execute a schedule, especially if locum vehicles would have to be used otherwise. One could relocate redundant vehicles to other operational areas (a.k.a. districts) where they could contribute to staffing more visits. Understandably, using a smaller fleet of vehicles may increase the total travel time needed to service the requested visits. Suppose the marginal increase in the total travel time below the threshold $u_v$ motivated by the prospect of releasing the vehicle $v$ is accepted. Then, the objective function is changed to the following definition.

$$\min \sum_{c \in C : y_{b_c} \neq e_c} u_c + \sum_{n \in N^b} \delta(n, y_n) + \rho \sum_{v \in V^1} (1 - a_v) + \frac{1}{2} \rho \sum_{v \in V^2} (1 - a_v) \tag{5.57}$$

The cost of using a vehicle is fixed and paid once, regardless of the number of visits it staffs. The clarity of exposition is the reason for this simple treatment. Understandably, it might be worthwhile in some applications to consider a variable cost according to the number of visits the vehicle performs or to distinguish between full-time and part-time vehicles. Since a general-purpose and open source CP solver is used, it suffices to introduce new variables and constraints to implement more advanced customisations. This topic is explored further in the last variant of the third stage.

### Delay Minimisation

Here the CP formulation introduced in Section 5.4, which employs the ERI decision criterion, is used. Consequently, the focus is shifted from the optimisation of the cost towards improving the quality of the schedule. The aim is to reduce the delay in commencing visits caused by disruptions, i.e., previous visits performed by the vehicle lasting longer than expected. If the final schedule found by the second stage is an infeasible initial guess according to the new formulation, then the solution will be pre-processed, and visits that have a positive expected delay will be removed. Finally, the derived optimisation problem is solved to staff all visits while minimising the riskiness index.

## Concluding Remarks

Due to the scalability limitations of MIP apparent when dealing with large VRP instances, the solution methodology is built on top of CP. The main contributions of this chapter are the multistage optimisation algorithm for solving large instances of VRPs' and the algorithm for computing the ERI. Computational results discussed in the following chapter demonstrate the usefulness of the solution methodology presented here. The concepts introduced in this chapter were implemented in the open-source CP solver [17]. The extensions proposed in the chapter are publicly available in the source code repository [24] for testing and benchmarking purposes. To the best of the author's knowledge, it is the first time when ERI optimisation is used in CP.

# Chapter 6

# Application: Vehicle Routing Problem with Side Constraints

## 6.1 Introduction

This chapter demonstrates the usefulness of the methodology proposed to solve the VRPTWSync. It features two groups of instances: the VRPTWSync benchmark [144] and a collection of large VRPSC instances. Computational results for the benchmark will demonstrate whether the multistage optimisation algorithm is competitive with other solution methods proposed in the literature. The second group comprises sample problem instances solved daily by the largest home care provider in Scotland. Regarding the number of visits, they are the largest instances with pairwise synchronisation constraints considered in the literature. Solving them should verify the scalability of the multistage optimisation algorithm. Besides validating the methodology, the experiment has a societal impact. It helps the home care organisation determine the scale of the possible savings due to routing and scheduling optimisation.

### Structure of the Chapter

Section 6.2 covers the literature on solution methods applied to the VRPTWSync and the VRPSC. Section 6.4 presents the computational results using the VRPTWSync benchmark [144]. Section 6.5 thoroughly describes the VRPSC application. In particular, it illustrates the performance characteristics of the multistage algorithm and demonstrates the benefits of every variant of the third stage configuration. Some concluding remarks close the chapter.

## 6.2 Relevant Literature

The specialisations of the VRP, i.e., VRPTW, introduced in Chapter 5 are pure and abstract problems. The simplifying assumptions, i.e., a homogeneous fleet of vehicles that work without taking breaks, allow focusing exclusively on the specific combinatorial structure of the problem (i.e. vehicle routing). However, solution methods developed for such problems may require further work to meet the requirements of real-world applications. To see this, consider scheduling personnel in HHC, which is the target application featured in this chapter. Understandably, the ultimate objective is to find a cost-effective schedule that can be executed. Such a schedule must be compliant with labour law. These observations motivated the definition of the VRPSC that extends the pure VRPTWSync by modelling working hours and contractual breaks, skill-based routing, and the continuity of visits. Besides domain-specific requirements, another crucial aspect is uncertainty which naturally affects service times, travel times, customers' presence and demand. The features related to modelling uncertainty close the literature review.

### 6.2.1 Vehicle Routing Problem in Home Health Care

The section focuses on daily (a.k.a. single period) scheduling of personnel in HHC applications for which the constraints VRPSC implements are of critical importance. Particular attention is devoted to modelling contractual breaks and the solution methods applied to instances containing visits with pairwise synchronisation constraints. Aspects related to the problem decomposition [164], districting [165] and scheduling over multiple days [153] are outside of the scope of this work. For a comprehensive discussion of OR applications in HHC, the interested reader is referred to recent literature reviews [166, 167]. The researchers thoroughly described the application domain, justified the constraints considered in this chapter, mentioned well-known optimisation problems, and the solution methods.

Despite the focus on HHC, any personnel scheduling involves solving routing problems whenever the aspect of travelling is present [168], i.e., scheduling technicians to perform maintenance tasks, scheduling patrols of security guards, refuse and recycling collection etc. Collectively, such applications can be classified as the Workforce Scheduling and Routing Problems (WSRP) [168] or the Vehicle Routing Problem with Resource Constraints (VRPRC) [169]. Problems in this group share a common set of domain objects and modelling assumptions, such as the treatment of time windows, the availability of different modes of transport, a homogeneous or heterogeneous skill set of employees, among others. Skill requirements could be complemented or replaced by preferences measuring satisfaction from assigning a vehicle to a given visit. Contrary to skills, if preferences are not satisfied, they do not render the schedule infeasible. A pure VRP with modelling skills is known as the Skill Vehicle Routing Problem (SkillVRP) [170].

The theory of vehicle routing has been developed separately from scheduling. The signature scheduling problem is known as Job-Shop. Its subject is scheduling a sequence of jobs on a computing cluster of machines with different hardware specifications to minimise the completion time of all jobs (a.k.a. makespan). Scheduling problems that combine the routing aspect are formulated as VRPs [171], and the scheduling component is handled using side constraints. The established way to model a VRP is to either use a network flow-based model [19,149,168,172,173] or a set-partitioning formulation [150,174]. The former models are arguably simpler to explain. However, they can solve only small problem instances using MIP solvers (i.e., 25-50 visits) due to a weak bound of the LP relaxation. The optimisation of the set-partitioning formulation requires developing the branch-and-cut-and-price framework (a.k.a. column generation). Despite a stronger formulation, problem instances that contain more than 150 visits are practically intractable for exact methods [169]. If synchronisation constraints are present (i.e., some pairs of visits should start at the same time), effectively solvable problem instances are even smaller, e.g., 25-50 visits [150]. As a result, motivated by the need to solve larger problem instances, researchers drop the desirable but practically unattainable goal to find a provably optimal solution and apply hybrid methods instead. Hybrid methods integrate a vast array of optimisation paradigms and search techniques: CP [153,155,175], LS, MIP [19,176], heuristics [82], and metaheuristics (i.e., Adaptive Large Neighbourhood Search (ALNS) [153], Genetic Algorithm (GA) [177], Guided Local Search (GLS) [178], Particle Swarm Optimisation (PSO) [179], Simulated Annealing (SA) [151], Tabu Search (TS) [155,173], and Variable Neighbourhood Search (VNS) [176, 180]). Besides the ability to find cost-effective solutions to large problem instances, hybrid methods offer shorter computational times. The latter aspect becomes relevant whenever several schedule variants should be computed [149, 155, 181] or when the demand for visits changes due to emergency requests and cancellations.

Solutions to a personnel scheduling problem can be compared based on their cost-effectiveness and quality [181]. The cost of a solution is driven by the number of employees required, the utilisation of their time (e.g., travel time, waiting time, overtime), and the number of visits performed. On the other hand, the quality measures how well-balanced the workload is and whether individual preferences of the personnel and clients are satisfied. Understandably, a carer may wish to work in the morning, or the policy to provide the continuity of visits (a.k.a. “the continuity of care”) could be enforced. Since cost and quality are conflicting goals, home care scheduling optimisation often uses a weighted sum of multiple objectives, such as the total travel time, waiting time, overtime, incentives for satisfying individual preferences, and a penalty for uncovered visits, as discussed by [154, 167]. A comprehensive reference list of the HHC domain features modelled as constraints and alternative objective functions can be found in the survey [167]. Regardless of the objective defined, all feasible solutions must obey constraints enforced by labour law and employment contracts.

Modelling approaches proposed in the literature for handling breaks often require simplifying assumptions. For example, breaks could be considered as visits for which a carer should return to the depot [182]. However, this is only justifiable for long breaks in urban areas, where travel times to reach the depot are short. Another approach was to reserve specific time slots in carers’ schedules as not available for work [183]. Nevertheless, it is inflexible as the waiting time preceding a visit could be accommodated as a break. In countries where labour law requires a break after a certain number of hours, shifts can be shorter than the period necessary to schedule a break [153]. Eventually, a more general approach suitable for one break (a.k.a. lunch break) of arbitrary length was presented in [149, 180]. The researchers defined routing in a graph, where visits and breaks were two separate classes of nodes. Break nodes could only be reached from visit nodes, and a carer must return to the preceding visit node after the break to continue the route. Taken together, the modelling techniques for breaks discussed so far impose at least one of the restrictions: the return to a depot is required, no time windows flexibility compared to visits is offered, and the number of breaks is restricted. Furthermore, the modelling techniques, though suitable for longer breaks such as lunch, are not well-posed for short breaks (a.k.a. relief breaks). For that reason, this chapter applies a generic model with an arbitrary number of breaks per carer which may also have time windows.

The synchronisation between carers, i.e., to support clients with reduced mobility [174], poses additional difficulty on the solution method. A pragmatic approach is to enforce synchronisation by fixing the visits' start times [19, 173]. Nonetheless, it may lead to suboptimality or infeasibility. Another idea is to form teams of carers as in the MAP [146, 184, 185]. For this application, [145] proposed an ALNS meta-heuristic with teaming-aware destroy and repair operators supporting the construction of arbitrary size teams. As a downside, the technicians grouped in a team were supposed to work together for the whole day, which could be overly restrictive in the HHC domain. Ultimately, if synchronisation is limited to pairwise synchronisation, the dominating theme in the literature is to avoid introducing simplifying assumptions, such as team formation.

Table 6.1 presents the literature on visits with pairwise synchronisation constraints. Most of the referenced articles were summarised in the surveys [166, 167, 186]. For each article, the columns of the table indicate the variant of the problem considered, the solution approach and the support for important features: contractual breaks (B), the continuity of care (CC), and routing based on skills or preferences (SP). The table does not include papers for which it is not clear how many visits required synchronisation (i.e., [181]).

Table 6.1: Solution methods applied to the VRPTWSync and the HCSRP with pairwise synchronisation constraints. The columns of the table report: the referenced article, the problem considered, the solution method, the features supported in the problem formulation, and the size of the largest instance solved. The following features are distinguished: shift patterns and contractual breaks (B), the continuity of visits (VC), and the skill-based routing (SR). The implementation of breaks could be limited to a lunch break (L) or supporting an arbitrary number of breaks (Y). The skill-based routing can include only skills (S), only preferences (P), or both (SP). The number of vehicles (C), the number of visits that require one vehicle ($V^1$), and the number of visits with synchronisation constraints ($V^2$) are reported for the largest problem instance solved. A pair of visits from the set $V^2$ corresponds to one regular visit from the problem domain. The table entries are sorted in ascending order according to the number of visits with pairwise synchronisation.

| Source | Problem | Solution Method | Features | | | Largest Instance | | |
|---|---|---|---|---|---|---|---|---|
| | | | B | VC | SP | C | $V^1$ | $V^2$ |
| [183] | HCSRP | MIP | - | - | S | 40 | 32 | 8 |
| [144] | VRPTWSync | Branch and Price | - | - | P | 16 | 64 | 16 |
| [151] | VRPTWSync | Heuristic + LNS | - | - | P | 16 | 64 | 16 |
| [187] | VRPTWSync | LNS | - | - | - | 16 | 64 | 16 |
| [188] | HCSRP | LNS | - | - | - | 16 | 64 | 16 |
| [189] | HCSRP | VNS | - | - | - | 16 | 64 | 16 |
| [177] | HCSRP | GA | - | - | - | 16 | 64 | 16 |
| [19] | VRPTWSync | Heuristic + MIP | - | - | P | 9 | 0 | 18 |
| [179] | HCSRP | GA, TS, SA, PSO | - | - | S | 5 | 0 | 20 |
| [190] | HCSRP | Commercial Product | L | Y | S | 5 | 80 | 20 |
| [191] | VRPTWSync | LNS | - | - | P | 50 | 160 | 40 |
| [174] | HCSRP | Branch and Price | - | - | SP | 15 | 104 | 46 |
| [192] | HCSRP | Heuristic + LNS | - | - | S | 40 | 100 | 200 |
| This Work | HCSRP | Heuristic + LNS + GLS | Y | Y | S | 63 | 388 | 240 |

All tabulated articles consider visits with time windows in the hard sense. Understandably, researchers who studied the pure VRPTWSync did not model contractual breaks and the continuity of visits. The absence of these features in models for HHC scheduling with synchronised visits is unusual because such constraints are well-justified in the problem domain and were implemented in formulations for HHC scheduling without pairwise synchronisation, i.e. [149]. Overall, only one author [190] provided support for breaks, albeit limited to a single lunch break per carer, and the continuity of visits. Conversely, skill-based routing is widely implemented.

Most importantly, the problem instances with pairwise synchronisation considered in the literature are disproportionally smaller compared to instances solved in articles that focus entirely on visits without synchronisation [167], i.e., 500-700 visits [149, 152, 154]. Predictably, the need to deal with problem instances of comparable size, albeit with synchronisation constraints included, arises in practical applications and is the primary motivation for this work. An additional benefit of solving large problem instances is the opportunity to surface computational bottlenecks, which may be less pronounced when the number of visits is smaller.

### 6.2.2 Vehicle Routing Problem under Uncertainty

Many parameters of VRPs' are affected by uncertainty in the real world. Common examples addressed in the literature include travel times, visits' duration (a.k.a. service times), customers' presence and demand. Conceptually, the treatment of uncertainty for travel times and service times is the same. Both parameters are pivotal for the VRPTW. On the other hand, demand uncertainty and customers' presence are typical for the CVRP.

Travel time and service time uncertainty are modelled using either SO or RO frameworks. The choice of the solution approach depends on the interpretation of time windows and the possibility to modify the incumbent solution in response to the realisation of uncertainty. In SO, time windows are soft, and their violation is subject to a penalty. Furthermore, the solution can be changed through recourse actions, which are taken after the uncertainty is revealed, i.e., re-optimisation, a detour to depot [193], etc. Conversely, time windows in RO must be respected for all possible realisations of uncertainty and altering the solution found during optimisation is disallowed.

Formulations of the uncertain VRPTW in SO employ CC's which restrict the probability of starting a visit late for each customer. An alternative approach is to use a two-stage stochastic optimisation model with recourse. The cost of the recourse action is included in the objective. It represents inconvenience or dissatisfaction due to inferior quality of service, i.e., the total amount of time windows' violation.

In the tutorial on the robust VRP [194], the authors presented various modelling techniques for defining capacity and time windows constraints affected by uncertainty in demand, travel and service time. In the deterministic variant of the problem, capacity and time window restrictions were enforced using Miller-Tucker-Zemlin constraints, a tighter variant of the big-M constraints. The uncertainty in travel time was modelled using a budgeted uncertainty set. Demand uncertainty was addressed using several alternative approaches: CC's, uncertainty sets (i.e., an ellipsoidal uncertainty set, a box uncertainty set), recourse actions (i.e., re-optimisation, skipping some customers), and deliberately keeping some vehicles' capacity unused. Computational experiments on small problem instances (i.e., 12 visits, four vehicles and five scenarios) indicate that the effectiveness of the treatment of uncertainty depends on its magnitude. Chance Constraints are useful only for small uncertainty (i.e., 5%) [194]. Robust Optimisation offers more cost-effective solutions for moderate uncertainty (i.e., 10-20%) [194]. Then, for a higher magnitude of uncertainty, the researchers recommend SO with recourse [194]. The ad hoc workaround with reserving some free capacity upfront could outperform solutions found using RO [194]. Nonetheless, it requires careful tuning of the buffer size through simulations and is sensitive to the characteristics of the problem.

For practical applications in the multi-period VRP, the authors postulate a workflow composed of the planning stage and the operational stage [194]. The objective of the planning stage is to compute a master route which then will be transformed into final routes after the uncertainty is observed. The merit of the master route is allowing human drives to get familiar with the area of operations. The master route can be computed using a RO model for customers who are visited most frequently. Daily routes are built to match the master route as closely as possible and meet the actual demand.

The Vehicle Routing Problem with Time Windows and stochastic travel times were investigated by [195]. The probability of late arrival to every visit was subject to CC's, and the objective was defined as the total travel time. The researchers devised an approximation scheme for computing probability distributions for the start time of a visit and the arrival time to the visit location. The approximation scheme made several assumptions on the structure of the probability distribution and the operations involving them. Travel times were assumed to be normally distributed and independent. Moreover, the researchers assumed that a maximum of two normally distributed variables is normally distributed. The same applies to the sum of two normally distributed variables. The set of assumptions was sufficient to obtain closed-form expressions for the propagation of mean and variance for the service time and the arrival time along the route a vehicle follows. The accuracy of the approximation scheme was tested during simulations. Although the precise results were depended on the tightness of time windows and the length of routes, the overall accuracy in the estimation of the mean and 95% percentile was satisfactory (i.e., the mean absolute percentage error below 1% and 4%, respectively) [195]. Arguably, the equations are simple enough to incorporate them in solution methods, which was demonstrated on the TS metaheuristic.

