SAPIENZA
UNIVERSITÀ DI ROMA

Optimization under Uncertainty for Supply Chain and Logistics Management

Department of computer, control and management engineering ANTONIO RUBERTI (DIAG)
Automatic Control, Bioengineering and Operations Research (ABRO)
(XXXVIII cycle)

Francesca Giancola
ID number 1715739

Advisor
Sara Mattia, Claudio Gentile

Director
Prof. Laura Palagi

Academic Year 2025/2026

This PhD scholarship was funded by the Istituto di Analisi dei Sistemi
e Informatica "Antonio Ruberti" (IASI-CNR).

Optimization under Uncertainty for Supply Chain and Logistics Management
PhD thesis. Sapienza University of Rome

© 2026 Francesca Giancola. All rights reserved

This thesis has been typeset by LaTeX and the Sapthesis class.

Author's email: giancola@diag.uniroma1.it

Abstract

Supply chains and logistics networks are complex, multi-actor systems in which decisions at every level—strategic, tactical, and operational—are affected by pervasive uncertainty. This thesis addresses three distinct problems arising in supply chain and logistics management under uncertainty, each tackled with a different combination of methodologies, reflecting the diverse nature of the problems considered. The problems are organized around two operational contexts in which uncertainty plays a central and mathematically challenging role. The first is the *procurement and planning* domain: effective purchase decisions require accounting for lead time variability, since delays in supplier deliveries directly translate into excess inventory costs or unmet demand. The second is *traffic congestion under uncertain demand*: road networks are the physical backbone of supply chains, and congestion inflates lead times, increases logistics costs, and introduces delivery variability that propagates upstream through the entire supply chain. Understanding and managing uncertainty in both domains is therefore essential for building resilient and cost-efficient supply chain operations.

In particular, the first contribution concerns supplier selection and purchase planning under lead time uncertainty. A buyer must decide which suppliers to activate and how much to order in each period, before actual delivery delays are known, so as to minimize worst-case total costs—purchase prices, fixed ordering fees, inventory holding, and backlogging penalties. We model the problem as a two-stage robust optimization problem over a structured uncertainty set that captures both a budget on the total number of late deliveries and temporal correlations across consecutive orders from the same supplier. Two solution methods are proposed: an exact branch-and-cut algorithm based on a Benders-like reformulation, and a horizon-split heuristic for large instances. A comprehensive computational study reveals managerial insights on procurement strategies and the cost of robustness.

The second contribution studies worst-case traffic congestion under demand uncertainty. We formulate a bilevel optimization model in which a traffic planner maximizes congestion by selecting the origin–destination demand vector, while travelers respond by routing themselves according to Wardrop equilibrium principles—either the user equilibrium or the system optimum. Three families of uncertainty sets are considered: budgeted, ellipsoidal, and hose. The bilevel problem is reformulated as a mixed-integer nonlinear program via KKT optimality conditions, and several strengthening techniques are developed. Computational experiments on the Sioux Falls network and SNDlib instances demonstrate the effectiveness of the approach and provide insights into the impact of different congestion measures and uncertainty models.

The third contribution presents a complete, reproducible data pipeline for city-scale travel demand modelling, implemented in the MATSim agent-based simulation framework for the city of Barcelona. The pipeline synthesizes population, network, and multi-modal transport demand from open-access statistical data, and is calibrated and validated against observed mobility statistics. The resulting simulator serves as both an empirical foundation for the uncertainty sets used in the congestion model and a testbed for evaluating urban mobility and logistics interventions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context and Motivation

Supply chains and logistics Management

In today's globalized economy, supply chains have evolved into large-scale, interdependent systems whose efficient management is widely recognized as a decisive source of competitive advantage. The increasing complexity of these networks—driven by global sourcing, rising customer expectations, and the proliferation of product variety—has placed supply chain management (SCM) at the center of both academic research and industrial practice. Effective SCM is no longer a mere operational concern; it is a strategic imperative that determines whether organizations can deliver the right product, at the right time, at the right cost. A supply chain is defined as a network of organizations, processes, and flows through which raw materials are sourced, transformed into products, and delivered to end customers. It involves several actors and decision-makers: suppliers, manufacturers, distributors, retailers, and logistics service providers and requires the coordination of a wide range of activities, including procurement, production planning, inventory management, transportation, and distribution. Supply chain management is the discipline of integrating and coordinating these activities across organizational boundaries so that the chain as a whole operates efficiently and responsively. Logistics, a central component of SCM, deals specifically with the planning and execution of the physical movement and storage of goods: transportation routing and scheduling, warehousing, inventory management, and last-mile distribution.

It becomes clear that supply chains and logistics are complex systems characterized by multiple interacting components, hierarchical decision-making,and pervasive uncertainty. Relevant decisions are made at multiple levels by multiple actors, and decisions at one level constrain and shape those at others. At the strategic level, organizations make long-horizon, hard-to-reverse choices: where to locate production facilities and distribution centers, which suppliers to include in the network, how to design the transportation infrastructure. At the tactical level, they plan over a medium-term horizon: production schedules, procurement quantities, inventory replenishment policies, fleet sizing. At the operational level, day-to-day decisions include vehicle routing and scheduling, order dispatching, and real-time traffic management.

Crucially, the actors involved at each level—suppliers, manufacturers, retailers, logistics operators, public authorities—often have misaligned objectives. A road authority sets tolls or traffic policies, and individual drivers react by choosing different routes; a manufacturer fixes wholesale prices, and downstream retailers adjust their order quantities accordingly. This interplay gives supply chain and transportation systems a naturally hierarchical structure in which no single actor controls the entire system, yet each actor's decisions affect all others.

Beyond the multi-actor structure, supply chain and logistics systems are further complicated by uncertainty and correlation in their data. Uncertainty pervades every planning level: demand fluctuates due to market dynamics and unpredictable events, lead times vary because of supplier reliability and transportation disruptions, and travel times on road networks change with congestion and weather conditions. Moreover, supply chain parameters are often correlated—for instance, a disruption in a shared transportation corridor may simultaneously delay deliveries from multiple suppliers, or congestion on one road segment may cascade to adjacent links.

Finally, there is a strong and critical interconnection among supply chain parameters and decisions. One of the most known examples to illustrate this phenomenon is the so-called "bullwhip effect", in which minor fluctuations in end-customer demand generate increasingly larger and distorted swings in inventory and order quantities as they propagate upstream in the supply chain from retailers to manufacturers. Typically triggered by delayed information sharing, panic ordering, or extended lead times, it results in excess inventory accumulation, elevated storage costs, and potential supply shortages. This phenomenon exemplifies how the interplay of multi-level structure, multiple interacting actors, and pervasive uncertainty can lead to complex dynamics that are difficult to predict and manage, yet have significant implications for supply chain performance. This is the proof of how fundamental it is to make informed decisions, at each level of the supply chain, that account for these complexities, and to develop tools and methodologies that can support decision-makers in navigating this challenging landscape.