Agra et al. [196] studied the VRPTW and stochastic travel times modelled using the budgeted uncertainty set. The researchers developed two alternative formulations based on the classical network flow-based model with three-index binary variables encoding routes. The formulations were solved using custom solution methods exploiting the special structure of the problem and lazy generation of either cuts or extreme points.

One of the formulations was a two-stage model developed using the ARO framework. Routes were computed in the first stage, then travel time uncertainty was revealed, and visits' start times were calculated in the second stage. Instead of the conventional solution approach for the formulations of this kind, an approximation scheme utilising decision rules, the authors applied an iterative method that generated extreme points of the uncertainty polytope. Overall, such an approach executed naively would be viable only for small problem instances due to a large number of vertices. However, the researchers considered only those extreme points, which contributed to finding the optimal solution.

The other approach was a novel formulation in which robust constraints were modelled implicitly using an expandable pool of paths violating time windows constraints. The solution method alternated between solving the master problem and a subproblem.

The former included the deterministic part of the formulation and a list of cuts found by solving the sub-problem. New cuts were added on demand whenever the incumbent solution returned by the master problem contained routes that violated some robust constraints.

Both models were tested using a synthetic benchmark of maritime transport with problem instances having between 20 up to 50 cargoes to be delivered to different ports. Surprisingly, the problem formulation with a pool of infeasible paths solving a model with the treatment of uncertainty did not require more computational time compared to the equivalent formulation for the deterministic model [196]. Nonetheless, it was slower than the two-stage model. The additional effort invested in developing acceleration strategies allowed the researchers to solve considerably larger formulations than the classical reformulation.

Solutions of the robust shortest path problem with uncertain travel time obtained using different uncertainty sets were compared by [197]. The study is focused on testing the suitability of a specific uncertainty set parametrised using real data. In particular, the researchers studied the implications of applying a given uncertainty set and the performance of the robust solution. The authors considered a vast array of uncertainty sets: a convex hull, a box uncertainty set, a budgeted uncertainty set, an ellipsoidal uncertainty set, and a polyhedral uncertainty set constructed from CV@R. Overall, the uncertainty sets provided a different trade-off between the average performance, the worst-case performance and the computational time. Researchers recommended the polyhedral uncertainty sets constructed from CV@R and the ellipsoidal uncertainty sets, which provided a good trade-off between the average and the worst-case performance [197]. The symmetric polyhedral uncertainty set parametrised using the CV@R obtained the best performance on average [197]. On the other hand, the solutions computed by the formulation with the box uncertainty set were the most robust [197].

Depending on the problem instance, finding a feasible solution for every possible realisation of uncertainty could have a high cost. In particular, it may not be suitable for applications (i.e., timetabling) that allow a delayed execution of a solution after uncertainty is revealed. The extra time could be used to apply cost reduction modifications of the solution tailored for the given scenario. The researchers [198] targeted this application context and explored the idea of building a solution pool for every possible scenario and then selecting one representative solution. A popular choice could be a solution with minimum distance to all elements of the solution pool (a.k.a. median) or a solution with the lowest maximum distance to any solution from the pool (a.k.a. centre). Suppose the latter is the representative of the solution pool, and the distance measures the cost of the update. In that case, the incumbent can be transformed into the optimal solution for a given scenario at a minimal cost in the worst case. A strong justification in favour of the proposed approach arises if the deterministic variant of the problem has a specialised algorithm that can be used to initialise the solution pool. Secondly, the centre or the median typically has a better objective value compared to the robust solution that must be feasible for every scenario.

A different approach to optimisation over scenarios was proposed by [85]. Authors considered the CVRP with uncertain travel time. The objective was defined as the minimum total travel time for all routes in the worst-case scenario. Ties were broken by the lexicographical comparison of vectors which contained the total travel time for each scenario sorted in descending order. The authors developed several solution methods, including a MIP model, construction heuristics, a GRASP and several variants of metaheuristics. They were based on the ILS and optionally combined with multiple restarts and a TSP heuristic known as Giant Tours. The heuristic temporarily relaxes the capacity constraints to reduce the VRP to the TSP, which enables an application of highly efficient solution algorithms. Then, the TSP tour is split into routes for individual vehicles that respect the capacity constraints. The best solution method in the computational study was the ILS with multiple restarts and the Giant Tours heuristic. It solved the largest problem instances with 100 visits, 20 vehicles and 20 scenarios in less than two minutes. Therefore, despite the additional effort to evaluate the objective for each scenario, which increases computational complexity by a multiplicative factor of $nlog(n)$ (i.e., the complexity of sorting the cost vector with n scenarios), the computation time requirements of the metaheuristic were plausible. On the other hand, the exact method failed to find better solutions than a construction heuristic within 4 hours of computation time for problem instances with 50 visits.

To conclude, there are various computational approaches for finding robust solutions to the VRP's. The literature review highlighted the context of use and limitations of RO and SO. The lack of a uniform and widely acknowledged solution method is partially a consequence of the scalability limitations of the existing methods and the differences in assumptions exploited to improve results. Robust Optimisation is not suitable for applications involving a large number of visits as effectively solvable problem instances include approximately 50 visits. Consequently, the second part of the literature review explored alternative approaches based on optimisation over scenarios. It is applied in the thesis in conjunction with the riskiness index optimisation to solve approximately ten times larger problem instances.

## 6.3 Implementation Details and Hardware Configuration

The multistage optimisation algorithm was implemented in C++ using the or-tools framework [17] as a software library. The CP solver and its internals (i.e., the parallel cheapest insertion heuristic, the search operators, and the GLS meta-heuristic) were reused. The software developed for the study is available open-source [24].

Computations were run on a machine with the AMD Ryzen 7 2700X processor and 32 GB of RAM.

## 6.4 Vehicle Routing Problem with Synchronised Visits

The section compares the performance of the multistage optimisation algorithm with other methods proposed in the literature by solving the VRPTWSync benchmark [144]. The multistage algorithm proposed in Section 5.5 will solve the CP formulation outlined in Section 5.3 with the total travel time as the objective to minimise. It is the most popular objective function in the literature for this group of problems. The benchmark contains 50 problem instances obtained from 10 baseline instances that differ in the total number of visits (i.e., 20, 50 and 80 visits) by applying one of five configurations of time windows. Time windows can be small (S), medium (M) or large (L). There are also corner cases with time windows of zero span (–), i.e., fixed visits' start times, or the span as wide as the scheduling horizon (H), i.e., no restrictions on visits' start times. Pairwise synchronisation constraints affect 20% of visits. The size of the fleet of vehicles is restricted, and every vehicle is available for the entire duration of the scheduling horizon (i.e., 9 hours). Performing all visits is mandatory. There are no additional side constraints.

Table 6.2 contrasts the cost of the best solutions reported in the literature with the results obtained using the multistage optimisation algorithm. The solutions for which the optimality has been proven are denoted with an asterisk.

The multistage optimisation algorithm reproduced the best solution for all problem instances solved to provable optimality in the literature for 28 out of 30 cases. Regarding the instances for which proving optimality remains an open problem, the multistage algorithm reproduced the best solution from the literature or improved it in 16 cases. The normalised marginal difference between the best solution reported in the literature and the solution found by the multistage algorithm is computed to quantify the accomplished improvement. It is denoted using $\Delta$ and formally defined as $\Delta = 100 \cdot (O_{\text{Lit}} - O_{\text{MS}})/O_{\text{Lit}}$, where $O_{\text{MS}}$ and $O_{\text{Lit}}$ indicate objective function values of the schedule found by the multistage algorithm and the best schedule reported in the literature, respectively. Overall, the multistage algorithm reported new best solutions for five instances (9S, 6A, 7A, 8A, 9A). In four cases (10S, 10M, 9L, 10L), the solution proposed in the literature were strictly better, albeit the difference compared to the solution found by the multistage algorithm was relatively small (less than 0.7%). Finally, the multistage algorithm did not find a feasible solution for two problem instances (8F and 9F). They seem to be the hardest instances in the benchmark because, as of this writing, only [187] can solve them. The authors used the SA metaheuristic combined with the ALNS, which employed custom repair operations enhanced to handle visits with synchronisation constraints.

To conclude, accounting for the cost of the solutions reported, the multistage algorithm is comparable with the most successful computational methods described in the literature [151, 187, 191]. For each method in this group, there is at least one problem

Table 6.2: Objective values for the best solutions reported in the literature compared to the multistage optimisation algorithm. Column (I) contains the problem instance. The subsequent Columns ($C$), ($V^1$), and ($V^2$) store respectively the size of the fleet of vehicles, the number of visits that require one vehicle, and the number of visits with pairwise synchronisation constraints. Column ($O_{\text{Lit}}$) reports the best objective value known from the literature and indicates the first reference article which published the result. Column ($O_{\text{MS}}$) presents the cost of the best solution found by the multistage algorithm. Computational time since starting optimisation to finding that solution is specified in Column ($T_{\text{MS}}$). Finally, Column ($\Delta$) displays the relative improvement defined as the difference between the cost of the solution found by the multistage algorithm and the result from the literature divided by the cost reported in the literature.

| P | C | $V^1$ | $V^2$ | $O_{\text{Lit}}$ | $O_{\text{MS}}$ | $T_{\text{MS}}$ [s] | $\Delta$ | P | C | $V^1$ | $V^2$ | $O_{\text{Lit}}$ | $O_{\text{MS}}$ | $T_{\text{MS}}$ [s] | $\Delta$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1F | 4 | 16 | 2 | 5.13* [19] | 5.13 | 0.1 | 0 | 6M | 10 | 40 | 5 | 7.7 [151] | 7.7 | 41.89 | 0 |
| 2F | 4 | 16 | 2 | 4.98* [19] | 4.98 | 0.01 | 0 | 7M | 10 | 40 | 5 | 7.48 [151] | 7.48 | 23.49 | 0 |
| 3F | 4 | 16 | 2 | 5.19* [19] | 5.19 | 3.51 | 0 | 8M | 10 | 40 | 5 | 8.54* [19] | 8.54 | 119.03 | 0 |
| 4F | 4 | 16 | 2 | 7.21* [19] | 7.21 | 0.01 | 0 | 9M | 16 | 64 | 8 | 10.92 [151] | 10.92 | 129.36 | 0 |
| 5F | 4 | 16 | 2 | 5.37* [19] | 5.37 | 0.76 | 0 | 10M | 16 | 64 | 8 | 7.62 [151] | 7.67 | 1590.21 | -0.66 |
| 6F | 10 | 40 | 5 | 14.45* [177] | 14.45 | 12.74 | 0 | 1L | 4 | 16 | 2 | 3.39* [144] | 3.39 | 0.18 | 0 |
| 7F | 10 | 40 | 5 | 13.02* [177] | 13.02 | 7.74 | 0 | 2L | 4 | 16 | 2 | 3.42* [144] | 3.42 | 7.43 | 0 |
| 8F | 10 | 40 | 5 | 34.94* [187] | - | - | - | 3L | 4 | 16 | 2 | 3.29* [144] | 3.29 | 0.07 | 0 |
| 9F | 16 | 64 | 8 | 43.48* [187] | - | - | - | 4L | 4 | 16 | 2 | 5.13* [144] | 5.13 | 0.3 | 0 |
| 10F | 16 | 64 | 8 | 12.08* [187] | 12.08 | 57.49 | 0 | 5L | 4 | 16 | 2 | 3.34* [144] | 3.34 | 0.37 | 0 |
| 1S | 4 | 16 | 2 | 3.55* [19] | 3.55 | 0.05 | 0 | 6L | 10 | 40 | 5 | 7.14* [144] | 7.14 | 53.92 | 0 |
| 2S | 4 | 16 | 2 | 4.27* [19] | 4.27 | 0.04 | 0 | 7L | 10 | 40 | 5 | 6.88 [144] | 6.88 | 52.03 | 0 |
| 3S | 4 | 16 | 2 | 3.63* [19] | 3.63 | 0.12 | 0 | 8L | 10 | 40 | 5 | 8 [151] | 8 | 2656.16 | 0 |
| 4S | 4 | 16 | 2 | 6.14* [19] | 6.14 | 7.38 | 0 | 9L | 16 | 64 | 8 | 10.43 [191] | 10.5 | 190.14 | -0.67 |
| 5S | 4 | 16 | 2 | 3.93* [19] | 3.93 | 0.24 | 0 | 10L | 16 | 64 | 8 | 7.36 [191] | 7.38 | 256.76 | -0.27 |
| 6S | 10 | 40 | 5 | 8.14* [144] | 8.14 | 1.82 | 0 | 1A | 4 | 16 | 2 | 2.95 [187] | 2.95 | 0.02 | 0 |
| 7S | 10 | 40 | 5 | 8.39* [144] | 8.39 | 9.16 | 0 | 2A | 4 | 16 | 2 | 2.88 [187] | 2.88 | 0.02 | 0 |
| 8S | 10 | 40 | 5 | 9.54* [144] | 9.54 | 86.72 | 0 | 3A | 4 | 16 | 2 | 2.74 [187] | 2.74 | 0.11 | 0 |
| 9S | 16 | 64 | 8 | 11.93 [151] | 11.92 | 2521.88 | 0.08 | 4A | 4 | 16 | 2 | 4.29 [199] | 4.29 | 0.02 | 0 |
| 10S | 16 | 64 | 8 | 8.54 [191] | 8.58 | 1423.75 | -0.47 | 5A | 4 | 16 | 2 | 2.81 [187] | 2.81 | 0.5 | 0 |
| 1M | 4 | 16 | 2 | 3.55* [144] | 3.55 | 0.16 | 0 | 6A | 10 | 40 | 5 | 6.48 [177] | 5.77 | 56.66 | 10.96 |
| 2M | 4 | 16 | 2 | 3.58* [144] | 3.58 | 3.25 | 0 | 7A | 10 | 40 | 5 | 5.71 [187] | 5.7 | 160.85 | 0.18 |
| 3M | 4 | 16 | 2 | 3.33* [144] | 3.33 | 1.05 | 0 | 8A | 10 | 40 | 5 | 6.52 [187] | 6.51 | 86.08 | 0.15 |
| 4M | 4 | 16 | 2 | 5.67* [144] | 5.67 | 0.81 | 0 | 9A | 16 | 64 | 8 | 8.51 [187] | 8.5 | 116.37 | 0.12 |
| 5M | 4 | 16 | 2 | 3.53* [19] | 3.53 | 0.51 | 0 | 10A | 16 | 64 | 8 | 6.31 [187] | 6.31 | 676.65 | 0 |

instance for which the given method reported the best solution that all other solution approaches failed to reproduce. Regarding computation time, the multistage algorithm is slower compared to methods developed by [151, 187, 191], especially on instances with 16 vehicles. For example, the ALNS proposed by [187] did not run longer than 45 seconds for any problem instance, whereas the multistage algorithm, in the worst case, ran for 45 minutes (8L). Nonetheless, the multistage algorithm solves a general-purpose formulation using a standard CP solver. Therefore, it offers the flexibility to model new features at the level of the formulation without changing the solution algorithm. This capability is explored in more detail in the next section. In contrast, it is not immediately apparent how the metaheuristics developed deliberately for solving VRPTWSync, e.g., [151, 187, 191], could be extended to accommodate new features, in particular scheduling breaks.

## 6.5 Home Care Routing and Scheduling Problem

This section presents the application of the multistage optimisation algorithm developed in Section 5.5 for solving authentic VRPSC instances of personnel routing and scheduling encountered in operations of a real-world home care organisation. The multistage algorithm will solve the CP formulation outlined in Section 5.3. In contrast to the VRPTWSync example presented before, which ignored the aspects of contractual working hours, scheduling breaks, the continuity of visits and skill-based routing, such side constraints are now taken into account. Besides the total travel time minimisation, alternative definitions of the objective functions that focus on reducing the number of vehicles and minimising the delays in commencing visits caused by disruptions are also considered. They were introduced in Section 5.5.4.

### 6.5.1 Problem Instances

The problem instances were created from a data set of visits delivered throughout the year 2017 by the largest home care provider in Scotland. The company employs approximately 2,700 carers who provide services to roughly 6,000 patients. Overall, the company delivers around 95,000 visits every week. The operational area is divided into 27 non-overlapping districts that do not differ significantly in the number of visits scheduled every day. Each district is an autonomous entity with an independent pool of carers for exclusive use.

A record representing a visit in the data set includes the postal address of the patient's home, the tasks to be executed during the visit, its planned start time, estimated duration, and the carer assigned by a human planner. Furthermore, each record is complemented by information collected during the execution of a schedule by a mobile application installed on smartphones used by carers. The most relevant data are the check-in and check-out times. They indicate the precise time when the visit was commenced and its real duration.

The problem instances for daily scheduling in the largest district were created using the following procedure. For a given day, the visits executed in the district were extracted. Relying on the carers assigned by human planners, working hours and contractual breaks for each carer who was available that day were inferred. It is a common way of extracting carers' working hours in the literature, see, e.g. [153, 154]. Skills were defined based on the list of tasks planned for the visit. If a human planner assigned a given carer to execute a visit that contained a specific task, the carer is assumed to have relevant skills to perform the task. Travel times between visits' locations were estimated using the Open Source Routing Machine [200]. The routing engine was configured to use the official map of the city where the home care provider operates. The configuration parameters created by the authors of the project for pedestrians were adopted. Technically, the travel time is computed based on the shortest path a pedestrian could follow and adjusted accordingly to account for the road infrastructure, i.e., extra waiting time is induced by the need to stop at the traffic lights, etc. The travel time estimation procedure requires providing the initial and the target location and using GPS coordinates. For that reason, postal addresses were translated to GPS coordinates (a.k.a. geocoding) using the Google Maps Platform [201].

To reproduce the setup used by the home care organisation during the pilot study, 14 problem instances that cover the first two weeks of October 2017 were created. The selected period does not overlap with a holiday season and therefore is arguably representative for the remaining weeks of the year. Other researchers also conducted computational studies in October, see, e.g. [154]. The district with the largest number of visits to schedule was selected to simulate the most computationally challenging setting. The preliminary results for other districts are comparable.

Table 6.3 contains the size of the problem instances labelled by the day of the month. Subsequent columns indicate the number of carers available the given day ($C$), the number of single carer visits ($V^1$), and the visits with pairwise synchronisation constraints ($V^2$). Every two visits in the last group correspond to one 'regular' visit from the application domain, which should be staffed by two carers. Overall, depending on the day, between 47 and 71 carers were available to cover from 559 up to 628 visit assignments. The problem instances in the anonymised form are available online [26] for testing and benchmarking purposes.

Table 6.3: Size of the HCSRP problem instances.

| Day | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $C$ | 47 | 62 | 60 | 63 | 62 | 64 | 47 | 47 | 58 | 60 | 58 | 71 | 58 | 53 |
| $V^1$ | 338 | 389 | 386 | 388 | 388 | 393 | 343 | 336 | 387 | 375 | 380 | 382 | 383 | 329 |
| $V^2$ | 224 | 232 | 240 | 240 | 232 | 224 | 224 | 226 | 224 | 236 | 232 | 234 | 218 | 230 |

The following parameters for solving the HCSRP instances were used. Start times for visits and breaks can be delayed or expedited no more than 90 minutes from their nominal values. Every carer has an overtime allowance of 60 minutes per day. At most 30 minutes of this period can be spent before the planned start of the shift. The remaining 30 minutes are available to prolong the shift. A fixed penalty is incurred for each 'regular' visit declined. Hence, if two visits bounded by a synchronisation constraint are declined, the penalty is incurred once because the pair corresponds to one 'regular' visit that requires two vehicles. The penalty is set to the sum of five longest travel time distances between any pair of visits' locations in the problem instance. The value is high enough to prevent situations in which a reduction of the total travel time is the sole motivation for declining a visit. The continuity of visits requirement is set to the same target human planners deliver. Therefore, at most two different carers per day can perform visits for a client who does not have visits that require two carers simultaneously. If such visits are provided for the client, then at most four different carers can be dispatched to visit the client on a given day. For the carer reduction in the third stage, the cost of a carer is equivalent to the total duration of the carer's shift minus contractual breaks in seconds. In the remaining configurations, there is no cost associated with using a carer.

### 6.5.2 Multistage Optimisation Algorithm

This section compares the computational approach of solving the CP formulation presented in Section 5.3 directly versus solving the formulation in three stages using the multistage optimisation algorithm, which is the subject of Section 5.5. The minimisation of the total travel time is the objective function. Other decision criteria discussed in Chapter 5, the workforce reduction and the riskiness index optimisation, rely on information collected in previous stages. For that reason, they are not available when the formulation is solved directly. The comparison between the computational approaches aims to surface the limitations of solving the CP formulation directly and justify the development of the multistage optimisation algorithm.

In contrast to the settings used for solving the VRPTWSync benchmark, here optimisation is stopped due to a lack of progress in the third stage after 5 minutes since the last improvement of the incumbent solution. The analogous time limit while solving the CP formulation stand-alone is set to 20 minutes. All remaining parameters assume the same values.

The performance of both computational approaches is presented first on one problem instance. The demonstrated example displays a general trend observable for all other problem instances. Figure 6.1 presents the objective function (i.e., the total travel time in seconds plus the penalty for declined visits) and the number of declined visits over computation time for solutions found by solving the CP formulation explained in Section 5.3 for Problem Instance 1 either directly by the stand-alone CP solver [17] or in three stages using the multistage optimisation algorithm presented in Section 5.5. The value of the penalty incurred for a declined visit can be found in Table 6.4.

The scheduling problem in the first stage was straightforward to solve. The parallel cheapest insertion heuristic constructed the initial feasible solution within milliseconds. The solution initialised LS, which reduced the total travel time until convergence to the local optimum. Overall, the first stage optimisation was completed in less than one second. During the transition between the first and the second stage, the cost of the

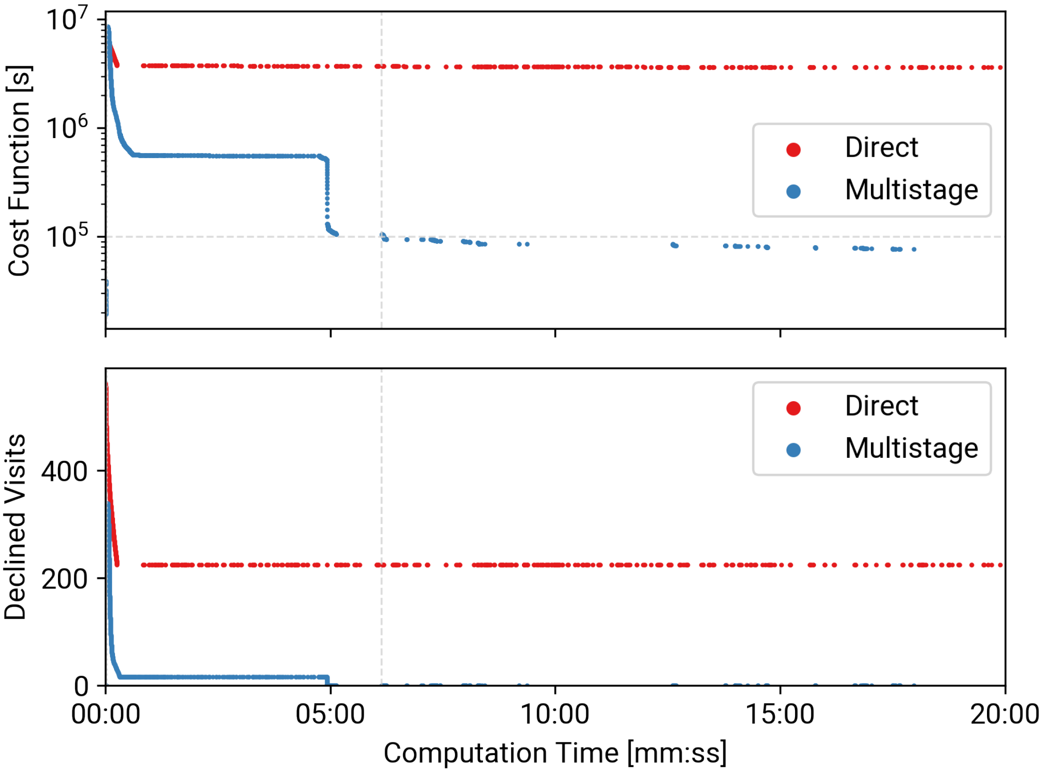


Figure 6.1: The cost function in seconds and the number of declined visits for solutions evaluated while solving the CP formulation for Problem Instance 1 by the multistage optimisation algorithm and using the CP solver stand-alone. The cost function is plotted on the logarithmic scale to display the pace of the cost reduction at a given stage. The dashed vertical line indicates the transition between the second and the third stage. The transition from the first to the second stage is not discernible because it happened in the first second of optimisation. The dashed horizontal line emphasises the merit of optimisation performed at the third stage to further decrease the cost of a solution.

initial solution increased due to declined single carer visits whose number was multiplied by the penalty factor. However, shortly after the warm start with the supplied initial guess, the second stage significantly reduced the number of declined visits. The optimisation continued for 5 minutes until it reached a local optimum. Predictably, it was premature, and the third stage decreased the objective value further using the GLS meta-heuristic and additional search operators. The third stage reduced the objective function by more than 25% compared to the cost of the best solution computed by the second stage.

The attempt to solve the formulation using a stand-alone CP solver did not yield satisfactory results. The solver did not find any solution with less than 200 visits declined despite running for more than 4 hours. In striking contrast to the direct approach, the multistage optimisation algorithm found better schedules in less than 5 seconds. Visits with pairwise synchronisation constraints caused the lack of convergence while solving the formulation directly. Relaxing the synchronisation constraints or supplying a feasible initial guess that satisfies the synchronisation constraints resolves the computational bottleneck. As the following results indicate, the issues discussed are part of a general trend apparent while solving other problem instances.

Figure 6.2 displays aggregate results obtained by solving the CP formulation presented in Section 5.3 for all problem instances using the multistage optimisation algorithm proposed in Section 5.5. The objective to minimise was the sum of the total travel time and the penalty for each declined visit.

The precise duration of the second and the third stage optimisation depends on the problem instance. However, there are no apparent differences in the overall trend of computations. The first stage optimisation was completed within a few seconds. Immediately after the transition to the second stage, the cost rapidly increased due to declined single carer visits considered in the problem instance and ignored by the first stage. However, the number of declined visits was significantly reduced shortly after the start of the LS. Overall, the second stage required between 5 and 30 minutes to converge to a local optimum. The third stage optimisation managed to improve the solution even further. Additional cost reductions were obtained by saving travel time. Understandably, only small improvements were made in the final phase of the optimisation. Nonetheless, the extra computation time effort was optional, and the optimisation can be stopped as soon as the cost-effectiveness of the incumbent solution becomes acceptable for the decision-maker.

Figure 6.3 illustrates the overall results for all problem instances obtained using the CP solver directly for solving the CP formulation introduced in Section 5.3 with the minimisation of the total travel time and penalties for declined visits as the objective

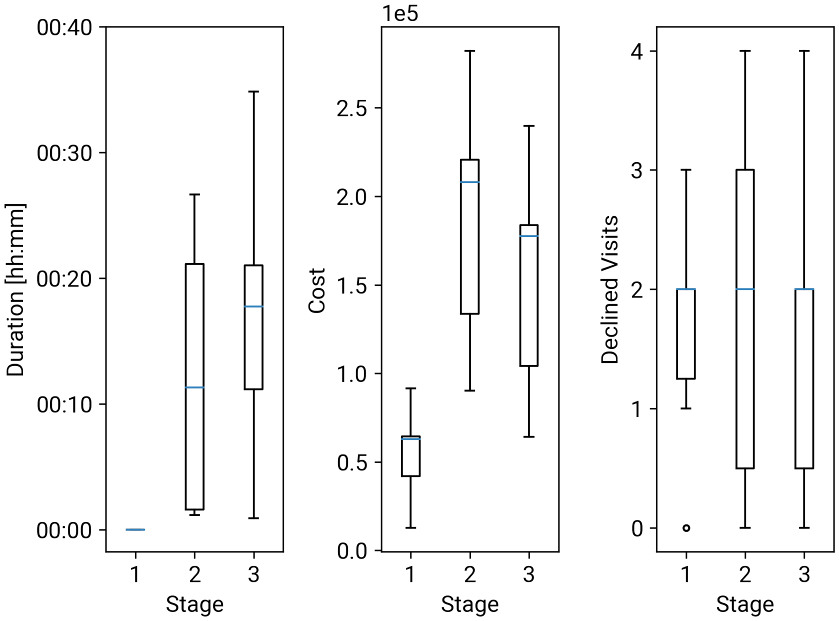


Figure 6.2: Computation time, the cost of the best solution in seconds and the number of declined visits aggregated for each stage of the multistage optimisation algorithm for all problem instances. Boxes indicate data points between the first and the third quantile. The line segment which divides the box represents the median. Whiskers denote the range between 0.05 and 0.95 percentile (boundaries included). Points, if present, indicate outliers.

function.

For every problem instance, the CP solver left at least 200 visits unassigned in the final schedule despite long computations, which lasted 8 hours in the worst case.

Overall, the tests performed using the multistage algorithm provide very encouraging evidence that the computational approach is suitable for solving scheduling problems arising in practice. On the other hand, using the CP formulation directly is prone to fail due to the difficulty in staffing visits with partial synchronisation constraints, which is effectively resolved by the first stage of the multistage algorithm.

The next subsection explores the opportunity to modify the objective function in the third stage and perform either the minimisation of the number of carers or the optimisation of the riskiness index. The resultant schedules are also compared with human planners.

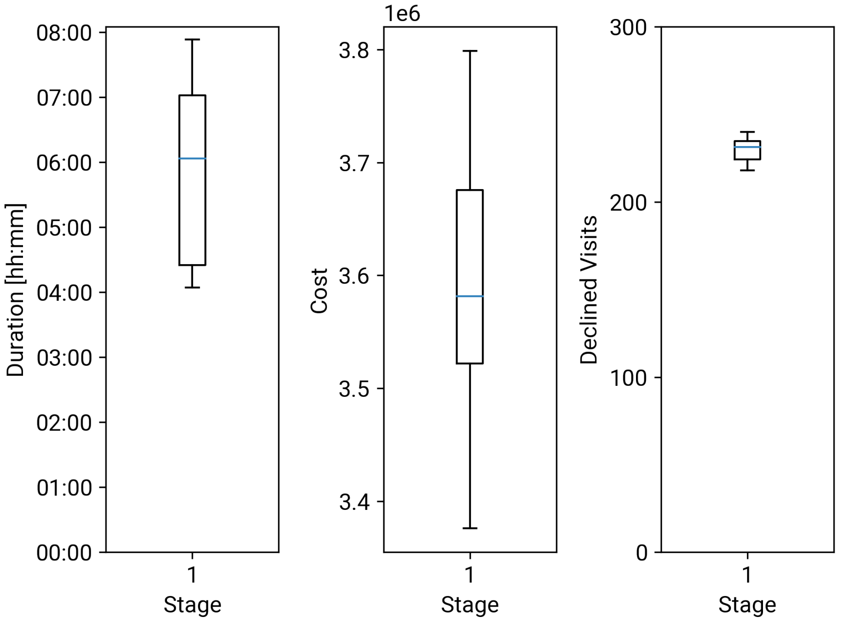


Figure 6.3: Computation time, cost function in seconds and the number of declined visits for solving the CP formulation directly aggregated for all problem instances. The guidelines adopted to draw boxplots are explained in the caption of Figure 6.2.

### 6.5.3 Comparison with Human Planners

The section compares solutions for all problem instances outlined in Table 6.3 computed by solving the CP formulation using the multistage optimisation algorithm with the schedules created by human planners for the home care organisation. The CP formulation and the multistage optimisation algorithm were presented in the previous chapter in Sections 5.3 and 5.5, respectively. The solutions will be assessed by evaluating the sum of the total travel time and penalties for declined visits.

Human planners build schedules manually. The initial assignment between visits and carers is made based on a template schedule, in which the same carer performs all visits for a given client. In practice, staffing all requested visits using baseline carers is seldom possible as visits are delivered in a wide time range (i.e., between 4 AM and 11 PM). If the baseline carer from the template schedule is not available (e.g., annual leave, different shift, etc.), the visits are marked as unassigned. Human planners delegate such visits to other available carers. If no backup assignments are readily apparent, then the planners make local adjustments to create them (e.g., by moving visits between carers, altering time windows or duration of a visit, relocating carers between areas, requesting externals contractors, etc.). As a last resort, the planners tentatively outsource unstaffed visits to locum carers. When making these decisions, human planners rely mainly on their knowledge and experience. Their decision support tools are limited to a dashboard that reports the total number of visits assigned to each carer available in the specified area on a given day.

Table 6.4 compares the schedules created by human planners with the schedules found by the multistage algorithm by solving the CP formulation with travel time optimisation in the third stage. The objective for the schedules created by planners and by the multistage optimisation algorithm obtained in the second and the third stage for all problem instances are tabulated. Besides the objective function, the principal components that influence the cost of the solution are presented: the number of declined visits and the total travel time.

The cost of schedules produced by the multistage optimisation algorithm is several times lower compared to human planners. A significant cost reduction is available already after the second stage is terminated. The schedules created at this point have approximately three times fewer declined visits compared to human planners, and the total time spent on travelling is reduced at least twofold. In the worst case, the second stage required 27 minutes of computation time. The third stage can reduce the cost of a schedule even further at the expense of additional computation time. The savings always come from finding more efficient routes and occasionally also from scheduling more visits.

Depending on the problem instance, the schedules produced by human planners and the multistage optimisation algorithm had some declined visits. Nonetheless, the optimised schedules staff considerably more visits. The first-line managers are aware of the oversubscription and compensate for it by moving the unstaffed visits to a different time or outsourcing them to locum carers.

The pronounced cost reduction and the ability to schedule more visits are desirable outcomes demonstrating the usefulness of the methodology developed in Chapter 5.

Table 6.4: Schedules obtained by human planners compared to travel time optimisation using the multistage algorithm. The first column indicates the Problem Instance (I). Information about the size of the problem instance, the number of carers, single carer visits, and visits with pairwise synchronisation constraints are reported in Table 6.3. The following three columns contain the value of the objective function for schedules created by human planners (HP) as well as final solutions obtained in the second stage ($\mathrm{MS}_2$) and the third stage ($\mathrm{MS}_3$). Column ($\rho$) contains the value of the penalty incurred for each declined visit in the problem domain. The following two clusters of columns display the components of the objective function: the number of declined visits (Declined V.) and travel time in hours (Travel Time [h]) reported for human planners and the final solutions found in the second and the third stage. The last cluster of columns (C. Time [s]) contains the computation time in seconds for the second and the third stage optimisation.

| I | Objective | | | $\rho$ | Declined V. | | | Travel Time [h] | | | C. Time [s] | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | HP | $\mathrm{MS}_2$ | $\mathrm{MS}_3$ | | HP | $\mathrm{MS}_2$ | $\mathrm{MS}_3$ | HP | $\mathrm{MS}_2$ | $\mathrm{MS}_3$ | $\mathrm{MS}_2$ | $\mathrm{MS}_3$ |
| 1 | 120.64 | 29.07 | 21.11 | 7.03 | 7 | 0 | 0 | 71.46 | 29.07 | 21.11 | 309 | 1079 |
| 2 | 204.9 | 51.6 | 36.28 | 7.07 | 19 | 3 | 2 | 70.59 | 30.39 | 22.14 | 858 | 2520 |
| 3 | 200.44 | 31.37 | 22.93 | 7.07 | 17 | 0 | 0 | 80.27 | 31.37 | 22.93 | 73 | 1322 |
| 4 | 192.52 | 52.55 | 32.73 | 7.07 | 17 | 4 | 2 | 72.35 | 24.27 | 18.59 | 968 | 2316 |
| 5 | 165.43 | 40.09 | 32.9 | 7.07 | 13 | 2 | 2 | 73.54 | 25.95 | 18.76 | 80 | 1464 |
| 6 | 147.93 | 25.09 | 17.8 | 7.07 | 10 | 0 | 0 | 77.24 | 25.09 | 17.8 | 503 | 1222 |
| 7 | 163.89 | 57.31 | 40.64 | 7.03 | 12 | 4 | 3 | 79.57 | 29.21 | 19.56 | 1335 | 3486 |
| 8 | 94.85 | 45.28 | 35.8 | 7.03 | 2 | 3 | 2 | 80.8 | 24.2 | 21.74 | 93 | 208 |
| 9 | 168.75 | 58.66 | 52.47 | 7.07 | 13 | 4 | 4 | 76.85 | 30.38 | 24.19 | 1534 | 2202 |
| 10 | 159.33 | 41.8 | 37.34 | 6.97 | 11 | 2 | 2 | 82.65 | 27.85 | 23.4 | 96 | 707 |
| 11 | 178.24 | 42.87 | 36.34 | 6.97 | 14 | 2 | 2 | 80.66 | 28.93 | 22.4 | 1541 | 2677 |
| 12 | 200.45 | 47.99 | 38.52 | 6.97 | 17 | 3 | 2 | 81.96 | 27.08 | 24.58 | 1601 | 2735 |
| 13 | 176.86 | 26.72 | 17.98 | 6.97 | 14 | 0 | 0 | 79.27 | 26.72 | 17.98 | 110 | 1042 |
| 14 | 176.64 | 44.91 | 35.04 | 6.89 | 14 | 2 | 2 | 80.14 | 31.13 | 21.26 | 1080 | 2201 |

The home care company reproduced similar results in the pilot study, which confirmed that the optimised schedules are valid and meet the targets set by the organisation (i.e., the continuity of visits, time windows, the overtime allowance).