Over the past decades, a rich set of approaches has been developed for this purpose, including mathematical optimization techniques—such as robust optimization, stochastic programming, and bilevel optimization—as well as simulation-based methods, notably discrete-event simulation and agent-based simulation. These tools support decision-making in settings where data is uncertain, stakeholders interact strategically, and system dynamics are too complex for closed-form analysis. Different methodologies are suited to different types of problems, and the choice of the right tool is a critical step in the research process. For instance, robust optimization is particularly effective for problems with uncertain parameters that can be bounded within a known set, while stochastic programming is more appropriate when a probability distribution over the uncertain parameters is available. Bilevel optimization is the natural framework for problems with hierarchical decision-making and strategic interactions, while simulation methods are invaluable for capturing complex system dynamics and emergent phenomena that are difficult to model analytically.

This thesis contributes to this line of research by addressing three distinct problems arising in supply chain and logistics management under uncertainty. Each contribution employs a different combination of methodologies, reflecting the diverse nature of the problems considered.

1.2 Contributions of this thesis

The themes introduced in the previous section—supply chain management, logistics, data uncertainty, correlated parameters, hierarchical decision-making, and the need for optimization and simulation methodologies—define the research landscape of this thesis.
In particular, this thesis focuses on two operational contexts within SCM where uncertainty plays a central and mathematically challenging role: the planning and procurement domain, and traffic demand and congestion in urban transportation networks.

Within the first context, i.e., the planning and procurement domain,we address the tactical-level decisions of supplier selection and purchase planning under lead time uncertainty. Given a time horizon and periodic demands for a single product, a buyer must decide which suppliers to source from and how much to order from each, balancing competing objectives: minimizing purchase prices and fixed ordering costs while avoiding the twin penalties of excess inventory—which ties up working capital—and backlogging, i.e., unmet demand—which erodes service levels and customer satisfaction. In practice, this already difficult trade-off is compounded by lead time uncertainty: the elapsed time between placing an order and receiving the delivery varies with supplier reliability, transportation conditions, and external disruptions. Orders are committed before the actual lead times are known, so the buyer cannot wait to observe delays before deciding how much to stock. This temporal decoupling—decisions before, uncertainty realization after—naturally leads to a two-stage optimization framework. In this thesis, we adopt a two-stage robust optimization approach in which decisions are partitioned into a first stage—choices committed before the uncertain parameters are revealed (here: which suppliers to activate and how much to order)—and a second stage, representing the recourse actions taken once the actual lead times are known (inventory adjustments and backlogging decisions). Rather than requiring a probability distribution over the uncertain lead times, the model optimizes first-stage decisions against the worst-case realization within a prescribed uncertainty set: a compact region that encodes all plausible combinations of delays without prior distributional assumptions. We define an uncertainty set that captures correlations across time periods and suppliers (e.g., simultaneous delays on shared shipping routes) and bounds the total number of late deliveries via a budget parameter, controlling the degree of conservatism.
Two solution methods are proposed: (i) an exact algorithm, a branch-and-cut method based on a Benders-like reformulation and (ii) a horizon-split heuristic that solves the near-term periods exactly and approximates the remaining horizon, yielding near-optimal solutions for large instances at a fraction of the computational cost.
This research was primarily conducted at the Institute for Systems Analysis and Computer Science of the National Research Council of Italy (IASI-CNR), under the supervision of Claudio Gentile (IASI-CNR) and Sara Mattia (IASI-CNR). Based on this work, a paper has been produced and submitted for publication to an international journal and is currently under review.

Within the second context, we study traffic demand and congestion in urban transportation networks, a problem that sits at the interface of SCM, logistics and urban mobility. Road networks are the physical backbone of supply chains: every shipment from a supplier, every delivery to a customer, and every inter-depot transfer must traverse road networks shared with all other urban and inter-urban mobility flows. Congestion inflates lead times, increases logistics costs, and introduces delivery variability that propagates upstream through the supply chain. For logistics network designers, toll-setting authorities, and supply chain managers, understanding the worst-case congestion that can arise under uncertain demand conditions is therefore a prerequisite for building resilient transportation infrastructures and robust logistics plans. In this thesis, we address two complementary problemans in this domain: (1) estimating the worst-case congestion that can arise when travel demand is uncertain (a problem with direct relevance for network design and toll-setting), and (2) designing a pipeline to model the travel demand itself and validating it through an agent-based simulation that generates realistic, city-level demand data and congestion estimations. These two aspects are naturally linked: the simulation could provide the empirical grounding for the uncertainty sets used in the congestion model. Moreover, it could represent an alternative methodology for estimating worst-case congestion, complementing the optimization-based approach with a data-driven, simulation-based one.

In the following, we provide a more detailed overview of each of these two aspects, outlining the main features of the problem, the methodological approach, and the key results.

The first problem has been addressed with a more methodological focus, developing a novel optimization model that combines bilevel optimization with Wardrop equilibrium principles to estimate worst-case congestion under demand uncertainty.
Three features can be identified as central to this problem: (i) the hierarchical structure of the decision-making process, in which a traffic planner seeks to identify the worst-case congestion, while travelers independently choose their routes; (ii) the equilibrium behavior of travelers, who react to the planner's choice by selecting their routes according to a Wardrop principle; and (iii) the uncertainty in travel demand, which is represented through a suitable model of uncertainty. These three features —hierarchical decisions, equilibrium behavior, and demand uncertainty—call for a combination of bilevel optimization, Wardrop equilibrium models, and robust optimization. Formally, we formulate the problem as a bilevel optimization problem in which the leader (the traffic planner) aims to maximize the congestion and selects the demand, while the follower (the travelers) selects the optimal flows on the network according to a Wardrop Principle (considering both the User Equilibrium and System Optimum cases). The uncertain demand is represented, following the robust optimization paradigm, by an uncertainty set—a compact region within which the origin–destination demand vector is assumed to vary. Three families of uncertainty sets are considered: budgeted (bounding the total demand deviation), ellipsoidal (accounting for correlations), and hose (constraining only aggregate traffic at each node), each capturing a different notion of what "plausible demand variation" means. The resulting bilevel problem is reformulated as a mixed-integer nonlinear program (MINLP) by replacing the lower-level equilibrium conditions with their Karush–Kuhn–Tucker (KKT) optimality conditions, introducing binary variables and valid big-M constants to handle the complementarity constraints. Several strengthening techniques improve computational performance. An extensive computational study on well-known benchmark networks (the Sioux Falls network and SNDlib instances) serves a dual purpose: it demonstrates the effectiveness and scalability of the proposed solution approach, and it yields managerial insights into how key modeling choices—the congestion measure, the travel cost function, and the uncertainty model—affect the worst-case congestion level.
This research was primarily developed by the author during a research visiting period at ESSEC Business School in Paris, under the supervision of Professor Ivana Ljubić (ESSEC Business School) and in collaboration with Yasmine Beck (ESSEC Business School) and Sara Mattia

(IASI-CNR). Based on this work, a paper has been produced and submitted for publication to an international journal and is currently under review.