### 6.5.4 Alternative Objective Functions at the Third Stage

The section presents computational results of using the CP formulation proposed in Section 5.3 with alternative definitions of the objective that emphasise secondary properties of a schedule apart from the total travel time and the number of declined visits which were the primary optimisation objectives. The reduction in the number of carers required to staff the requested visits and increasing the likelihood of delivering visits on time in the face of unexpected disruptions are among desirable features of a solution for a VRPSC instance in the HHC application context. The objective function for minimisation of the total number of vehicles was formulated in Section 5.5.4. The process of increasing resiliency of a schedule against delays in commencing visits due to disruptions (i.e., some visits taking longer than expected) was accomplished by the riskiness index optimisation, which was the subject of Section 5.4.

The section compares the solutions to the HHC problem instances presented in Table 6.3 computed by solving the CP formulation using the multistage optimisation algorithm with different variants of the objective function at the third stage: the sum of the total travel time and penalties for declined visits, minimisation of the number of carers required to run the schedule, and the optimisation of the ERI of a delay in commencing a visit. Regardless of the third stage configuration, the minimisation of the total travel time and penalties for declined visits was the objective function of the CP formulation solved in the first two stages.

Table 6.5 illustrates various performance indicators which influence the cost of the schedule found by solving the CP formulation by the multistage optimisation algorithm depending on the choice of the objective function at the third stage. The total travel time in hours, the number of carers who execute the schedule, the maximum average delay in commencing a visit in minutes, and the number of declined visits are reported for every third stage configuration and every problem instance.

Figure 6.4 displays results from Table 6.5 in the aggregated format.

Predictably, the travel time optimisation yielded the best results regarding the total time spent on travelling (i.e., between 18 and 25 [h] depending on the problem instance). The ERI optimisation obtained the second-best schedules according to this criterion (i.e., between 20 and 27 [h]). The carer reduction closed the ranking (i.e., between 24 and 29 [h] spent on travelling). However, the configuration with the ERI optimisation in the third stage delivered considerably fewer visits (i.e., the number of declined visits is 23 in the worst case compared to at most 4 in other configurations). Hence, it is likely that such a schedule would require less travelling.

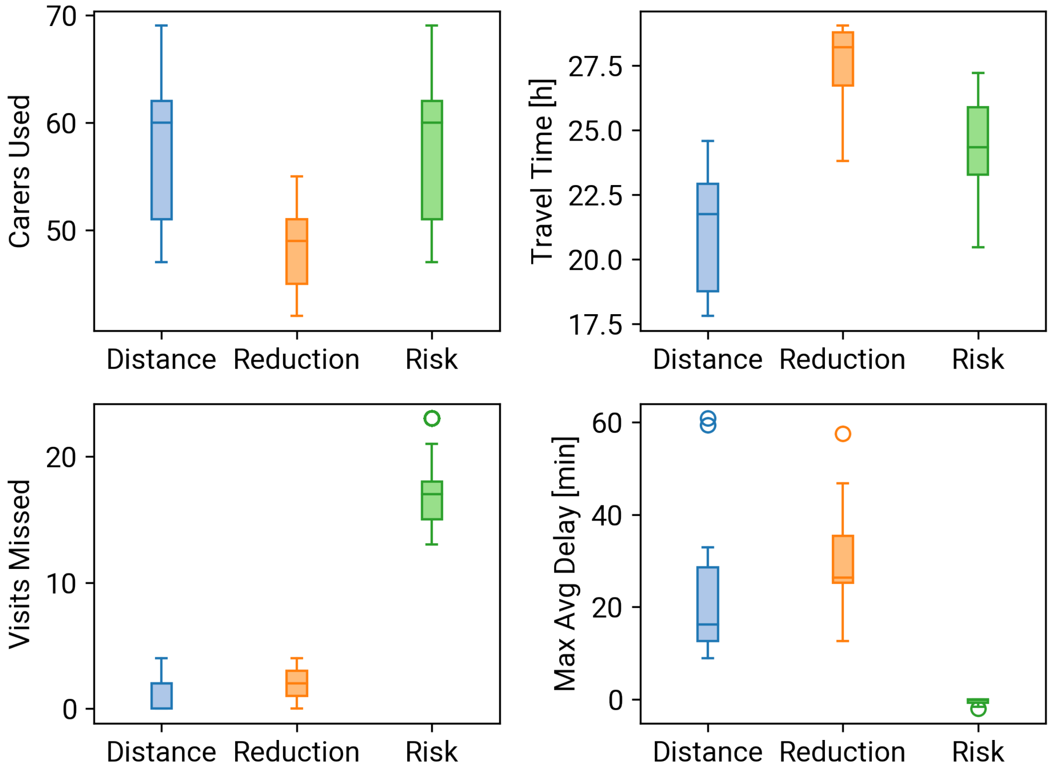
Table 6.5: Comparison of final solutions reported for each configuration of the three-stage optimisation: travel time optimisation (Dist), carer reduction (Red) and the riskiness index optimisation (Risk). The first column indicates the problem instance (I). The following clusters of columns report the properties of the final solutions: the total travel time in hours (Travel Time [h]), the number of carers used (Carers Used), the maximum of the average delay for each visit scheduled (MA Delay [min]), and the number of declined visits (Declined Visits). Figure 6.4 displays the results in the aggregated format.

| I | Travel Time [h] | | | Carers Used | | | MA Delay [min] | | | Declined Visits | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Dist | Red | Risk | Dist | Red | Risk | Dist | Red | Risk | Dist | Red | Risk |
| 1 | 21 | 24 | 26 | 47 | 43 | 47 | 18 | 26 | -2 | 0 | 0 | 17 |
| 2 | 22 | 27 | 25 | 62 | 51 | 62 | 13 | 47 | 0 | 2 | 3 | 21 |
| 3 | 23 | 29 | 27 | 60 | 50 | 60 | 61 | 57 | 0 | 0 | 1 | 16 |
| 4 | 19 | 27 | 22 | 63 | 53 | 63 | 10 | 13 | -1 | 2 | 2 | 23 |
| 5 | 19 | 29 | 24 | 62 | 49 | 62 | 9 | 26 | 0 | 2 | 3 | 15 |
| 6 | 18 | 29 | 22 | 64 | 52 | 64 | 9 | 14 | -1 | 0 | 1 | 19 |
| 7 | 20 | 25 | 25 | 47 | 43 | 47 | 16 | 27 | 0 | 3 | 3 | 18 |
| 8 | 22 | 24 | 20 | 47 | 42 | 47 | 14 | 38 | -2 | 2 | 3 | 16 |
| 9 | 24 | 28 | 27 | 58 | 49 | 58 | 17 | 30 | 0 | 4 | 4 | 14 |
| 10 | 23 | 29 | 24 | 60 | 51 | 60 | 24 | 21 | 0 | 2 | 2 | 13 |
| 11 | 22 | 29 | 26 | 56 | 45 | 56 | 33 | 36 | 0 | 2 | 2 | 14 |
| 12 | 25 | 29 | 26 | 69 | 55 | 69 | 12 | 25 | 0 | 2 | 4 | 17 |
| 13 | 18 | 28 | 23 | 58 | 45 | 58 | 30 | 25 | 0 | 0 | 1 | 18 |
| 14 | 21 | 27 | 24 | 51 | 49 | 51 | 59 | 32 | 0 | 2 | 2 | 17 |

Unsurprisingly, the formulation with the carer reduction built schedules with the fewest staff members (i.e., between 2 and 14 carers less depending on the problem instance). The remaining configurations do not consider the cost of using a carer and employ the maximum number of carers available.

Although the average delay for most visits was negative for every third stage configuration (i.e. visits were delivered on time), the whiskers representing the maximum average delay are always below zero only for the riskiness index optimisation. Conversely, for the travel time optimisation and carer reduction, some clients experience delays on average. Obtaining a schedule in which every visit staffed was performed on average without delays was possible through declining more visits compared to formulations with the carer reduction and travel time optimisation. Otherwise, staffing more visits caused some clients to experience excessive delays on average (i.e., approximately

Figure 6.4: Comparison of final solutions for each configuration of the third stage optimisation. Individual plots display the total travel time, the number of carers used, the maximum average delay for a customer and the number of declined visits aggregated for all problem instances. The guidelines adopted to draw boxplots are explained in the caption of Figure 6.2.

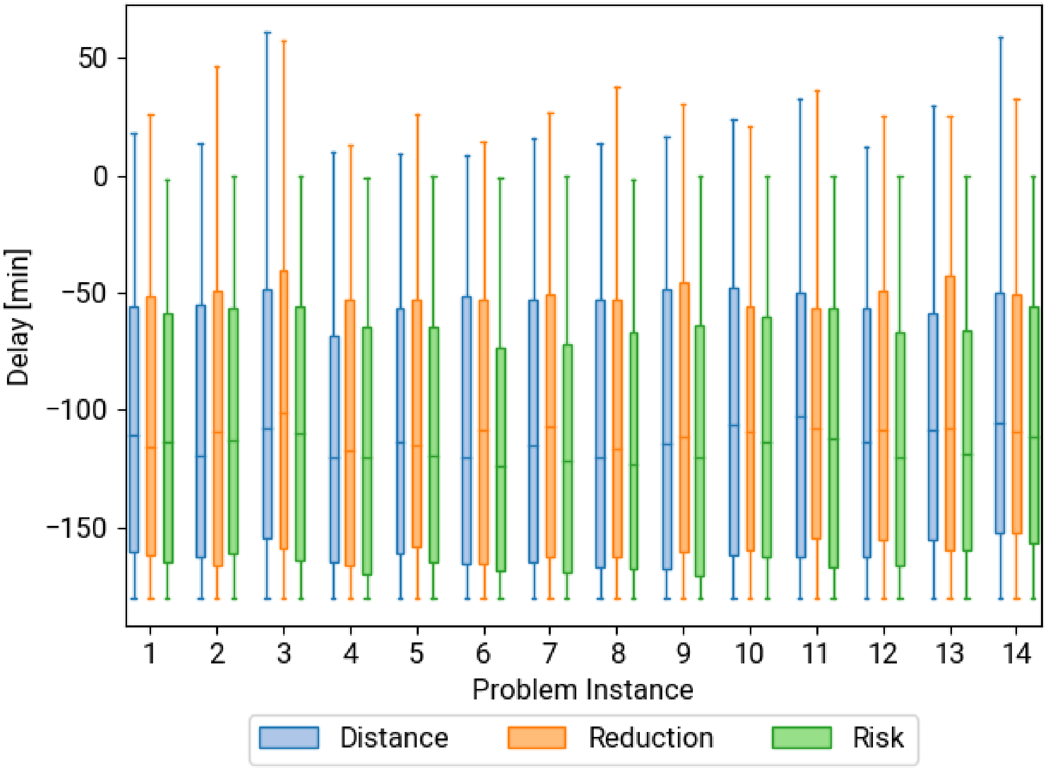


10 minutes to one hour in the worst case).

Regarding the declined visits, the ERI optimisation did not perform visits for which the on-time provision of services was not achievable for the solver. The travel time optimisation was the most successful in staffing as many visits as possible. Its advantage over the carer reduction arose in the third stage because it was more important to release carers in the latter configuration. For example, in instance six, releasing ten carers by the formulation with carer reduction was at the expense of additional 11 hours of travel time for the remaining carers and one more declined visit. Carers who are not used in the schedule could be relocated to other operational areas. Otherwise, their shift patterns could be adjusted to cover the declined visits.

Figure 6.5 illustrates a box plot displaying the distribution of average delays for all visits scheduled in the final solutions of every problem instance presented in Table 6.3. The solutions were computed by solving the CP formulation using the multistage optimisation algorithm with alternative configurations of the third stage: minimisation of the total travel time and penalties for declined visits, reduction of the total number of carers needed to staff required visits, and the riskiness index optimisation of a delay in commencing a visit.

Figure 6.5: Maximum average delay in commencing a visit in minutes for every problem instance and all configurations of the third stage optimisation: the total travel time optimisation (Distance), the minimisation of the number of carers (Reduction), and the riskiness index optimisation (Risk). The guidelines adopted to draw boxplots are explained in the caption of Figure 6.2.



Overall, the average delay for most visits is negative (i.e., visits are performed on time) regardless of the third stage configuration, which is desirable. Moreover, medians and the regions between the second and the third quartile are located in similar areas. From the definition of the ERI, whiskers representing the maximum average delay are always below zero. For other configurations, the top-most whiskers are positive and indicate the degree of the most excessive delays customers experience.

## Concluding Remarks

Computational results obtained using the multistage optimisation algorithm developed for solving the VRPTWSync and the VRPSC are the primary contributions of this chapter. Firstly, it was shown that the multistage algorithm remains competitive with the most successful solution methods presented in the literature regarding the quality of the solution. In particular, new best solutions were found for five problem instances from the popular VRPTWSync benchmark [144]. Furthermore, the multistage optimisation algorithm solved considerably larger VRPSC problem instances than those discussed in the literature on visits with pairwise synchronisation constraints (i.e., see Table 6.1). The VRPSC instances were not made artificially large but are motivated by a real-world application in HHC, and therefore include additional domain-specific constraints, such as the continuity of visits and the scheduling of contractual breaks.

The multistage optimisation algorithm solved the VRPSC instances in attractive computing time (i.e., less than half an hour for the second stage optimisation). The results agreeably demonstrate that large routing and scheduling problem instances with synchronised visits can be practically optimised using the software. Doing so does not negatively impact the target continuity of visits human planners deliver. Apart from the automation, the additional advantage is a considerable reduction in the travel time due to routing optimisation. Consequently, the board of the home care company is leading the adoption of scheduling optimisation in the daily operations of the organisation.

Besides the minimisation of the total travel time, the opportunity to reduce the number of carers and to optimise the ERI of a delay in the third stage optimisation solving the same suite of problem instances were explored in the chapter. Overall, the riskiness index optimisation prevented the average delay in commencing a visit from becoming positive at the expense of leaving more visits declined. Previously, the riskiness index optimisation was limited to problem instances with at most 100 visits and no synchronisation constraints [16].

# Chapter 7

# Conclusions

The final chapter revises the contributions of the thesis and explains how they address the research aims and objectives. The work is critically assessed by evaluating the significance of the key findings and discussing the limitations of the adopted approach. Other possible applications of the methodology and suggestions regarding future research directions conclude the chapter.

## 7.1 Contributions

The primary goal is to develop methods for solving vehicle routing and scheduling problems whose optimisation models contain parameters affected by uncertainty. The methodology should enable solving large problem instances from the real world without introducing simplifying assumptions that could negatively impact the practical utility of the solutions. Two target applications are featured to simplify the exposition.

The first application is scheduling satellite space-to-ground optical communication for the purpose of the SatQKD. As an emerging application driven by the increasing adoption of optical communication with the LEO satellites, it was demonstrated only in scientific experiments and was not studied by the optimisation community. Nonetheless, as the technology matures, commercial communication systems should benefit from optimisation during the design phase, which motivated the interest in this topic.

The work on the thesis leads to the formulation of the deterministic variant of the

SatQKD problem for the first time in the literature and proving its computational complexity. The latter reveals that the deterministic SatQKD problem does not have a FPTAS. Therefore, looking for an efficient algorithm to compute an optimal solution for an arbitrary SatQKD problem instance is futile.

On the other hand, the linear relaxation of the MIP model with time discretisation proposed in the thesis for solving the SatQKD problem is strong. Hence, although the deterministic SatQKD problem is NP-hard, its instances considered in the thesis are solved to optimality in reasonable computing time by a commercial-grade MIP solver. In particular, the gap between the incumbent solution and the LP relaxation is below 1.5% for the majority of problem instances.

The ability to find optimal schedules enables computing the best possible performance a given SatQKD communication system could achieve, assuming weather conditions are known precisely. The computational study contains such analysis for a potential design of a SatQKD communication system with one satellite and a fixed network of ground stations located in the mainland UK. The study proposes the configuration of orbital parameters for the satellite that maximises the number of keys possible to deliver to the network. After tuning the initial RAAN using the grid search, the spacecraft could distribute two times more keys to the system compared to the least efficient configuration. Given the orbital parameters of the satellite that enable the best performance and historical weather observations, one could compute the maximum number of keys each ground station could consume weekly without depleting its key buffer. The key consumption rate provides insight into the maximum performance the communication system could possibly attain. It is valuable information to decision-makers who could determine whether the design of the communication system and the status of the technology satisfy the estimated demand for keys given the cost of the system.