Finally, the second problem has been addressed with a more empirical and applied focus, developing a complete, reproducible data modelling pipeline for the city of Barcelona using the MATSim agent-based simulation framework [72]. The primary objective of this problem is to estimate and model urban travel demand at city scale, producing a tool that supports the understanding of traffic flows and their efficiency—with direct consequences both on citizens' quality of life and on the transportation and logistics activities that constitute the physical backbone of supply chain operations. The pipeline synthesizes population, network, and multi-modal transport demand from open-access statistical data sources; the resulting model is calibrated and validated against observed mobility statistics, and serves a dual role: it provides empirical grounding for the uncertainty sets used in the congestion model, and it acts as a testbed for evaluating urban mobility interventions such as network redesigns, charging station placement, or public transport fleet dimensioning. In agent-based simulation, each individual (here, each urban traveler) is explicitly represented as an autonomous agent endowed with attributes and behavioral rules; system-level outcomes—traffic flows, congestion, modal splits—emerge from the aggregate of millions of individual interactions rather than being prescribed by a top-down equation. MATSim implements an activity-based variant in which each agent carries a daily plan (a sequence of activities and trips) and iteratively re-optimizes it in response to experienced congestion via a co-evolutionary algorithm, converging to a stochastic user equilibrium.
This research was developed in collaboration with Spindox S.p.A. within the framework of the European-funded research project "Optimizing Car-sharing and Ride-sharing Mobility in Smart Sustainable Cities", coordinated by Professor Angel A. Juan (Universitat Oberta de Catalunya) and participated by Spindox among other partners. The project was co-funded by the Barcelona City Council and the "la Caixa" Foundation under the Barcelona Science Plan 2020–2023 (grant 21S09355-001), with the objective of providing efficient solutions to urban mobility problems through a new family of agile-optimization algorithms capable of processing big data to support real-time decision-making.
Based on this work, a paper has been published in the proceedings of the 2023 Winter Simulation Conference (WSC 2023), the premier international conference on simulation. The paper is co-authored with Jonas F. Leon (Spindox S.A. / Universitat Oberta de Catalunya), Andrea Boccolucci (Spindox S.p.A.), and Mattia Neroni (Spindox S.p.A.) [85].
Figure 1.1 provides a visual summary of the three contributions of this thesis, highlighting the different problem contexts within the supply chain and logistics domain.

1.3 Thesis Outline

The remainder of this thesis is organized as follows.
Chapter 2 introduces the background notions on supply chains, supply chain management, and logistics that are necessary to contextualize the contributions of this thesis. It discusses the main criticalities of supply chain systems, examines the link between traffic congestion and logistics performance, and outlines the role of mathematical optimization and simulation in addressing uncertainty.
Chapter 3 presents the underlying theoretical framework and the methodological tools neces-

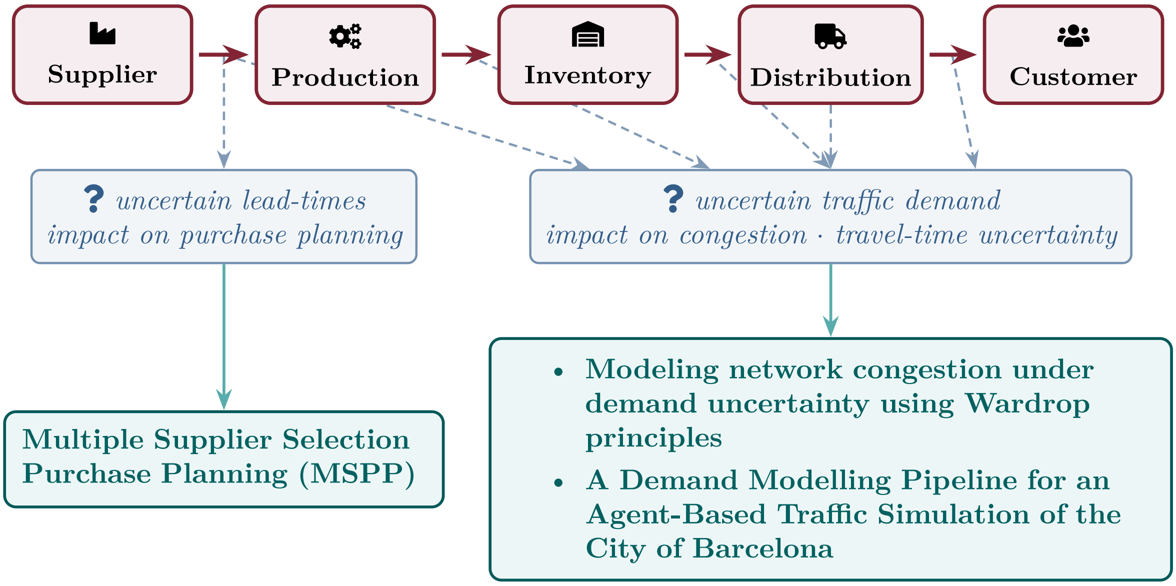


Figure 1.1. Visual summary of the thesis contributions.

sary to interpret and understand our contributions. In particular, it introduces the Branch&Cut method for solving mixed-integer linear programs, the Benders decomposition algorithm, the two-stage optimization problems, the principles of bilevel optimization and the fundamentals of agent-based simulation.

Chapters 4, 6, and 6 present the three main contributions of this thesis, each addressing a different problem in supply chain and logistics management under uncertainty and employing a different methodological approach. In particular, Chapter 4 focuses on the problem of Multiple Supplier Selection and Purchase Planning under Lead time uncertainty and adopts a two-stage robust optimization approach; Chapter 6 addresses the problem of Modeling Network Congestion under Demand Uncertainty using Wardrop Principles and employs a bilevel optimization framework; finally, Chapter 6 presents the problem of defining a Demand Modelling Pipeline for an Agent-Based Traffic Simulation of the City of Barcelona through an agent-based simulation.

Chapter 2

Background: Supply Chains, Logistics, and Optimization Approaches

This chapter provides the conceptual and technical foundations underlying the research presented in this thesis. We begin by defining the core notions of supply chain, supply chain management, and logistics (Section 2.1). We then examine in detail the principal functional areas that characterize modern supply chains (Section 2.2), followed by a discussion of the main criticalities inherent in these systems (Section 2.3). Section 2.4 introduces traffic congestion as a performance-critical factor for logistics. Finally, Sections 2.5 and 2.6 review the role of mathematical optimization and the challenge of uncertainty that motivate the methodological choices of this thesis.

2.1 Supply Chain, Supply Chain Management, and Logistics

A *supply chain* can be defined as a network of organizations involved, through upstream and downstream linkages, in the different processes and activities that produce value in the form of products and services delivered to the ultimate consumer [44]. In its simplest form, a supply chain consists of at least three entities—a supplier, a focal company, and a customer—linked by the flow of products, services, finances, and information [100]. In practice, however, modern supply chains are highly complex networks spanning multiple tiers of suppliers, manufacturers, distributors, retailers, and logistics providers, often operating across several continents [130].
Supply chain management (SCM) is the task of integrating organizational units along a supply chain and coordinating material, information, and financial flows in order to fulfill customer demands, with the aim of improving the competitiveness of the supply chain as a whole [130]. This definition, broadly consistent with the one adopted by the Council of Supply Chain Management Professionals (CSCMP), captures the two main pillars of SCM: *integration* of the organizations involved and *coordination* of the flows among them. [100] provide an extensive review of SCM definitions and identify three complementary perspectives: SCM as a management philosophy (a systems approach to viewing the supply chain as a whole), as a set of activities to implement that philosophy (information sharing, risk and reward sharing, cooperation, process integration), and as a set of management processes (demand management, order fulfillment, manufacturing flow management, procurement). The unifying principle is

that competition has shifted from individual firms to supply chains: the competitiveness of a product in the eyes of the end customer depends on the performance of the supply chain as a whole.

Logistics is a foundational discipline within SCM. It encompasses the planning, execution, and control of the efficient forward and reverse flow of goods, services, and related information between the point of origin and the point of consumption, with the goal of meeting customer requirements [44]. Logistics activities include transportation, warehousing, inventory management, order processing, and the design of distribution networks. Its governing principles—systems thinking, total cost optimization, and service-level orientation—aim at achieving efficiency across the economic, technological, and ecological dimensions simultaneously [130]. While logistics focuses primarily on the physical movement and storage of goods, SCM broadens this scope to include the strategic integration of all partners and the coordination of all flows—material, information, and financial—across the entire network.

Figure 2.1 provides an integrated overview of a supply chain. The *top layer* depicts the physical flow of goods through five canonical stages: suppliers, who provide raw materials and components; production facilities, which transform inputs into finished products; warehouses, which store and consolidate goods; distribution centers, which dispatch shipments toward final destinations; and customers, who represent the end demand. The *middle layer* zooms into the network structure, showing how multiple entities at each stage are interconnected by material flows. The figure also highlights the main sources of *uncertainty* and the principal *cost components* that characterize supply chain operations. On the uncertainty side, information uncertainty (e.g., the bullwhip effect and forecast errors) and disruption risk (e.g., natural disasters and pandemics) span the entire network; travel-time uncertainty affects all inter-stage links; production, capacity, and delivery uncertainties are localized at the manufacturing, warehousing, and distribution stages, respectively; supply and demand uncertainties act at the two extremes of the chain. On the cost side, each stage carries characteristic cost components—material, manufacturing, inventory, distribution, and service costs—while transportation costs arise on every inter-stage link. Two transversal cost layers underlie the whole network: coordination and IT costs (ERP systems, data integration, contracts) and management costs (overhead, administration, regulatory compliance). Understanding these cost and uncertainty elements is essential for formulating the optimization models discussed later in this thesis.

2.2 Functional Areas of Supply Chain Management

The decision problems introduced above can be organized into five complementary functional areas, each addressing a different aspect of supply chain management and spanning one or more decision levels (strategic, tactical, operational). These areas map to the network stages they primarily impact: *Planning and Procurement* governs suppliers, manufacturers, and warehouses; *Production and Manufacturing* focuses on the manufacturing stage; *Warehousing and Inventory Management* covers warehouses and distributors; *Transportation and Network Management* spans every inter-stage link; and *Demand Forecasting and Analytics* propagates its outputs to all five stages. A cross-functional coordination layer underpins the entire system.

For a comprehensive treatment of these topics, the reader is referred to Stadtler et al. [130] and Christopher [44] for general supply chain management foundations; García-Flores and Wang [63] for an operations-research perspective on SCM; Simchi-Levi et al. [128] for inventory and logistics theory; and Daskin [51] for network optimization. The subsections below describe each area in detail, citing only the references most specific to each decision problem.