Besides key performance indicators, the study reveals that the Sun constantly illuminates the satellite during communication windows with ground stations located at high latitudes for several weeks during summer. Such conditions preclude establishing space-to-ground communication. This phenomenon is observed because the key transmission rate analysis, besides cloud cover conditions and the elevation angle between the ground station and the satellite, accounts for the spacecraft's illumination. The latter has a pivotal role during the summer months in modelling the transfer rate for ground stations located in the UK. Ignoring this effect in modelling would lead to an oversimplification valid only from late autumn to early spring.

Apart from the application in the performance analysis of a SatQKD communication system, the MIP formulation with the time-discretisation is an entry point to develop more advanced models with the treatment of cloud cover uncertainty. To the author's best knowledge, they are the first models for scheduling optical space-to-ground communication in uncertain cloud cover conditions proposed in the literature. A suite of alternative formulations that are parametrised using cloud cover predictions from official weather forecasts is proposed. It includes a DRO model with mean and standard deviation as well as Stochastic Optimisation models that employ the CV@R and the ERI optimisation. The application of the riskiness index optimisation in scheduling satellite operations is new.

The models developed according to the DRO and SO paradigms are complemented by a RO formulation with a novel uncertainty set that combines the VAR model and a box uncertainty set. It is developed to represent spatial and temporal correlations naturally observed in the behaviour of weather conditions. Expressing correlations in the uncertainty set reduces the conservativeness of the model. Although a generic polyhedral uncertainty set is well known, to the best knowledge of the author, a time series model has never been used before to derive an uncertainty set.

The proposed formulations for the SatQKD problem with uncertain cloud cover are compared on a data set of problem instances built using official weather forecasts. Then, the effectiveness of obtained solutions is assessed using actual weather conditions observed in practice when the schedule would be executed. The VAR and the mean-standard deviation models are the best stand-alone formulations in the comparison considering all problem instances. Further improvement is possible by selecting a model from the suite of available formulations based on cloud cover conditions predicted by the weather forecast.

Overall, the study demonstrates the quantitative advantage of using models with the treatment of uncertainty compared to deterministic models. Consequently, practitioners should expect an improvement in the performance of schedules by switching to the models with the treatment of uncertainty. Furthermore, although the formulations are developed for the SatQKD, which is a specific problem, no self-evident circumstances are preventing the application of the modelling techniques for the treatment of cloud cover uncertainty to other optimisation problems that have a MIP formulation.

The second part of the thesis is focused on developing and validating the methodology for solving large VRPTWSync problem instances in which some visits require the simultaneous presence of two vehicles. The problem is a generalisation of the HCSRP which is a critical application regarding its societal impact, the scale of operations and the cost for the public sector.

The featured solution method is a multistage optimisation algorithm built on top of the CP solver [79] which combines constraint propagation with LS and metaheuristics. Such a computational approach is adopted because deterministic optimisation models for the VRPTW can solve to optimality only instances that contain up to 150 visits [169]. If pairwise synchronisation constraints are present, the size of the problem instances solved using exact methods is even smaller, see Table 6.1. The multistage design is motivated by unsatisfactory results of solving the formulation with pairwise synchronisation constraints using the CP solver stand-alone. Once the first stage finds an initial solution for visits with synchronisation constraints, the second stage can successfully improve that solution and compute the allocation of vehicles for the remaining visits.

As of this writing, the multistage optimisation algorithm is a competitive computational approach for solving the deterministic variant of the VRPTWSync. The algorithm reproduces the majority of the best results for the suite of benchmark problems [144] (39 out of 50 instances) and strictly improves the objective for five cases. Reporting the best solutions is critical because researchers use benchmark problems to validate their ideas and propose new optimisation methods. Furthermore, since the algorithm employs a generic CP solver, it is straightforward to introduce additional side constraints required for a specific application, i.e., the continuity of visits, the skill matching and the treatment of contractual breaks for the HCSRP.

The extensibility of the multistage algorithm is demonstrated by solving the HCSRP instances obtained from the largest home care organisation which operates in Scotland. Regarding the number of visits, these instances are the biggest problem instances considered in the academic literature. The multistage algorithm can solve each instance in less than one hour. The proposed method consistently schedules more visits than human planners and reduces the total travel time approximately threefold. The significant reduction in the total travel time has no negative impact on the quality of the solution. The home care provider reproduced the computational results during the pilot deployment, and the organisation is now adopting optimisation tools for scheduling operations.

The computational study demonstrates using practical examples that significantly larger VRPTWSync instances are effectively solvable compared to the problem instances considered in the literature. Obtaining such results does not require introducing simplifying assumptions that could render the solutions unrealistic. The formulation contains constraints enforcing the treatment of contractual working hours and breaks. The same applies to skill matching and the continuity of visits. Furthermore, a routing engine is employed for the accurate estimation of travel time. Eliminating any of these features could result in creating invalid solutions that cannot be executed in practice.

Besides the ease of including additional constraints in the formulation, the multistage design allows for modifying objective function between stages, i.e., to reduce the number of carers required to run the schedule or to minimise delays in commencing visits later than their time windows. For the latter objective, the methodology postulates applying the ERI optimisation of delays using the set of scenarios containing visits' durations observed in the past. Apart from minimising the probability of commencing a visit late, such an approach makes the likelihood of a delay inversely proportional to its magnitude. Even though visits with pairwise synchronisation constraints complicate the evaluation of a delay, the methodology presents an algorithm to compute the riskiness index of delays for each visit in polynomial time in the number of scenarios. The algorithm is integrated with the CP solver as a global constraint and tested on the suite of the HCSRP instances.

Overall, the computational study exemplifies that the proposed computational approach built on top of the CP paradigm, initially designed to solve deterministic DO problems, is expandable to optimisation models with the treatment of uncertainty using a finite set of scenarios. The latter could be accomplished by the riskiness index optimisation, which is scalable enough to solve large HCSRP instances.

To conclude, the contributions mentioned above provide strong evidence suggesting that the aims and objectives set for the thesis are met. A strong emphasis on applying the results of this research project in the real world is a desirable byproduct supporting the contributions to science and methodology. Articles describing the multistage optimisation algorithm [23], the deterministic SatQKD scheduling [20], and models for the treatment of cloud cover uncertainty in the SatQKD scheduling [112] were published in academic journals.

## 7.2 Constructive Criticism of Limitations

Regardless of the effort and best intentions, this project is affected by some limitations explained below.

Firstly, the VRPTWSync and the SatQKD optimisation problems are solved using different computational approaches: MIP and CP, respectively. Not applying a uniform approach to both problems could be the reason for criticism.

The decision to employ alternative solution approaches was made following careful consideration to use the most suitable computational method available for the given problem. The preliminary experiments with the MIP model for solving the VRPTWSync confirmed that only small problem instances (i.e., 50 visits) are effectively solvable within a few hours of computing time. Results that corroborate these findings are widely reported in the academic literature [169]. Therefore, rather than applying decomposition to solve a collection of instances independently, the CP paradigm was used instead.

Conversely, the SatQKD problem instances could be solved using the CP solver, which would remain consistent with the solver technology applied to the VRPTWSync.

Then, after the deterministic variant of the problem is solved, the treatment of uncertainty could be implemented using the riskiness index optimisation over scenarios. The sequence of steps outlined above used to be the plan to approach the optimisation of the SatQKD problem. Nonetheless, this path was abandoned after obtaining the preliminary optimisation results for the deterministic problem, which were suboptimal. On the other hand, the LP relaxation of the SatQKD problem had a strong bound, and the MIP solver did not require extensive computational effort to find provably optimal solutions. Since problem instances can be solved to optimality by an exact method, applying heuristic approaches is not justifiable. Other researchers drew similar conclusions selecting the same optimisation technology for scheduling satellite operations [96, 97].

Consequently, by formulating the SatQKD as a MIP and proposing alternative models for the treatment of uncertainty, the thesis contributed to the methodology that is actively used. Furthermore, the additional advantage of applying the exact method is the opportunity to employ and extend the SO, RO and DRO frameworks. Arguably, the treatment of optimisation under uncertainty without these paradigms would not be comprehensive.

Another limitation concerns SO models with uncertain cloud cover developed for the SatQKD problem. They are formulated as an optimisation over scenarios assuming the discrete uniform distribution. A more advanced approach could employ DRO using the Wasserstein ambiguity set. This way, resorting to scenario generation and assuming a specific distribution could be avoided. The DRO model for the riskiness index optimisation using the Wasserstein ambiguity set and the $L_1$ norm was developed by [16].

## 7.3 Other Applications

The featured applications set the context for the methodology developed in the thesis and provided means to validate it. However, the list of potential applications that could benefit from the proposed solution methods and optimisation models is not limited to those considered in the computational study.

The models developed for scheduling SatQKD with cloud cover uncertainty are generic and portable to other MIP formulations with time discretisation. In particular, the proposed optimisation models appear to be transferable with minimal modifications to optimisation problems that involve scheduling operations of spacecraft whose orbits do not change, and tasks to execute are affected by cloud cover. For instance, the MaxPDT problem [92] with optical communication matches these criteria.

The multistage optimisation algorithm could solve any vehicle routing problem with pairwise synchronisation constraints. The example with the HHC application presented in the computational study highlights the ease of defining additional constraints. Hence, another possible extension of the generic formulation proposed for the VRPTWSync problem should not be an obstacle.

Finally, the optimisation of the riskiness index of delays in commencing visits is formulated as a generic global constraint compliant with the CP paradigm. Its example implementation for the CP solver from the or-tools framework [17] is available open source [21]. The constraint could provide the treatment of uncertainty in service time or travel time for any vehicle routing problem. The presence of the visits with pairwise synchronisation constraints is not required. The constraint could also be used outside the multistage optimisation framework with the stand-alone CP solver.

## 7.4 Future Research

The discussion of potential future research directions concludes the chapter.

Research on the formulation of the deterministic optimisation problem for the SatQKD considered in the thesis with one satellite and a fixed number of ground stations and no treatment of cloud cover uncertainty seems complete. The definition of the optimisation problem is established, and its computational complexity is rigorously proven. Finally, the formulation with time discretisation postulated in the thesis can be solved to provable optimality by a MIP solver in a short time. Further research on the deterministic SatQKD problem is possible by extending its scope. For instance, a communication system could include a constellation of multiple satellites. In a similar vein, spacecraft could execute manoeuvres that alter their orbits. Last but not least, spacecraft operations could be modelled more precisely, i.e., by tracking power spent on performing a given task compared to energy generated through solar panels.

The methodology proposed an uncertainty set that models temporal and geospatial correlations for the formulation of the SatQKD with the treatment of cloud cover uncertainty. As of this writing, there are no other readily available uncertainty sets or ambiguity sets proposed for modelling weather conditions. It is not immediately apparent how alternative uncertainty sets or ambiguity sets which provide such modelling capability could be developed. On the other hand, there seem to be multiple real-world applications that could benefit from including weather conditions into consideration, i.e., to schedule satellite Earth observations, manage disaster recovery operations, etc. Apart from the broad impact, data sets of weather conditions observed in the past and weather forecasts are easily obtainable and do not raise privacy-related concerns. The latter reduces the organisational and structural obstacles in the management and execution of such research projects.

An established way to report results obtained using optimisations models which differ in the treatment of uncertainty is to tabulate the value of the objective function for some suite of benchmark problems. This approach of presenting results facilitates ranking formulations to find the best one. However, the results obtained by the best stand-alone model for the suite of test problems could still be improved by taking multiple optimisation models into account. The ensemble model proposed in Chapter 4 combined two stand-alone formulations with the treatment of cloud cover uncertainty and selected which model to use based on the cloud cover conditions announced in the weather forecast. Such a solution approach outperformed each model individually on the suite of problem instances considered. The possibility to use a set of formulations with alternative treatments of uncertainty leads to many open research questions, such as the examples below. How to make decisions based on solutions obtained using models with different treatments of uncertainty? Are there models with the treatment of uncertainty that complement one another, i.e., one model works best in some circumstances but should be replaced by another model if these conditions are not met? Are there any systematic rules or guidelines suggesting which uncertainty treatment should be applied to make the most profitable decision? Providing comprehensive answers to the research questions proposed above could be the subject of future work.

Solution methods available in the literature as of this writing seem to be sufficient for the level of complexity instances of the VRPTWSync benchmark [144] require. All problem instances available in the benchmark are now solved. For the majority of them, optimal solutions were found. Subsequent improvements reported in the literature provide a relatively small reduction in the objective function, i.e., less than 1 %. On the other hand, the need for solving much larger problem instances arises in practical applications. Further research progress in solving the VRPTWSync could be accelerated by compiling a new benchmark which would contain problem instances with a much larger number of visits compared to the set of problem instances proposed by [144] more than ten years ago. Solving such a benchmark could help in identifying scalability limitations that may affect some solution methods. Resolving them should lead to more efficient computational approaches that can effectively solve problem instances considered large by current standards.

Optimisation models developed according to the RO, DRO and SO paradigms that are transformable to MIP programs can be solved using off-the-shelf solvers without the need to develop or extend the solution technology. The implementation cost is notably reduced in this manner. Nonetheless, some problem instances are beyond the scalability limitations of the MIP solvers and have to be solved using metaheuristics. Popular metaheuristics for vehicle routing problems and a vast array of local search operators are implemented in the open-source CP solver [79]. Unfortunately, CP is inherently developed for deterministic DO problems and lacks support for optimisation under uncertainty. That leads to a conundrum. A solution method developed from scratch is unlikely to match the performance and the reusability of a CP solver, which took the open-source community several years to develop. On the other hand, contributing to an open-source project requires overcoming the lack of documentation by analysing an extensive codebase developed by many other developers. This undesirable situation could be resolved by including support for optimisation under uncertainty to CP solvers. The author of the thesis made a step towards such an extension by developing a global constraint that computes an essential riskiness index of a delay over a set of scenarios.

To conclude, the section proposed multiple directions in which research topics considered in the thesis could be developed further. From the methodological standpoint, the development of best practices for using various optimisations models with the treatment of uncertainty simultaneously to make a decision for a given problem instance appears to be the most generic and broad considering scope. The extension of a CP solver to support uncertainty treatment over the set of scenarios requires expertise in software development and an in-depth study of the CP solver internals. The implementation of such an extension does not directly lead to contributions to science or methodology. Still, it could be an opportunity to develop and validate new algorithms, i.e., for the riskiness index optimisation. Finally, the formulation of a new set of VRPTWSync problem instances and the extension of an existing formulation for the deterministic SatQKD could be a viable short-term project for a student who makes the initial steps in academia.

# Appendix A

## A.1 Review of Norms

This section reviews the key concepts related to norms that are important from the perspective of RO. A more thorough introduction to norms and their use in Convex Optimization is offered by [72][Appendix 1].

A norm is the name of the class of functions $\mathbb{R}^n \rightarrow \mathbb{R}_{\geq 0}$ distinguished in the notation using $\|\cdot\|$ that satisfy the following properties:

**Triangle Inequality** $\|\mathbf{x}+\mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\| \quad \forall \mathbf{x}, \mathbf{y} \in \mathbb{R}^n$

**Positive Definiteness** $\|\mathbf{x}\| \geq 0 \quad \forall \mathbf{x} \in \mathbb{R}^n$ and $\|\mathbf{x}\| = 0 \iff \mathbf{x} = 0$

**Homogeneity** $\|a\mathbf{x}\| = |a|\|\mathbf{x}\| \quad \forall \mathbf{x} \in \mathbb{R}^n$ and $a \in \mathbb{R}$

A norm could be a measure of the distance between two vectors of the same length, i.e., $\Delta(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|$. For instance, given some reference vector $\mathbf{x}$, a norm could be used to define the set of all vectors which remain within some prescribed radius $r$ from the reference vector $\mathbf{x}$, i.e., $\{\mathbf{y} \in \mathbb{R}^n \mid \|\mathbf{x} - \mathbf{y}\| \leq r\}$. Such sets are known to be convex, closed, bounded and have a non-empty interior [72].

The names of popular norms, their symbols and definitions are listed in Table A.1.

### Dual Norm

Given an arbitrary norm $\|\cdot\|$, its dual can be defined as follows:

Table A.1: Popular Norms

| Name | Symbol | Formula |
|---|---|---|
| Manhattan Norm | $L_1$ | $\|\mathbf{x}\|_1 = \sum_{i=1}^{n} \|x_i\|$ |
| Euclidean Norm | $L_2$ | $\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2}$ |
| Chebyshev Norm | $L_\infty$ | $\|\mathbf{x}\|_\infty = \max_{i\in[1,\dots,n]} \|x_i\|$ |

$$\|\mathbf{x}\|_* = \max_{\|\mathbf{y}\|\leq 1} \mathbf{x}^\top \mathbf{y}$$

The Euclidean norm is self-dual, i.e., $\|\cdot\|_2 = \|\cdot\|_{2*}$. The norms $\|\cdot\|_1$ and $\|\cdot\|_\infty$ are dual to each other. The concept of dual norms is vital in deriving dual reformulations of conic programs which involve sets bounded by a norm, see, e.g., [202][Section 8.3].

## A.2 Reformulation of Worst-Case Expectation

This sketch of the proof demonstrates how to reformulate the worst-case expectation over a moment-based ambiguity set to a classical RC. The example illustrates essential reformulation techniques availed by moment-based ambiguity sets. The original proof was published in [71].

*Proof.* Let consider an ambiguity set $\mathcal{P}$ constructed over a convex, closed and bounded support set $\mathcal{W}$. Suppose the ambiguity set contains probability distributions with mean $\mu$ and variance restricted from above by the upper bound $\overline{\sigma}^2$. For the sake of simplicity, the family of probability distributions is univariate. The lifted counterparts of the support set and the ambiguity set are defined below.

$$\overline{\mathcal{W}} = \left\{(z, u) \in \mathbb{R} \times \mathbb{R} \,\middle|\, z \in \mathcal{W}, (z - \mu)^2 \leq u\right\}$$

$$\overline{\mathcal{P}} = \left\{ \overline{\mathbb{P}} \in \mathcal{P}(\mathbb{R} \times \mathbb{R}) \;\middle|\; \begin{array}{l} \overline{\mathbb{P}}((z,u) \in \overline{\mathcal{W}}) = 1 \\ \mathbb{E}_{\overline{\mathbb{P}}}(z) = \mu \\ \mathbb{E}_{\overline{\mathbb{P}}}(u) \leq \overline{\sigma}^2 \end{array} \right\}$$

Let derive the reformulation for the problem of estimating the worst-case expectation of some function $f(\cdot)$ over the ambiguity set $\overline{\mathcal{P}}$.

$$\max_{\overline{\mathbb{P}} \in \overline{\mathcal{P}}} \mathbb{E}_{\overline{\mathbb{P}}}(f(\tilde{z}))$$

For ease of exposition, the function $f(\cdot)$ takes only a random variable as the input and does not depend on decision variables. The evaluation of the worst-case expectation is an example of a constrained optimisation problem because the structure of the ambiguity set and the definition of a probability distribution function enforce the following constraints.

$$\iint_{\overline{\mathcal{W}}} \overline{\mathbb{P}}(z,u) du dz = 1 \tag{A.1}$$

$$\iint_{\overline{\mathcal{W}}} z\overline{\mathbb{P}}(z,u) du dz = \mu \tag{A.2}$$

$$\iint_{\overline{\mathcal{W}}} u\overline{\mathbb{P}}(z,u) du dz \leq \sigma^2 \tag{A.3}$$

$$\overline{\mathbb{P}}(z,u) \geq 0 \quad \forall (z,u) \in \overline{\mathcal{W}} \tag{A.4}$$

The worst-case expectation could be bounded from above by constructing a Lagrangian. Let $p, q \in \mathbb{R}$ and $r \in \mathbb{R}_{\geq 0}$ be the dual multipliers associated with Constraints (A.1), (A.2) and (A.3), respectively.

$$\begin{aligned}L(z,u,p,q,r) =& \iint\limits_{\overline{\mathcal{W}}} f(z)\overline{\mathbb{P}}(z,u)dudz - p\left(\iint\limits_{\overline{\mathcal{W}}} \overline{\mathbb{P}}(z,u)dudz - 1\right)\\ &-q\left(\iint\limits_{\overline{\mathcal{W}}} z\overline{\mathbb{P}}(z,u)dudz - \mu\right) - r\left(\iint\limits_{\overline{\mathcal{W}}} u\overline{\mathbb{P}}(z,u)dudz - \sigma^2\right)\\ &= p + q\mu + r\sigma^2 + \iint\limits_{\overline{\mathcal{W}}} \left(f(z) - (p + qz + ru)\right)\overline{\mathbb{P}}(z,u)dudz\end{aligned}$$

Let consider the maximisation of the Lagrangian subject to the probability distribution $\overline{\mathbb{P}}$.

$$p + q\mu + r\sigma^2 + \max_{\overline{\mathbb{P}}\in\overline{\mathcal{P}}} \iint\limits_{\overline{\mathcal{W}}} \left(f(z) - (p + qz + ru)\right)\overline{\mathbb{P}}(z,u)dudz$$

If values $z$ and $u$ that satisfy $f(z) \geq p + qz + ru$ can be found, then it is possible to select dual coefficients and the probability distribution for which the integral is unbounded. It would make the upper bound meaningless. As a result, hereafter, let introduce the implicit constraint $f(z) \leq p + qz + ru \quad \forall z, u \in \overline{\mathcal{W}}$. The new constraint enforces the integral to be non-positive. Consequently, $p + q\mu + r\sigma^2$ is a valid upper bound on the worst-case expectation over the ambiguity set.

Taken together, the upper bound on the worst-case expectation can be computed by solving the following optimisation problem.

$$\begin{aligned} \min p + q\mu + r\sigma^2 & \\ p + qz + ru \geq f(z) & \qquad\qquad \forall z, u \in \overline{\mathcal{W}} \\ p, q \in \mathbb{R}, r \in \mathbb{R}_{\geq 0} & \end{aligned}$$

□

Overall, the optimisation problem has the same structure as the classical RC in which the support set $\overline{\mathcal{W}}$ acts as the uncertainty set. The computational difficulty of solving the problem depends on the structure of the support set $\overline{\mathcal{W}}$ and the function $f(\cdot)$. For instance, the optimisation problem is tractable if the support set is convex, closed and bounded, and the function is piecewise linear.

## A.3 Order Selection for the Vector Autoregressive Model

The order selection process for the VAR model was conducted using the BIC score. This and alternative decision criteria were described in detail in the book [18][Chapter 4].

The BIC score [203] is computed given the number of parameters to fit in the model ($k$), the number of training samples ($n$), and the maximum likelihood estimation ($L_{max}$).

$$\mathrm{BIC} = k \ln(n) - 2 \ln(L_{\mathrm{max}})$$

The VAR model is trained using a data set of historical weather observations. Weather conditions were recorded every 3 hours in selected cities in the UK (i.e., Birmingham, Bristol, Glasgow, London, Manchester) throughout the entire 2018.

Figure A.1 illustrates the BIC scores for VAR models of different orders.

Overall, the order selection procedure suggests setting the lag to one. The result is aligned with previous studies described in the literature related to the modelling of meteorological phenomena such as temperature, precipitation [117] and solar irradiance [107].

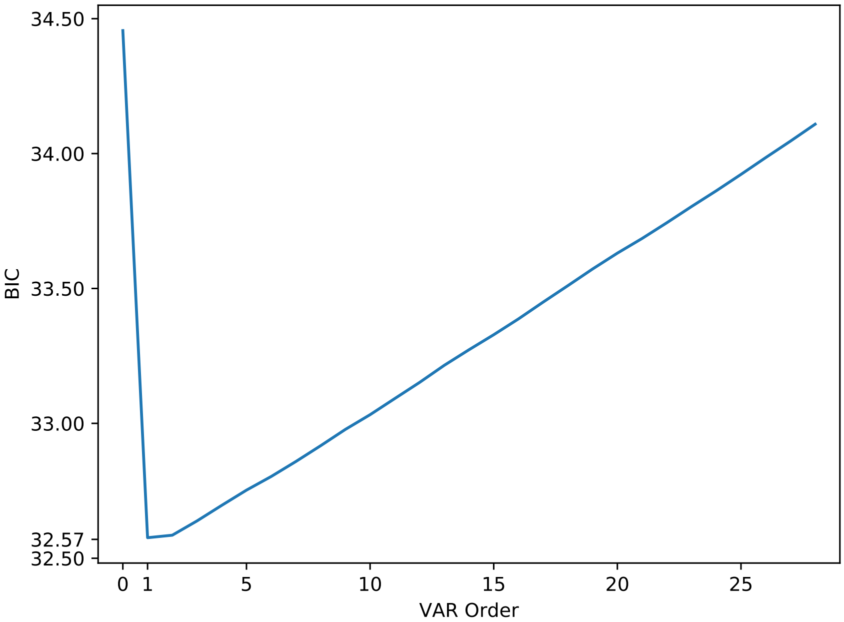


Figure A.1: BIC score vs order of the VAR model trained using historical weather observations recorded in selected cities in the UK in 2018.

# Bibliography

[1] I. Stewart, *Do Dice Play God? The Mathematics of Uncertainty.* Profile Books LTD, 2019, ch. Six Ages of Uncerainty, pp. 1–13.

[2] G. B. Dantzig, *Linear Programming and Extensions.* Princeton University Press, 2021/04/06/ 1991. [Online]. Available: https://doi.org/10.2307/j.ctt1cx3tvg

[3] D. Bertsimas and J. Tsitsiklis, *Introduction to Linear Optimization*, 1st ed. Athena Scientific, 1997, ch. Introduction, pp. 1–40.

[4] ——, *Introduction to Linear Optimization*, 1st ed. Athena Scientific, 1997, ch. The simplex method, pp. 81–139.

[5] ——, *Introduction to Linear Optimization*, 1st ed. Athena Scientific, 1997, ch. The art in linear optimisation, pp. 533–569.

[6] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski, *Robust Optimization*, ser. Princeton Series in Applied Mathematics. Princeton University Press, October 2009.

[7] A. Ben-Tal and A. Nemirovski, "Robust solutions of linear programming problems contaminated with uncertain data," *Mathematical Programming*, vol. 88, no. 3, pp. 411–424, 2000.

[8] D. Bertsimas and M. Sim, "The price of robustness," *Operations Research*, vol. 52, no. 1, pp. 35–53, 2004. [Online]. Available: https://doi.org/10.1287/opre.1030.0065

[9] A. Ben-Tal and A. Nemirovski, “Robust convex optimization,” *Mathematics of Operations Research*, vol. 23, no. 4, pp. 769–805, 1998. [Online]. Available: https://doi.org/10.1287/moor.23.4.769

[10] ——, “Robust solutions of uncertain linear programs,” *Operations Research Letters*, vol. 25, no. 1, pp. 1 – 13, 1999. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167637799000164

[11] A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski, “Adjustable robust solutions of uncertain linear programs,” *Mathematical Programming*, vol. 99, no. 2, pp. 351–376, 2004. [Online]. Available: https://doi.org/10.1007/s10107-003-0454-y

[12] D. Bertsimas and M. Sim, “Robust discrete optimization and network flows,” *Mathematical Programming*, vol. 98, no. 1, pp. 49–71, 2003. [Online]. Available: https://doi.org/10.1007/s10107-003-0396-4

[13] E. Delage and Y. Ye, “Distributionally robust optimization under moment uncertainty with application to data-driven problems,” *Operations Research*, vol. 58, no. 3, pp. 595–612, 2010. [Online]. Available: https://doi.org/10.1287/opre.1090.0741

[14] W. Wiesemann, D. Kuhn, and M. Sim, “Distributionally robust convex optimization,” *Operations Research*, vol. 62, no. 6, pp. 1358–1376, 2014. [Online]. Available: https://doi.org/10.1287/opre.2014.1314

[15] A. Shapiro, D. Dentcheva, and A. Ruszczyński, *Lectures on Stochastic Programming.* Society for Industrial and Applied Mathematics, 2009.

[16] Z. Yu, Z. Zhang, A. Lim, and M. Sim, “Robust Data-Driven Vehicle Routing with Time Windows,” *Available at Optimisation Online*, 2018, accessed 15/01/2020.

[17] Google, “Google Optimization Tools,” https://developers.google.com/optimization/, 2018, accessed: 2018-06-19.

[18] H. Lütkepohl, *New Introduction to Multiple Time Series Analysis.* Springer, 2005, ch. VAR Order Selection and Checking the Model Adequacy, pp. 135 – 192.

[19] D. Bredström and M. Rönnqvist, "Combined vehicle routing and scheduling with temporal precedence and synchronization constraints," *European Journal of Operational Research*, vol. 191, no. 1, pp. 19–31, 2008.

[20] M. Polnik, L. Mazzarella, M. Di Carlo, D. Oi, A. Riccardi, and A. Arulselvan, "Scheduling of space to ground quantum key distribution," *EPJ Quantum Technology*, vol. 7, 1 2020.

[21] M. Polnik and M. Di Carlo, "Source code repository for: Scheduling of space to ground quantum key distribution," https://github.com/pmateusz/uncertain-quake, Aug. 2020.

[22] M. Polnik, L. Mazzarella, M. Di Carlo, D. Oi, A. Riccardi, and A. Arulselvan, "Data for: "scheduling of space to ground quantum key distribution"," Jan. 2020.

[23] M. Polnik, A. Riccardi, and K. Akartunali, "A multistage optimisation algorithm for the large vehicle routing problem with time windows and synchronised visits," *Journal of Operational Research Society*, 8 2020.

[24] M. Polnik and A. Riccardi, "Source code repository for: A multistage optimisation algorithm for the large vehicle routing problem with time windows and synchronised visits," https://github.com/pmateusz/cordia, Aug. 2020.

[25] M. Polnik, A. Riccardi, and K. Akartunali, "Benchmark solutions of the vehicle routing problem with synchronised visits," Jul. 2020.

[26] ——, "Dataset of home care scheduling and routing problems with synchronized visits," Oct. 2018.

[27] M. Polnik and A. Riccardi, "Indexing discrete sets in a label setting algorithm for solving the elementary shortest path problem with resource constraints,"

in *2018 IEEE Congress on Evolutionary Computation*, 7 2018, 2018 IEEE Congress on Evolutionary Computation ; Conference date: 08-07-2018 Through 13-07-2018. [Online]. Available: http://www.ecomp.poli.br/~wcci2018/

[28] B. Verweij, S. Ahmed, A. J. Kleywegt, G. L. Nemhauser, and A. Shapiro, "The Sample Average Approximation Method Applied to Stochastic Routing Problems: A Computational Study," *Computational Optimization and Applications*, vol. 24, pp. 289–333, 2003.

[29] A. J. Kleywegt, A. Shapiro, and T. Homem-de Mello, "The Sample Average Approximation Method for Stochastic Discrete Optimization," *SIAM J. on Optimization*, vol. 12, no. 2, p. 479–502, Feb. 2002. [Online]. Available: https://doi.org/10.1137/S1052623499363220

[30] F. H. Knight, *Risk, uncertainty and profit.* Courier Corporation, 1921.

[31] D. Dentcheva, *Lectures on Stochastic Programming.* Society for Industrial and Applied Mathematics, 2009, ch. Optimization Models with Probabilistic Constraints, pp. 87–153.

[32] C. Buchheim and J. Kurtz, "Robust combinatorial optimization under convex and discrete cost uncertainty," *EURO Journal on Computational Optimization*, vol. 6, no. 3, pp. 211–238, 2018. [Online]. Available: https://doi.org/10.1007/s13675-018-0103-0

[33] R. Rockafellar and S. Uryasev, "Conditional value-at-risk for general loss distributions," *Journal of Banking & Finance*, vol. 26, no. 7, pp. 1443 – 1471, 2002. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378426602002716

[34] S. Sarykalin, G. Serraino, and S. Uryasev, *Value-at-Risk vs. Conditional Value-at-Risk in Risk Management and Optimization.* The Institute for Operations Research and the Management Sciences, 2014, ch. Chapter 13, pp. 270–294. [Online]. Available: https://pubsonline.informs.org/doi/abs/10.1287/educ.1080.0052

[35] J. von Neumann, O. Morgenstern, and A. Rubinstein, *Theory of Games and Economic Behavior*. Princeton University Press, 1944. [Online]. Available: http://www.jstor.org/stable/j.ctt1r2gkx

[36] A. Ben-Tal and M. Teboulle, "An old-new concept of convex risk measures: The optimized certainty equivalent," *Mathematical Finance*, vol. 17, no. 3, pp. 449–476, 2007.

[37] L. Eeckhoudt, C. Gollier, and H. Schlesinger, *Economic and Financial Decisions under Risk*. Princeton University Press, 2005, ch. Risk Aversion.

[38] J. W. Pratt, "Risk aversion in the small and in the large," *Econometrica*, vol. 32, no. 1/2, pp. 122–136, 1964. [Online]. Available: http://www.jstor.org/stable/1913738

[39] H. Bühlmann, *Mathematical Methods in Risk Theory*. Springer-Verlag Berlin Heidelberg, 1970.

[40] R. J. Aumann and R. Serrano, "An Economic Index of Riskiness," *Journal of Political Economy*, vol. 116, no. 5, pp. 810–836, 2008.

[41] K. Schulze, "Existence and computation of the Aumann-Serrano index of riskiness and its extension," *Journal of Mathematical Economics*, vol. 50, pp. 219 – 224, 2014.

[42] P. Jaillet, J. Qi, and M. Sim, "Routing Optimization Under Uncertainty," *Operations Research*, vol. 64, no. 1, pp. 186–200, 2016.

[43] Y. Zhang, R. Baldacci, M. Sim, and J. Tang, "Routing optimization with time windows under uncertainty," *Mathematical Programming*, vol. 175, no. 1, pp. 263–305, May 2019.

[44] T. Homem-de Mello and G. Bayraksan, "Monte Carlo sampling-based methods for stochastic optimization," *Surveys in Operations Research and Management Science*, vol. 19, no. 1, pp. 56–85, Jan 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1876735414000038

[45] G. A. Hanasusanto, D. Kuhn, and W. Wiesemann, “A comment on computational complexity of stochastic programming problems,” *Mathematical Programming*, vol. 159, no. 1, pp. 557–569, 2016.

[46] G. Bayraksan and D. P. Morton, *Assessing Solution Quality in Stochastic Programs via Sampling*, ser. INFORMS TutORials in Operations Research. INFORMS, Sep 2009, pp. 102–122, 0. [Online]. Available: https://doi.org/10.1287/educ.1090.0065

[47] M. B. Freimer, J. T. Linderoth, and D. J. Thomas, “The impact of sampling methods on bias and variance in stochastic linear programs,” *Computational Optimization and Applications*, vol. 51, no. 1, pp. 51–75, Jan 2012. [Online]. Available: https://doi.org/10.1007/s10589-010-9322-x

[48] W.-K. Mak, D. P. Morton, and R. Wood, “Monte carlo bounding techniques for determining solution quality in stochastic programs,” *Operations Research Letters*, vol. 24, no. 1, pp. 47–56, Feb 1999. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167637798000546

[49] A. Shapiro, T. Homem-de Mello, and J. Kim, “Conditioning of convex piecewise linear stochastic programs,” *Mathematical Programming*, vol. 94, no. 1, pp. 1–19, Dec 2002. [Online]. Available: https://doi.org/10.1007/s10107-002-0313-2

[50] G. Bayraksan and D. P. Morton, “A sequential sampling procedure for stochastic programming,” *Operations Research*, vol. 59, no. 4, pp. 898–913, 2021/05/29/ 2011, full publication date: July • August • 2011. [Online]. Available: http://www.jstor.org/stable/23013154

[51] A. Nemirovski and A. Shapiro, “Convex approximations of chance constrained programs,” *SIAM Journal on Optimization*, vol. 17, no. 4, pp. 969–996, 2007. [Online]. Available: https://doi.org/10.1137/050622328

[52] R. Rockafellar and S. Uryasev, “Optimization of Conditional Value-At-Risk,” *Journal of risk*, vol. 2, pp. 21–42, 01 2000.