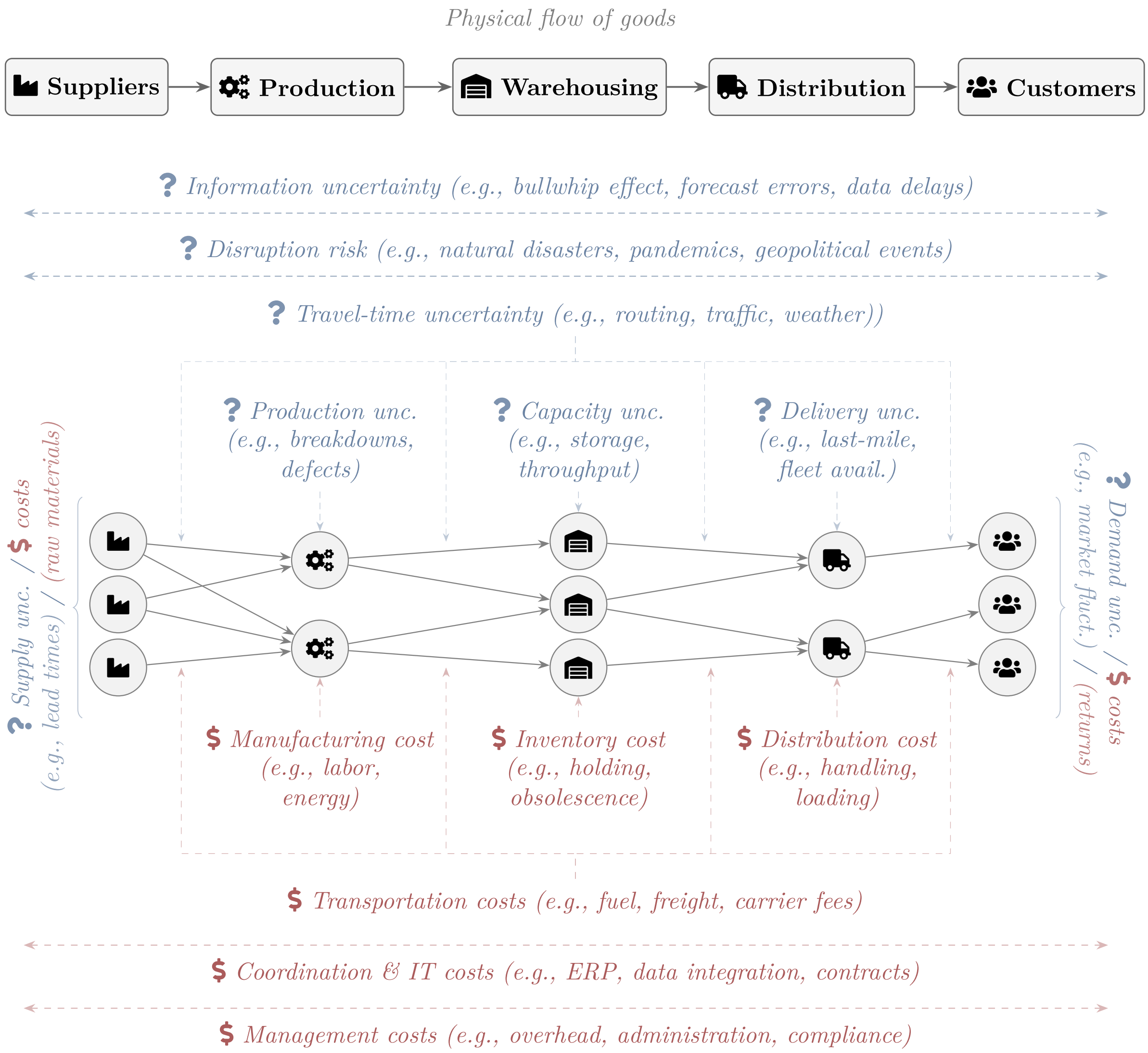


Figure 2.1. Integrated view of the supply chain. *Top:* the physical flow of goods across five stages. *Middle:* the network connecting multiple suppliers, manufacturers, warehouses, distributors, and customers; sources of uncertainty and cost elements are indicated alongside the network.

2.2.1 Transportation and Network Management

Transportation is responsible for the physical movement of raw materials, work-in-process items, and finished goods across the supply chain network. This area impacts every inter-stage link in the chain and spans all three decision levels—strategic (network design), tactical (mode and carrier selection), and operational (daily routing and scheduling). The main decision problems, ordered from the strategic to the operational horizon, are the following:

- **Network design and facility location.** At the strategic level, the design of the distribution network—deciding how many facilities to open and where to locate warehouses, distribution centers, and cross-docking facilities—determines the structure of all downstream transportation.

- **Fleet management.** Fleet sizing, vehicle replacement, and maintenance policies are strategic and tactical decisions that constrain the capacity available for routing and scheduling.

- **Mode and carrier selection.** Choosing the appropriate transportation mode (road, rail, maritime, air, or intermodal combinations) and the corresponding carrier is a tactical decision that balances cost, transit time, reliability, and environmental impact.

- **Vehicle routing and scheduling.** At the operational level, the vehicle routing problem (VRP) and its many variants—capacitated (CVRP), time-windowed (VRPTW), stochastic, multi-depot—seek to determine the optimal set of routes for a fleet of vehicles to service a set of customers, while respecting capacity, time, and distance constraints. Scheduling adds the temporal dimension, sequencing pickups and deliveries within planning horizons.

Transportation decisions are strongly affected by external factors such as traffic congestion, weather, and regulatory constraints. In particular, the interaction between freight logistics and urban traffic is a critical issue for supply chain performance, as discussed in Section 2.4.

2.2.2 Production and Manufacturing

Production and manufacturing activities transform raw materials into finished products. This area primarily impacts the manufacturing stage and spans all three planning levels—strategic (plant location and technology), tactical (capacity and master scheduling), and operational (job sequencing). The main decision problems are:

- **Plant location and technology choice.** Strategic decisions on where to locate production plants and which manufacturing technologies to adopt determine the long-term capacity and cost structure of the supply chain.

- **Master production scheduling.** At the tactical level, the master production schedule (MPS) translates aggregate plans into detailed production targets for individual products over a planning horizon.

- **Capacity planning.** Tactical-level decisions on workforce levels, overtime, and subcontracting ensure that production capacity matches anticipated demand over the medium term.

- **Lot-sizing.** Determining the production quantities for each product in each period involves balancing fixed setup costs against holding costs.

- **Production scheduling and sequencing.** Operational-level decisions determine the order in which jobs are processed on machines, with the goal of minimizing makespan, tardiness, or setup costs, while respecting due dates and machine availability.

2.2.3 Planning and Procurement

Planning and procurement activities ensure that the right materials are available at the right time and cost. This area impacts the supplier, manufacturer, and warehouse stages and operates primarily at the strategic and tactical levels. The main decision problems are:

- **Strategic sourcing.** Long-term decisions on single vs. dual sourcing, domestic vs. global procurement, and make-or-buy policies shape the structure and risk profile of the entire supply chain.
- **Supplier selection and evaluation.** At the strategic–tactical boundary, choosing suppliers involves multi-criteria assessment of cost, quality, reliability, lead time, flexibility, and sustainability. Supplier evaluation is a recurring process that feeds into long-term sourcing strategies.
- **Contract design and management.** Supply contracts—including wholesale-price, revenue-sharing, and quantity-flexibility contracts—are tactical mechanisms to coordinate decisions between buyers and suppliers and to allocate risk and profit along the chain.
- **Purchase and order planning.** Once suppliers are selected, purchase plans define the quantities, timing, and frequency of orders. Multi-period procurement problems must balance ordering costs, quantity discounts, and inventory holding costs under demand and lead-time uncertainty.

2.2.4 Warehousing and Inventory Management

Warehousing and inventory management bridge production and demand, buffering the supply chain against variability and ensuring product availability. This area impacts the warehouse and distributor stages and operates mainly at the tactical and operational levels. The main decision problems are:

- **Warehouse design and operations.** At the strategic–tactical level, decisions on warehouse layout, storage policy (random, class-based, dedicated), and the automation level affect both space utilization and order-picking efficiency.
- **Inventory control policies.** Classical models provide rules for when and how much to order. Modern extensions incorporate stochastic demand, lead-time variability, and service-level constraints.
- **Safety stock optimization.** Safety stocks protect against demand and supply uncertainty. Their dimensioning depends on the desired service level, demand variability, and replenishment lead time.
- **Distribution and cross-docking.** At the operational level, cross-docking facilities consolidate and re-sort incoming shipments for immediate outbound delivery, reducing storage time and inventory holding costs.

2.2.5 Demand Forecasting and Analytics

Demand forecasting is a prerequisite for virtually every other supply chain activity: production planning, procurement, inventory management, and transportation all depend on reliable estimates of future demand. The outputs of this area propagate to every stage of the network, from suppliers to customers, and its decisions span all three levels—strategic (long-range demand projections), tactical (seasonal planning), and operational (short-term demand sensing). The main approaches range from classical *causal and time-series models* (econometric regressions, exponential smoothing, ARIMA), which support strategic and tactical planning, to *machine learning methods* (neural networks, gradient boosting) that capture nonlinear patterns for short-term forecasting, and *demand sensing* techniques that incorporate real-time signals (e.g., point-of-sale data, weather) into operational planning.

2.3 Criticalities in Supply Chain Management

Managing these intertwined activities is a formidable challenge due to several intrinsic characteristics of supply chain systems:

Multiple independent decision-makers. A supply chain typically involves several legally separated organizations, each pursuing its own objectives. Decisions made by one entity affect all others, yet no single actor controls the entire system. This multi-player structure gives rise to strategic interactions, incentive misalignment, and the need for coordination mechanisms such as contracts, revenue-sharing schemes, or collaborative planning.

Interdependence across decisions. Supply chain decisions are deeply interconnected: procurement choices affect production schedules, which in turn drive inventory levels and transportation needs. Optimizing any single activity in isolation may lead to suboptimal outcomes for the chain as a whole—a phenomenon well captured by the "bullwhip effect", in which small demand fluctuations at the retail level are amplified into large order swings upstream.

Multi-scale temporal structure. Decisions span very different time horizons: strategic choices (network design, facility location) involve years of planning; tactical decisions (master planning, inventory policies) operate on a monthly or quarterly basis; and operational decisions (scheduling, routing) must be made daily or in real time. These levels are hierarchically linked, and each phase depends on the others.

Uncertainty and variability. Real-world supply chains operate in environments pervaded by uncertainty. Demand fluctuates due to seasonality, market dynamics, and unpredictable events; lead times vary because of supplier reliability, transportation disruptions, and customs delays; costs and prices are volatile. Ignoring this uncertainty leads to solutions that perform poorly when confronted with actual conditions.

Dynamism and globalization. Supply chains must continuously adapt to changing market requirements, technological developments, evolving regulations, and disruptions (natural disasters, pandemics, geopolitical events). Global sourcing increases complexity by adding exchange-rate risk, cross-cultural communication challenges, and longer lead times.

2.4 Traffic Congestion and Supply Chain Performance

Because a large share of supply chain flows depends on road transport, *traffic congestion* emerges as a critical factor for logistics performance. Empirical evidence shows that congestion directly affects delivery times, transportation costs, and schedule reliability, with cascading consequences on inventory management, vehicle fleet sizing, and the displacement of business locations [124, 142]. From a strategic standpoint, recent work on supply chain network design has demonstrated that planners should explicitly account for traffic congestion when making location and allocation decisions, since it influences not only the economic efficiency of the network but also its environmental sustainability [8, 9]. Traffic congestion can therefore be regarded both as a disruptive factor to be mitigated in day-to-day logistics operations and as a performance indicator for evaluating the effectiveness of strategic supply chain decisions [142, 124].

2.5 Quantitative Methods: Optimization and Simulation

The complexity of supply chain decision problems has motivated the development of a broad arsenal of quantitative methods, *mathematical optimization* and *simulation* among the most widely used.

Mathematical Optimization. Mathematical optimization provides a rigorous framework for making the best possible decisions under given constraints and objectives. In the context of supply chain and logistics management, optimization models allow practitioners to:

- minimize total costs (procurement, production, transportation, inventory, and backlogging costs);
- maximize service levels and on-time delivery performance;
- balance trade-offs between conflicting objectives (e.g., cost vs. responsiveness);
- identify bottlenecks and critical points in the supply chain;
- evaluate and compare alternative strategies before implementation.

Classical approaches rely on well-established mathematical programming techniques, including linear programming (LP), mixed-integer linear programming (MILP), and nonlinear programming (NLP). These models have been successfully applied to a wide spectrum of logistics problems, ranging from vehicle routing [136] and facility location [51] to lot-sizing [118] and network design [92]. A critical limitation of deterministic models, however, is that they assume all input data—demands, costs, travel times, lead times—are known with certainty. In practice, this assumption rarely holds, which has motivated extensions such as Stochastic programming, Robust optimization, and Distributionally robust optimization. Multi-player interactions have been modeled using game-theoretic frameworks, including cooperative games, non-cooperative games, and Bilevel optimization (Stackelberg games) [38].

Simulation. Simulation provides a complementary approach that is especially valuable when the system under study is too complex, stochastic, or decentralized for closed-form analytical solutions. *Discrete-event simulation* models the system as a sequence of events occurring at specific points in time, allowing practitioners to evaluate the performance of supply chain policies (e.g., inventory rules, scheduling heuristics) under realistic stochastic conditions. *Agent-based simulation* goes further by representing each decision-maker (e.g., vehicle, customer, supplier) as an autonomous agent with its own rules and objectives, making it possible to study emergent behaviors in decentralized systems. Agent-based models are particularly suited to transportation and mobility applications, where thousands of individual travelers interact simultaneously on a shared network.

Both optimization and simulation can be combined in hybrid frameworks—for instance, using simulation to evaluate solutions proposed by an optimizer, or embedding optimization within each agent's decision logic. The specific methodologies adopted in this thesis are presented in detail in Chapter 3.