[53] G. C. Calafiore and M. C. Campi, "The scenario approach to robust control design," *IEEE Transactions on Automatic Control*, vol. 51, no. 5, pp. 742–753, May 2006.

[54] W. Chen, M. Sim, J. Sun, and C.-P. Teo, "From CVaR to Uncertainty Set: Implications in Joint Chance-Constrained Optimization," *Operations Research*, vol. 58, no. 2, pp. 470–485, 2010. [Online]. Available: https://doi.org/10.1287/opre.1090.0712

[55] X. Chen, M. Sim, and P. Sun, "A robust optimization perspective on stochastic programming," *Operations Research*, vol. 55, no. 6, pp. 1058–1071, 2007. [Online]. Available: https://doi.org/10.1287/opre.1070.0441

[56] P. Mohajerin Esfahani and D. Kuhn, "Data-driven distributionally robust optimization using the Wasserstein metric: performance guarantees and tractable reformulations," *Mathematical Programming*, vol. 171, no. 1, pp. 115–166, 2018. [Online]. Available: https://doi.org/10.1007/s10107-017-1172-1

[57] B. L. Gorissen, İhsan Yanıkoğlu, and D. den Hertog, "A practical guide to robust optimization," *Omega*, vol. 53, pp. 124 – 137, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0305048314001698

[58] D. Bertsimas, D. B. Brown, and C. Caramanis, "Theory and applications of robust optimization," *SIAM Review*, vol. 53, no. 3, pp. 464–501, 2011. [Online]. Available: https://doi.org/10.1137/080734510

[59] A. L. Soyster, "Convex programming with set-inclusive constraints and applications to inexact linear programming," *Operations Research*, vol. 21, no. 5, pp. 1154–1157, 1973. [Online]. Available: http://www.jstor.org/stable/168933

[60] D. Bertsimas, D. Pachamanova, and M. Sim, "Robust linear optimization under general norms," *Operations Research Letters*, vol. 32, no. 6, pp. 510 – 516, 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167637704000082

[61] D. Bertsimas and D. B. Brown, "Constructing uncertainty sets for robust linear optimization," *Operations Research*, vol. 57, no. 6, pp. 1483–1495, 2009. [Online]. Available: https://doi.org/10.1287/opre.1080.0646

[62] G. Y. Kouvelis, Panos, *Robust Discrete Optimization and Its Applications*. Springer US, 1997.

[63] D. Bertsimas, D. A. Iancu, and P. A. Parrilo, "Optimality of affine policies in multistage robust optimization," *Mathematics of Operations Research*, vol. 35, no. 2, pp. 363–394, 2010. [Online]. Available: https://doi.org/10.1287/moor.1100.0444

[64] D. Bertsimas, M. Sim, and M. Zhang, "A practically efficient approach for solving adaptive distributionally robust linear optimization problems," *Management Science*, vol. 65, 05 2018.

[65] X. Chen and Y. Zhang, "Uncertain linear programs: Extended affinely adjustable robust counterparts," *Operations Research*, vol. 57, no. 6, pp. 1469–1482, 2009. [Online]. Available: https://doi.org/10.1287/opre.1080.0605

[66] X. Chen, M. Sim, P. Sun, and J. Zhang, "A linear decision-based approximation approach to stochastic programming," *Operations Research*, vol. 56, no. 2, pp. 344–357, 2008. [Online]. Available: https://doi.org/10.1287/opre.1070.0457

[67] G. A. Hanasusanto, D. Kuhn, S. W. Wallace, and S. Zymler, "Distributionally robust multi-item newsvendor problems with multimodal demand distributions," *Mathematical Programming*, vol. 152, no. 1, pp. 1–32, 2015. [Online]. Available: https://doi.org/10.1007/s10107-014-0776-y

[68] D. Bertsimas and C. Caramanis, "Finite adaptability in multistage linear optimization," *IEEE Transactions on Automatic Control*, vol. 55, pp. 2751–2766, 2010. [Online]. Available: https://doi.org/10.1109/TAC.2010.2049764

[69] D. Bertsimas, V. Gupta, and N. Kallus, "Data-driven robust optimization," *Mathematical Programming*, vol. 167, no. 2, pp. 235–292, 2018. [Online]. Available: https://doi.org/10.1007/s10107-017-1125-8

[70] Z. Hu and J. Hong, "Kullback-Leibler Divergence Constrained Distributionally Robust Optimization," *Available at Optimization Online*, 2012, accessed 25/03/2020.

[71] K. Isii, "On sharpness of tchebycheff-type inequalities," *Annals of the Institute of Statistical Mathematics*, vol. 14, no. 1, pp. 185–197, 1962. [Online]. Available: https://doi.org/10.1007/BF02868641

[72] S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge University Press, 2004.

[73] A. Mutapcic and S. Boyd, "Cutting-set methods for robust convex optimization with pessimizing oracles," *Optimization Methods Software*, vol. 24, no. 3, p. 381–406, Jun. 2009. [Online]. Available: https://doi.org/10.1080/10556780802712889

[74] J. E. Kelley, "The cutting-plane method for solving convex programs," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 4, pp. 703–712, 2020/03/20/ 1960. [Online]. Available: www.jstor.org/stable/2099058

[75] D. Bertsimas, I. Dunning, and M. Lubin, "Reformulation versus cutting-planes for robust optimization," *Computational Management Science*, vol. 13, no. 2, pp. 195–217, 2016. [Online]. Available: https://doi.org/10.1007/s10287-015-0236-z

[76] E. Nikolova, "Approximation algorithms for reliable stochastic combinatorial optimization," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, M. Serna, R. Shaltiel, K. Jansen, and J. Rolim, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 338–351.

[77] J.-X. R. Nicolas Beldiceanu, Mats Carlsson, “Global constraint catalog,” 2014. [Online]. Available: https://web.imt-atlantique.fr/x-info/sdemasse/aux/doc/catalog.pdf

[78] “IBM ILOG CPLEX Optimization Studio CP Optimizer User’s Manual,” accessed 07/04/2020. [Online]. Available: https://www.ibm.com/support/knowledgecenter/SSSA5P_12.10.0/ilog.odms.cpo.help/CP_Optimizer/User_manual/topics/usrcpoptimizer.html

[79] “OR Tools Guides - Constraint Optimization,” accessed 07/04/2020. [Online]. Available: https://developers.google.com/optimization/cp

[80] “The MiniZinc Challenge,” accessed 07/04/2020. [Online]. Available: https://www.minizinc.org/challenge.html

[81] P. Shaw, “Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems,” in *Principles and Practice of Constraint Programming — CP98*, M. Maher and J.-F. Puget, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 417–431.

[82] G. Laporte, S. Ropke, and T. Vidal, *Chapter 4: Heuristics for the Vehicle Routing Problem.* Society for Industrial and Applied Mathematics, 2014, ch. 4, pp. 87–116. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/1.9781611973594.ch4

[83] F. Glover, “Tabu Search – Part I,” *ORSA Journal on Computing*, vol. 1, no. 3, pp. 190–206, 1989. [Online]. Available: https://doi.org/10.1287/ijoc.1.3.190

[84] K. Sörensen, “Finding robust solutions using local search,” *Journal of Mathematical Modelling and Algorithms*, vol. 3, no. 1, pp. 89–103, 2004. [Online]. Available: https://doi.org/10.1023/B:JMMA.0000026710.74315.3e

[85] E. Solano-Charris, C. Prins, and A. C. Santos, “Local search based metaheuristics for the robust vehicle routing problem with discrete scenarios,”

*Applied Soft Computing*, vol. 32, pp. 518 – 531, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1568494615002380

[86] W. Ogryczak and T. Śliwiński, "On direct methods for lexicographic min-max optimization," in *Computational Science and Its Applications - ICCSA 2006*, M. Gavrilova, O. Gervasi, V. Kumar, C. J. K. Tan, D. Taniar, A. Laganá, Y. Mun, and H. Choo, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 802–811.

[87] P. Perny and O. Spanjaard, "An axiomatic approach to robustness in search problems with multiple scenarios," in *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence*, ser. UAI'03. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2002, p. 469–476.

[88] J. Wang, E. Demeulemeester, and D. Qiu, "A pure proactive scheduling algorithm for multiple Earth observation satellites under uncertainties of clouds," *Computers & Operations Research*, vol. 74, pp. 1 – 13, 2016.

[89] X. Wang, G. Song, R. Leus, and C. Han, "Robust Earth observation satellite scheduling with uncertainty of cloud coverage," *IEEE Transactions on Aerospace and Electronic Systems*, pp. 1–1, 2019.

[90] S.-K. Liao, W.-Q. Cai, W.-Y. Liu, L. Zhang, Y. Li, J.-G. Ren, J. Yin, Q. Shen, Y. Cao, Z.-P. Li, F.-Z. Li, X.-W. Chen, L.-H. Sun, J.-J. Jia, J.-C. Wu, X.-J. Jiang, J.-F. Wang, Y.-M. Huang, Q. Wang, Y.-L. Zhou, L. Deng, T. Xi, L. Ma, T. Hu, Q. Zhang, Y.-A. Chen, N.-L. Liu, X.-B. Wang, Z.-C. Zhu, C.-Y. Lu, R. Shu, C.-Z. Peng, J.-Y. Wang, and J.-W. Pan, "Satellite-to-ground quantum key distribution," *Nature*, vol. 549, p. 43, Aug 2017.

[91] G. S. Mecherle, "Mitigation of atmospheric effects on terrestrial free-space optical communication systems," vol. 5338, International Society for Optics and Photonics. San Jose, United States: SPIE, 2004, pp. 102 – 118.

[92] M. Capelle, M.-J. Huguet, N. Jozefowiez, and X. Olive, "Optimizing ground station networks for free space optical communications: Maximizing the data transfer," *Networks*, vol. 73, no. 2, pp. 234–253, 2019.

[93] H. Kaushal and G. Kaddoum, "Optical communication in space: Challenges and mitigation techniques," *IEEE Communications Surveys Tutorials*, vol. 19, no. 1, pp. 57–96, Firstquarter 2017.

[94] C. Bennett and G. Brassard, "Withdrawn: Quantum cryptography: Public key distribution and coin tossing," in *Proceedings of the International Conference on Computers, Systems & Signal Processing*, vol. 560, 01 1984, pp. 175–179.

[95] M. Uysal, C. Capsoni, Z. Ghassemlooy, A. Boucouvalas, and E. Udvary, *Optical Wireless Communications – An Emerging Technology.* Springer International Publishing, 01 2017.

[96] S. Spangelo, J. Cutler, K. Gilson, and A. Cohn, "Optimization-based scheduling for the single-satellite, multi-ground station communication problem," *Computers and Operations Research*, vol. 57, pp. 1 – 16, 2015.

[97] J. Castaing, "Scheduling downloads for multi-satellite, multi-ground station missions," in *Proceedings of the Small Satellite Conference, Technical Session VIII, SSC14-VIII-4*, Logan, United States, 2014. [Online]. Available: https://digitalcommons.usu.edu/smallsat/2014/FJRStudentComp/4/

[98] Z. Zhang, Q. Zhao, M. Razavi, and X. Ma, "Improved key-rate bounds for practical decoy-state quantum-key-distribution systems," *Physical Review A*, vol. 95, p. 012333, Jan 2017.

[99] E. Diamanti, H.-K. Lo, B. Qi, and Z. Yuan, "Practical challenges in quantum key distribution," *npj Quantum Information*, vol. 2, no. 1, p. 16025, 2016.

[100] C. C. W. Lim, M. Curty, N. Walenta, F. Xu, and H. Zbinden, "Concise security bounds for practical decoy-state quantum key distribution," *Physical Review A*, vol. 89, p. 022307, Feb 2014.

[101] J. Yin, Y. Cao, Y.-H. Li, S.-K. Liao, L. Zhang, J.-G. Ren, W.-Q. Cai, W.-Y. Liu, B. Li, H. Dai, G.-B. Li, Q.-M. Lu, Y.-H. Gong, Y. Xu, S.-L. Li, F.-Z. Li, Y.-Y. Yin, Z.-Q. Jiang, M. Li, J.-J. Jia, G. Ren, D. He, Y.-L. Zhou, X.-X. Zhang, N. Wang, X. Chang, Z.-C. Zhu, N.-L. Liu, Y.-A. Chen, C.-Y. Lu, R. Shu, C.-Z. Peng, J.-Y. Wang, and J.-W. Pan, "Satellite-based entanglement distribution over 1200 kilometers," *Science*, vol. 356, no. 6343, pp. 1140–1144, 2017.

[102] J.-G. Ren, P. Xu, H.-L. Yong, L. Zhang, S.-K. Liao, J. Yin, W.-Y. Liu, W.-Q. Cai, M. Yang, L. Li, K.-X. Yang, X. Han, Y.-Q. Yao, J. Li, H.-Y. Wu, S. Wan, L. Liu, D.-Q. Liu, Y.-W. Kuang, Z.-P. He, P. Shang, C. Guo, R.-H. Zheng, K. Tian, Z.-C. Zhu, N.-L. Liu, C.-Y. Lu, R. Shu, Y.-A. Chen, C.-Z. Peng, J.-Y. Wang, and J.-W. Pan, "Ground-to-satellite quantum teleportation," *Nature*, vol. 549, p. 70, Aug 2017.

[103] D. K. L. Oi, A. Ling, G. Vallone, P. Villoresi, S. Greenland, E. Kerr, M. Macdonald, H. Weinfurter, H. Kuiper, E. Charbon, and R. Ursin, "Cubesat quantum communications mission," *EPJ Quantum Technology*, vol. 4, no. 1, p. 6, Apr 2017.

[104] S. P. Neumann, S. K. Joshi, M. Fink, T. Scheidl, R. Blach, C. Scharlemann, S. Abouagaga, D. Bambery, E. Kerstel, M. Barthelemy, and R. Ursin, "Q3Sat: quantum communications uplink to a 3U CubeSat—feasibility & design," *EPJ Quantum Technology*, vol. 5, no. 1, p. 4, Apr 2018.

[105] T. Vergoossen, S. Loarte, R. Bedington, H. Kuiper, and A. Ling, "Satellite constellations for trusted node QKD networks," 2019.

[106] D. Liao and Y. Yang, "Imaging Order Scheduling of an Earth Observation Satellite," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 5, pp. 794–802, Sep. 2007.

[107] H. Morf, "Sunshine and cloud cover prediction based on Markov processes," *Solar Energy*, vol. 110, pp. 615 – 626, 2014.

[108] B. L. Deuermeyer, D. K. Friesen, and M. A. Langston, "Scheduling to maximize the minimum processor finish time in a multiprocessor system," *SIAM Journal*

*on Algebraic Discrete Methods*, vol. 3, no. 2, pp. 190–196, 1982. [Online]. Available: https://doi.org/10.1137/0603019

[109] E. L. Schreiber, R. E. Korf, and M. D. Moffitt, "Optimal multi-way number partitioning," *J. ACM*, vol. 65, no. 4, Jul. 2018. [Online]. Available: https://doi.org/10.1145/3184400

[110] T. Walsh, "Where are the really hard manipulation problems? the phase transition in manipulating the veto rule," in *Proceedings of the 21st International Jont Conference on Artifical Intelligence*, ser. IJCAI'09. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2009, p. 324–329.

[111] D. K. Friesen and B. L. Deuermeyer, "Analysis of greedy solutions for a replacement part sequencing problem," *Mathematics of Operations Research*, vol. 6, no. 1, pp. 74–87, 1981. [Online]. Available: https://doi.org/10.1287/moor.6.1.74

[112] M. Polnik, A. Arulselvan, and A. Riccardi, "Scheduling space-to-ground optical communication under cloud cover uncertainty," *IEEE Transactions on Aerospace and Electronic Systems*, March 2021. [Online]. Available: https://doi.org/10.1109/TAES.2021.3069286

[113] Open Weather Map. (2019) Service homepage. Last accessed 25/1/2019. [Online]. Available: https://openweathermap.org

[114] E. Zivot and J. Wang, *Vector Autoregressive Models for Multivariate Time Series*. New York, NY: Springer New York, 2006, ch. 11, pp. 385–429.

[115] W. McKinney, J. Perktold, and S. Seabold, "Time series analysis in python with statsmodels," in *Proceedings of the 10th Python in Science Conference (SciPy 2011)*, 2011, pp. 96–102. [Online]. Available: http://conference.scipy.org/proceedings/scipy2011/pdfs/statsmodels.pdf

[116] H. Bozdogan, “Model Selection and Akaike’s Information Criterion (AIC): The General Theory and Its Analytical Extensions,” *Psychometrika*, vol. 52, pp. 345–370, 02 1987.

[117] N. J. Sparks, S. R. Hardwick, M. Schmid, and R. Toumi, “IMAGE: a multivariate multi-site stochastic weather generator for European weather and climate,” *Stochastic Environmental Research and Risk Assessment*, vol. 32, no. 3, pp. 771–784, Mar 2018.

[118] D. Bertsimas, M. Sim, and M. Zhang, “Adaptive distributionally robust optimization,” *Management Science*, vol. 65, no. 2, pp. 604–618, 2019.

[119] A. Shapiro and A. Nemirovski, *On Complexity of Stochastic Programming Problems.* Boston, MA: Springer US, 2005, pp. 111–146. [Online]. Available: https://doi.org/10.1007/0-387-26771-9_4

[120] E. Herz and J. Campagna, *BridgeSat Laser Communication Scheduling: a Case Study*, ser. SpaceOps Conferences. American Institute of Aeronautics and Astronautics, May 2018, 0. [Online]. Available: https://doi.org/10.2514/6.2018-2353

[121] F. Marinelli, S. Nocella, F. Rossi, and S. Smriglio, “A Lagrangian heuristic for satellite range scheduling with resource constraints,” *Computers and Operations Research*, vol. 38, no. 11, pp. 1572 – 1583, 2011.

[122] B. Chen, C. N. Potts, and G. J. Woeginger, *Handbook of Combinatorial Optimization: Volume1–3.* Boston, MA: Springer US, 1998, ch. A Review of Machine Scheduling: Complexity, Algorithms and Approximability, pp. 1493–1641.