2.6 The Challenge of Uncertainty

Real-world supply chains operate in environments characterized by pervasive uncertainty. Demand fluctuations are driven by market dynamics, seasonality, and unpredictable events; lead times vary due to supplier reliability, transportation disruptions, and customs delays; travel times on transportation networks change due to congestion, weather, and incidents. Ignoring this uncertainty in optimization models can lead to solutions that perform poorly when confronted with actual conditions—plans that appear optimal on paper may result in stockouts, excessive inventory, or unacceptable congestion levels in practice. The recognition of this challenge has motivated the development of optimization methodologies that explicitly account for data uncertainty. The literature on this topic is vast. For a general perspective on optimization under uncertainty, the reader is referred to [123], which reviews the interplay between optimization and uncertainty across operations research, to [62], which surveys robust optimization methods and their applications, and to [147], which focuses on adjustable (multi-stage) robust optimization. [88] provides a comprehensive review of robust optimization approaches in the context of operations management. A comprehensive treatment of stochastic programming—including two-stage and multi-stage formulations, scenario generation, and decomposition algorithms—can be found in [86]. Regarding supply chain applications specifically, [68] survey network design models that incorporate demand, supply, and cost uncertainty and [116] classify quantitative models for supply chain planning under uncertain conditions. Finally, for simulation-based approaches, [131] survey the use of discrete-event and agent-based simulation for analyzing supply chain dynamics, inventory policies, and coordination mechanisms.

Chapter 3

Methodological Framework

This chapter offers a self-contained overview of the mathematical methodologies underlying the contributions of this thesis. The presentation is not intended to be exhaustive; rather, it focuses on the specific frameworks and tools that are directly relevant to the developments in the subsequent chapters. For a more comprehensive treatment of the topics covered here, we refer the reader to the cited references and to the textbooks listed in the bibliography. The chapter is organized into four sections, each dedicated to a different methodological pillar. Section 3.1 reviews exact solution methods for mixed-integer programming—namely Branch-and-Bound, Branch-and-Cut, and Benders Decomposition—which constitute the algorithmic foundation of the solution approach proposed in Chapter 4. Section 3.2 introduces Bilevel Optimization, the modeling framework for hierarchical decision-making that plays a central role in Chapter 6. Section 3.3 presents Robust Optimization, the uncertainty modeling paradigm adopted in Chapter 4. Finally, Section 3.4 outlines Agent-Based Simulation, the methodological framework underlying Chapter 6.

3.1 Mixed-Integer Programming

Many supply chain and logistics optimization problems involve discrete decisions—whether to open a facility, which supplier to select, how to assign routes—and are naturally formulated as *mixed-integer linear programs* (MILPs). A generic MILP can be written as

$$\min_{x} \{c^\top x : Ax \leq b,\ x_j \in \mathbb{Z}\ \forall\, j \in \mathcal{I},\ x \geq 0\}, \tag{3.1}$$

where $c \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, and $\mathcal{I} \subseteq \{1, \ldots, n\}$ is the set of indices corresponding to integer-constrained variables. When $\mathcal{I} = \{1, \ldots, n\}$, the problem is a *pure integer program*; when $\mathcal{I} = \emptyset$, it reduces to a *linear program* (LP). In general, MILPs are NP-hard [146], and solving them exactly requires specialized enumeration techniques.

3.1.1 Branch-and-Bound

The *Branch-and-Bound* (B&B) algorithm, introduced by Land and Doig [82], is the foundational enumeration method for solving integer programs. The key idea is to systematically partition the feasible region into smaller subproblems (branching) while using bounds derived from LP relaxations to prune portions of the search space that provably cannot contain an optimal solution.

The *LP relaxation* of the MILP (3.1) is obtained by dropping the integrality requirements:

$$z_{\mathrm{LP}} = \min_{x} \{c^\top x : Ax \leq b,\ x \geq 0\}. \tag{3.2}$$

Since every feasible solution to (3.1) is also feasible for (3.2), the LP relaxation value z_{LP} provides a *lower bound* on the optimal MILP objective (for a minimization problem). If the LP relaxation solution x^* satisfies all integrality constraints, then x^* is optimal for the MILP; otherwise, a fractional variable x_j with $j \in \mathcal{I}$ is selected for branching.

Branching. Given a fractional solution component $x_j^* \notin \mathbb{Z}$, two child subproblems are created by imposing the disjunction

$$x_j \leq \lfloor x_j^* \rfloor \qquad \text{or} \qquad x_j \geq \lceil x_j^* \rceil.$$

This partitions the current region into two subsets, each of which is further explored recursively, forming an enumeration tree.

Bounding and pruning. At each node of the enumeration tree, the LP relaxation of the corresponding subproblem is solved. Let z^* denote the value of the best integer-feasible solution found so far (the *incumbent*). A node can be *pruned* (i.e., discarded without further branching) in three cases:

- *by bound*: the LP relaxation value at the node satisfies $z_{\mathrm{LP}} \geq z^*$, so no descendant can improve upon the incumbent;
- *by infeasibility*: the LP relaxation of the subproblem is infeasible;
- *by integrality*: the LP relaxation solution is integer-feasible, and z^* is updated if $z_{\mathrm{LP}} < z^*$.

The algorithm terminates when all nodes have been pruned, and the current incumbent is certified as an optimal solution. The effectiveness of B&B depends critically on the quality of the LP relaxation bounds and on the branching strategy; comprehensive discussions of these aspects can be found in the textbooks by Wolsey [146] and Nemhauser and Wolsey [108].

3.1.2 Cutting Planes and Branch-and-Cut

The *cutting plane method*, pioneered by Gomory [66], strengthens the LP relaxation by iteratively adding *valid inequalities* (cuts) that are satisfied by all integer-feasible solutions but violated by the current fractional LP solution. Given a fractional LP relaxation solution x^*, a valid inequality $\alpha^\top x \leq \beta$ is a *cutting plane* if $\alpha^\top x \leq \beta$ holds for all x feasible to (3.1) but $\alpha^\top x^* > \beta$. Adding such inequalities tightens the LP relaxation, potentially yielding integer-optimal solutions without branching.
Several families of general-purpose cuts have been developed, including *Gomory mixed-integer cuts* [67], *mixed-integer rounding* (MIR) cuts, and *lift-and-project cuts* [11]. In addition, problem-specific cuts exploit the combinatorial structure of particular formulations to produce stronger relaxations.
The *Branch-and-Cut* (B&C) algorithm [114] integrates cutting planes into the Branch-and-Bound framework: at each node of the enumeration tree, before branching, the algorithm attempts to tighten the LP relaxation by generating cutting planes. If the cuts suffice to make

the LP solution integer-feasible or to raise the lower bound above the incumbent, branching at that node is avoided. This combination significantly reduces the size of the enumeration tree in practice and is the algorithmic backbone of modern MILP solvers [146].