[123] M. Drozdowski, “Scheduling multiprocessor tasks — an overview,” *European Journal of Operational Research*, vol. 94, no. 2, pp. 215 – 230, 1996.

[124] J. C. Pemberton and F. Galiber, III, “A constraint-based approach to satellite scheduling,” in *DIMACS Workshop on on Constraint Programming and Large*

*Scale Discrete Optimization.* Boston, MA, USA: American Mathematical Society, 2001, pp. 101–114.

[125] W. Herroelen, B. D. Reyck, and E. Demeulemeester, "Resource-constrained project scheduling: A survey of recent developments," *Computers and Operations Research*, vol. 25, no. 4, pp. 279 – 302, 1998.

[126] D. Karapetyan, S. M. Minic, K. T. Malladi, and A. P. Punnen, "Satellite downlink scheduling problem: A case study," *Omega*, vol. 53, pp. 115 – 123, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0305048315000031

[127] F. Xhafa, J. Sun, A. Barolli, A. Biberaj, and L. Barolli, "Genetic algorithms for satellite scheduling problems," *Mobile Information Systems*, vol. 8, no. 4, pp. 351–377, Oct. 2012.

[128] C. Han, X. Wang, G. Song, and R. Leus, "Scheduling multiple agile Earth observation satellites with multiple observations," 2018.

[129] L. Barbulescu, J.-P. Watson, L. D. Whitley, and A. E. Howe, "Scheduling Space–Ground Communications for the Air Force Satellite Control Network," *Journal of Scheduling*, vol. 7, no. 1, pp. 7–34, Jan 2004.

[130] L. Barbulescu, A. Howe, and D. Whitley, "AFSCN scheduling: How the problem and solution have evolved," *Mathematical and Computer Modelling*, vol. 43, no. 9, pp. 1023 – 1037, 2006, optimization and Control for Military Applications.

[131] R. Stottler and R. Richards, "Managed intelligent deconfliction and scheduling for satellite communication," in *2018 IEEE Aerospace Conference*, March 2018, pp. 1–7.

[132] G. Arkali, M. Dawande, and C. Sriskandarajah, "Scheduling support times for satellites with overlapping visibilities," *Production and Operations Management*, vol. 17, no. 2, pp. 224–234, 2008.

[133] B. M. Khumawala, “An efficient branch and bound algorithm for the warehouse location problem,” *Management Science*, vol. 18, no. 12, pp. B718–B731, 1972.

[134] C. Fuchs and F. Moll, “Ground station network optimization for space-to-ground optical communication links,” *IEEE/OSA Journal of Optical Communications and Networking*, vol. 7, no. 12, pp. 1148–1159, Dec 2015.

[135] F. Lacoste, A. Guérin, A. Laurens, G. Azema, C. Periard, and D. Grimal, “FSO ground network optimization and analysis considering the influence of clouds,” in *Proceedings of the 5th European Conference on Antennas and Propagation (EUCAP)*, April 2011, pp. 2746–2750.

[136] D. Giggenbach, B. Epple, J. Horwath, and F. Moll, *Optical Satellite Downlinks to Optical Ground Stations and High-Altitude Platforms.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 331–349.

[137] R. Bedington, J. M. Arrazola, and A. Ling, “Progress in satellite quantum key distribution,” *npj Quantum Information*, vol. 3, no. 1, p. 30, 2017.

[138] M. Hayashi, “Finite-block-length analysis in classical and quantum information theory,” *Proceedings of the Japan Academy, Series B*, vol. 93, no. 3, pp. 9–124, 2017.

[139] The Office of Communications. (2018) Connected Nations 2018. Last accessed 15/10/2019. [Online]. Available: https://www.ofcom.org.uk/research-and-data/multi-sector-research/infrastructure-research/connected-nations-2018

[140] X. Jia, M. Xu, X. Pan, and X. Mao, “Eclipse Prediction Algorithms for Low-Earth-Orbiting Satellites,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 53, no. 6, pp. 2963–2975, Dec 2017.

[141] D. L. Applegate, R. E. Bixby, V. Chvatal, and W. J. Cook, *The Traveling Salesman Problem: A Computational Study (Princeton Series in Applied Mathematics).* USA: Princeton University Press, 2007.

[142] S. Irnich, P. Toth, and D. Vigo, *Chapter 1: The Family of Vehicle Routing Problems.* Society for Industrial and Applied Mathematics, 2014, ch. 1, pp. 1–33. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/1.9781611973594.ch1

[143] G. Desaulniers, O. B. Madsen, and S. Ropke, *Vehicle Routing.* Society for Industrial and Applied Mathematics, 2014, ch. Chapter 5: The Vehicle Routing Problem with Time Windows, pp. 119–159. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/1.9781611973594.ch5

[144] D. Bredström and M. Rönnqvist, "A branch and price algorithm for the combined vehicle routing and scheduling problem with synchronization constraints," Norwegian School of Economics, Department of Business and Management Science, Discussion Papers 2007/7, Feb. 2007. [Online]. Available: https://ideas.repec.org/p/hhs/nhhfms/2007_007.html

[145] A. A. Kovacs, S. N. Parragh, K. F. Doerner, and R. F. Hartl, "Adaptive large neighborhood search for service technician routing and scheduling problems," *Journal of Scheduling*, vol. 15, no. 5, pp. 579–600, 2012.

[146] Y. Li, A. Lim, and B. Rodrigues, "Manpower Allocation with Time Windows and Job-teaming Constraints," *Naval Research Logistics (NRL)*, vol. 52, no. 4, pp. 302–311, 2005.

[147] J. Larsen, A. Dohn, and M. S. Rasmussen, "The vehicle routing problem with time windows and temporal dependencies," in *International workshop in vehicle routing, intermodal transport and related areas*, 2009.

[148] J.-F. Cordeau, G. Laporte, M. W. Savelsbergh, and D. Vigo, "Vehicle Routing," *Handbooks in Operations Research and Management Science*, vol. 14, pp. 367–428, 2007.

[149] A. Trautsamwieser and P. Hirsch, "Optimization of daily scheduling for home health care services," *Journal of Applied Operational Research*, vol. 3, no. 3, pp. 124–136, 2011.

[150] A. Dohn, M. S. Rasmussen, and J. Larsen, “The Vehicle Routing Problem with Time Windows and Temporal Dependencies,” *Networks*, vol. 58, no. 4, pp. 273–289, 2011.

[151] S. Afifi, D.-C. Dang, and A. Moukrim, “Heuristic solutions for the vehicle routing problem with time windows and synchronized visits,” *Optimization Letters*, vol. 10, no. 3, pp. 511–525, Mar 2016.

[152] G. Hiermann, M. Prandtstetter, A. Rendl, J. Puchinger, and G. R. Raidl, “Metaheuristics for solving a multimodal home-healthcare scheduling problem,” *Central European Journal of Operations Research*, vol. 23, no. 1, pp. 89–113, Mar 2015.

[153] S. Nickel, M. Schröder, and J. Steeg, “Mid-term and short-term planning support for home health care services,” *European Journal of Operational Research*, vol. 219, no. 3, pp. 574–587, 2012.

[154] P. M. Duque, M. Castro, K. Sörensen, and P. Goos, “Home care service planning. The case of Landelijke Thuiszorg,” *European Journal of Operational Research*, vol. 243, no. 1, pp. 292–301, 2015.

[155] S. Bertels and T. Fahle, “A hybrid setup for a hybrid scenario: combining heuristics for the home health care problem,” *Computers & Operations Research*, vol. 33, no. 10, pp. 2866 – 2890, 2006, part Special Issue: Constraint Programming.

[156] P. Cappanera and M. G. Scutellà, “Joint assignment, scheduling, and routing models to home care optimization: A pattern-based approach,” *Transportation Science*, vol. 49, no. 4, pp. 830–852, 2015.

[157] G. A. Croes, “A method for solving traveling-salesman problems,” *Operations Research*, vol. 6, no. 6, pp. 791–812, 2021/07/25/ 1958. [Online]. Available: http://www.jstor.org/stable/167074

[158] S. Lin, “Computer solutions of the traveling salesman problem,” *The Bell System Technical Journal*, vol. 44, no. 10, pp. 2245–2269, 1965.

[159] I. Or, “Traveling salesman-type combinatorial problems and their relation to the logistics of regional blood banking,” Ph.D. dissertation, Northwestern University, Illinois, 1976.

[160] K. Helsgaun, “An effective implementation of the Lin–Kernighan traveling salesman heuristic,” *European Journal of Operational Research*, vol. 126, no. 1, pp. 106–130, Oct 2000.

[161] S. Lin and B. W. Kernighan, “An effective heuristic algorithm for the traveling-salesman problem,” *Operations Research*, vol. 21, no. 2, pp. 498–516, 1973. [Online]. Available: http://www.jstor.org/stable/169020

[162] O. C. Martin and S. W. Otto, “Combining simulated annealing with local search heuristics,” *Annals of Operations Research*, vol. 63, no. 1, pp. 57–75, Feb 1996. [Online]. Available: https://doi.org/10.1007/BF02601639

[163] M. Savelsbergh, “A parallel insertion heuristic for vehicle routing with side constraints,” *Statistica Neerlandica*, vol. 44, no. 3, pp. 139–148, 1990.

[164] W. Laesanklang and D. Landa-Silva, “Decomposition techniques with mixed integer programming and heuristics for home healthcare planning,” *Annals of Operations Research*, vol. 256, no. 1, pp. 93–127, 2017.

[165] E. Benzarti, E. Sahin, and Y. Dallery, “Operations management applied to home care services: Analysis of the districting problem,” *Decision Support Systems*, vol. 55, no. 2, pp. 587–598, 2013.

[166] M. Cissé, S. Yalçındağ, Y. Kergosien, E. Şahin, C. Lenté, and A. Matta, “Or problems related to home health care: A review of relevant routing and scheduling problems,” *Operations Research for Health Care*, vol. 13-14, pp. 1 – 22, 2017.

[167] C. Fikar and P. Hirsch, “Home health care routing and scheduling: A review,” *Computers & Operations Research*, vol. 77, pp. 86 – 95, 2017.

[168] J. A. Castillo-Salazar, D. Landa-Silva, and R. Qu, "Workforce scheduling and routing problems: literature survey and computational study," *Annals of Operations Research*, vol. 239, no. 1, pp. 39–67, 2016.

[169] D. C. Paraskevopoulos, G. Laporte, P. P. Repoussis, and C. D. Tarantilis, "Resource constrained routing and scheduling: Review and research prospects," *European Journal of Operational Research*, vol. 263, no. 3, pp. 737 – 754, 2017.

[170] P. Cappanera, L. Gouveia, and M. G. Scutellà, "The skill vehicle routing problem," in *Proceedings of the 5th International Conference on Network Optimization*, ser. INOC'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 354–364.

[171] J. C. Beck, P. Prosser, and E. Selensky, "Vehicle routing and job shop scheduling: What's the difference?" in *Proceedings of the Thirteenth International Conference on International Conference on Automated Planning and Scheduling*, ser. ICAPS'03. AAAI Press, 2003, p. 267–276.

[172] P. Cappanera and M. G. Scutellà, "Joint Assignment, Scheduling, and Routing Models to Home Care Optimization: A Pattern-Based Approach," *Transportation Science*, vol. 49, no. 4, pp. 830–852, 2015.

[173] K. Thomsen, "Optimization on home care," Master's thesis, Informatics and Mathematical Modelling, Technical University of Denmark, DTU, 2006. [Online]. Available: http://localhost/pubdb/p.php?4710

[174] M. S. Rasmussen, T. Justesen, A. Dohn, and J. Larsen, "The Home Care Crew Scheduling Problem: Preference-based visit clustering and temporal dependencies," *European Journal of Operational Research*, vol. 219, no. 3, pp. 598–610, 2012.

[175] E. Rahimian, K. Akartunalı, and J. Levine, "A hybrid integer and constraint programming approach to solve nurse rostering problems," *Computers & Operations Research*, vol. 82, pp. 83–94, 2017.

[176] ——, “A hybrid integer programming and variable neighbourhood search algorithm to solve nurse rostering problems,” *European Journal of Operational Research*, vol. 258, no. 2, pp. 411–423, 2017.

[177] J. Decerle, O. Grunder, A. H. E. Hassani, and O. Barakat, “A memetic algorithm for a home health care routing and scheduling problem,” *Operations Research for Health Care*, vol. 16, pp. 59 – 71, 2018.

[178] C. Voudouris and E. Tsang, “Guided local search and its application to the traveling salesman problem,” *European Journal of Operational Research*, vol. 113, no. 2, pp. 469 – 499, 1999.

[179] M. Mutingi and C. Mbohwa, “Multi-objective homecare worker scheduling: A fuzzy simulated evolution algorithm approach,” *IIE Transactions on Healthcare Systems Engineering*, vol. 4, 10 2014.

[180] A. Trautsamwieser, M. Gronalt, and P. Hirsch, “Securing home health care in times of natural disasters,” *OR Spectrum*, vol. 33, no. 3, pp. 787–813, Jul 2011.

[181] P. Eveborn, P. Flisberg, and M. Rönnqvist, “Laps Care - an operational system for staff planning of home care,” *European Journal of Operational Research*, vol. 171, no. 3, pp. 962–976, 2006.

[182] G. M. Thompson and M. E. Pullman, “Scheduling workforce relief breaks in advance versus in real-time,” *European Journal of Operational Research*, vol. 181, no. 1, pp. 139–155, 2007.

[183] Y. Kergosien, C. Lenté, and J.-C. Billaut, “Home health care problem: An extended multiple Traveling Salesman Problem,” in *4th Multidisciplinary International Conference on Scheduling: Theory and Applications*, Dublin, Ireland, Aug. 2009. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01025660

[184] A. Dohn, E. Kolind, and J. Clausen, "The manpower allocation problem with time windows and job-teaming constraints: A branch-and-price approach - technical report," *Computers & Operations Research*, vol. 36, no. 4, pp. 1145 – 1157, 2009.

[185] A. Lim, B. Rodrigues, and L. Song, "Manpower allocation with time windows," *Journal of the Operational Research Society*, vol. 55, no. 11, pp. 1178–1186, 2004.

[186] M. Drexl, "Synchronization in Vehicle Routing–A Survey of VRPs with Multiple Synchronization Constraints," *Transportation Science*, vol. 46, no. 3, pp. 297–316, 2012.

[187] S. N. Parragh and K. F. Doerner, "Solving routing problems with pairwise synchronization constraints," *Central European Journal of Operations Research*, vol. 26, no. 2, pp. 443–464, 2018.

[188] L. En-nahli, S. Afifi, H. Allaoui, and I. Nouaouri, "Local search analysis for a vehicle routing problem with synchronization and time windows constraints in home health care services," *IFAC-PapersOnLine*, vol. 49, no. 12, pp. 1210 – 1215, 2016.

[189] S. Frifita, M. Masmoudi, and J. Euchi, "General variable neighborhood search for home healthcare routing and scheduling problem with time windows and synchronized visits," *Electronic Notes in Discrete Mathematics*, vol. 58, pp. 63 – 70, 2017.

[190] R. B. Bachouch, A. Guinet, and S. Hajri-Gabouj, "A decision-making tool for home health care nurses' planning," *Supply Chain Forum: An International Journal*, vol. 12, no. 1, pp. 14–20, 2011.

[191] R. Liu, Y. Tao, and X. Xie, "An adaptive large neighborhood search heuristic for the vehicle routing problem with time windows and synchronized visits," *Computers & Operations Research*, vol. 101, pp. 250 – 262, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S030505481830220X

[192] D. S. Mankowska, F. Meisel, and C. Bierwirth, "The home health care routing and scheduling problem with interdependent services," *Health Care Management Science*, vol. 17, no. 1, pp. 15–30, Mar 2014.

[193] M. Bernardo and J. Pannek, "Robust solution approach for the dynamic and stochastic vehicle routing problem," *Journal of Advanced Transportation*, vol. 2018, p. 9848104, Mar 2018. [Online]. Available: https://doi.org/10.1155/2018/9848104

[194] F. Ordóñez, *Robust Vehicle Routing*, ser. INFORMS TutORials in Operations Research. INFORMS, Sep 2010, pp. 153–178, 0. [Online]. Available: https://doi.org/10.1287/educ.1100.0078

[195] J. F. Ehmke, A. M. Campbell, and T. L. Urban, "Ensuring service levels in routing problems with time windows and stochastic travel times," *European Journal of Operational Research*, vol. 240, no. 2, pp. 539 – 550, 2015.

[196] A. Agra, M. Christiansen, R. Figueiredo, L. M. Hvattum, M. Poss, and C. Requejo, "The robust vehicle routing problem with time windows," *Computers & Operations Research*, vol. 2, pp. 273–287, Mar 2013. [Online]. Available: https://doi.org/10.1016/j.cor.2012.10.002A

[197] T. Dokka and M. Goerigk, "An experimental comparison of uncertainty sets for robust shortest path problems," 2017.

[198] M. Goerigk and A. Schöbel, "A scenario-based approach for robust linear optimization," in *Theory and Practice of Algorithms in (Computer) Systems*, A. Marchetti-Spaccamela and M. Segal, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 139–150.

[199] F. Gayraud, "Vehicle routing problem with synchronization constraints in home care support services - Mathematical formulation and hybridization based on metaheuristics," Theses, Université Blaise Pascal - Clermont-Ferrand II, Jul. 2015. [Online]. Available: https://tel.archives-ouvertes.fr/tel-01289845

[200] D. Luxen and C. Vetter, “Real-time routing with OpenStreetMap data,” in *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ser. GIS ’11. New York, NY, USA: ACM, 2011, pp. 513–516.

[201] Google, “Google Maps,” https://developers.google.com/maps/documentation/, 2018, accessed: 2018-07-25.

[202] MOSEK ApS, “Modeling cookbook,” 2020, accessed 30/03/2020. [Online]. Available: https://docs.mosek.com/modeling-cookbook/index.html

[203] K. P. Burnham and D. R. Anderson, “Multimodel inference: Understanding AIC and BIC in model selection,” *Sociological Methods & Research*, vol. 33, no. 2, pp. 261–304, 2004.