3.1.3 Benders Decomposition

Benders decomposition (BD) is a decomposition method introduced by Benders in [23] for solving large-scale mixed-integer programs whose variables naturally partition into two groups: a set of *complicating variables* (typically discrete) and a set of remaining variables (typically continuous) that, once the complicating variables are fixed, give rise to a much easier problem. The central idea consists in projecting out the continuous variables entirely, so that the original problem is reformulated in terms of the complicating variables alone by means of iteratively generated constraints known as *Benders cuts*. Since its introduction, BD has been extended well beyond its original scope and is now widely applied in stochastic programming, robust optimization, and network design, among other fields [120].
Consider a mixed-integer program with continuous variables y and discrete (complicating) variables x:

$$\min_{x,y} \{c^\top x + f^\top y : Ax + By \geq d,\ x \in X,\ y \geq 0\}, \tag{3.3}$$

where $x \in X \subseteq \mathbb{Z}^{n_1}$, $y \in \mathbb{R}^{n_2}$, $A \in \mathbb{R}^{m\times n_1}$, $B \in \mathbb{R}^{m\times n_2}$, $d \in \mathbb{R}^m$, $c \in \mathbb{R}^{n_1}$, and $f \in \mathbb{R}^{n_2}$. Crucially, the objective function is *separable* in x and y, and fixing x to any value $\hat{x} \in X$ reduces (3.3) to a linear program in y alone. This separability can be exploited to eliminate the continuous variables and to work exclusively with the discrete ones. Indeed, (3.3) can be rewritten as a nested optimization problem:

$$\min_{x\in X} \Big\{c^\top x + \underbrace{\min_{y\geq 0}\{f^\top y : By \geq d - Ax\}}_{Q(x)}\Big\}, \tag{3.4}$$

where the outer problem optimizes over the complicating variables x and the inner problem evaluates the best possible continuous decisions y for each candidate x. The function $Q(x)$, called the *value function* (or *recourse function*), encapsulates the cost contribution of the continuous variables. In the BD literature, the outer optimization is termed the *master problem* and the inner one the *Benders subproblem*.

Dual reformulation of the subproblem. For a given $\hat{x} \in X$, the Benders subproblem is the linear program

$$Q(\hat{x}) = \min_{y\geq 0} \{f^\top y : By \geq d - A\hat{x}\}. \tag{3.5}$$

Applying LP duality, we obtain the equivalent dual representation

$$Q(\hat{x}) = \max_{\pi\geq 0} \{\pi^\top (d - A\hat{x}) : B^\top \pi \leq f\}, \tag{3.6}$$

where $\pi \in \mathbb{R}^m$ is the vector of dual multipliers. A fact of central importance for BD is that the *dual feasible region*

$$\mathcal{F} = \{\pi \in \mathbb{R}^m : B^\top \pi \leq f,\ \pi \geq 0\} \tag{3.7}$$

is entirely determined by B and f, and thus *does not depend on the choice of* $\hat{x}$. The complicating variables enter only the dual objective $\pi^\top(d - A\hat{x})$, which means that varying $\hat{x}$ changes the objective coefficients but leaves the feasible polyhedron unchanged.

Optimality and feasibility cuts. Since $\mathcal{F}$ is a polyhedron, it is characterized by a finite set of *extreme points* $\{\pi^1, \ldots, \pi^P\}$ and *extreme rays* $\{r^1, \ldots, r^R\}$ (the latter spanning its recession cone). Two cases arise depending on the feasibility of the subproblem for a given $\hat{x}$. When the subproblem (3.5) is feasible and bounded, strong LP duality guarantees that the dual optimum is attained at some extreme point π^p of $\mathcal{F}$, so that

$$Q(\hat{x}) = \max_{p=1,\ldots,P} (\pi^p)^\top (d - A\hat{x}).$$

Introducing an auxiliary variable $\eta \geq Q(x)$, the requirement $\eta \geq (\pi^p)^\top (d - Ax)$ for every extreme point π^p defines the *Benders optimality cuts*, each providing a linear outer approximation of the value function $Q(x)$. If, on the other hand, the primal subproblem is infeasible for $\hat{x}$, then the dual is unbounded: by Farkas' lemma, there exists an extreme ray r^q of the recession cone of $\mathcal{F}$ such that $(r^q)^\top (d - A\hat{x}) > 0$. To exclude such infeasible first-stage decisions, the *Benders feasibility cut*

$$(r^q)^\top (d - Ax) \leq 0$$

is appended to the master problem.

Benders reformulation. Combining both types of cuts, the original problem (3.3) admits the reformulation

$$\begin{aligned} \min_{x,\eta} \quad & c^\top x + \eta \\ \text{s.t.} \quad & \eta \geq (\pi^p)^\top (d - Ax), \quad p = 1, \ldots, P, \quad \text{(optimality cuts)} \\ & (r^q)^\top (d - Ax) \leq 0, \quad q = 1, \ldots, R, \quad \text{(feasibility cuts)} \\ & x \in X, \; \eta \geq 0, \end{aligned} \tag{3.8}$$

expressed solely in terms of the complicating variables x and the auxiliary variable η. The continuous variables y have been completely projected out: their contribution to the objective is captured by η, and the coupling constraints are encoded implicitly through the Benders cuts.

Iterative solution procedure. Because the number of extreme points and extreme rays of $\mathcal{F}$ can be exponential, enumerating all Benders cuts a priori is impractical. Instead, the master problem (3.8) is solved via a *relaxation-and-cut* strategy that generates only the constraints that are actually needed. The procedure alternates between two phases:

1. Master phase. Solve a *relaxed* version of (3.8) containing only the cuts generated so far, yielding a candidate solution $(\hat{x}, \hat{\eta})$ and a *lower bound* $\text{LB} = c^\top \hat{x} + \hat{\eta}$ on the optimal value.

2. Subproblem phase. Fix $x = \hat{x}$ and solve the dual subproblem (3.6):

   - If the dual is *unbounded*, identify an extreme ray r^q and add the corresponding feasibility cut to the relaxed master problem.
   - If the dual is *bounded* with optimal solution π^p and value $Q(\hat{x})$, compute an *upper bound* $\text{UB} = \min\{\text{UB}, \; c^\top \hat{x} + Q(\hat{x})\}$ and add the corresponding optimality cut to the relaxed master problem.

At each iteration the lower bound is non-decreasing, since additional cuts can only tighten the relaxed master problem. The algorithm terminates when $\mathrm{UB} - \mathrm{LB} \leq \varepsilon$ for a user-specified tolerance $\varepsilon \geq 0$, at which point the best incumbent solution is certified as ε-optimal. Finite convergence is guaranteed because X is a finite set and the number of distinct extreme points and rays of $\mathcal{F}$ is finite [23].

Classical versus modern implementation. The iterative scheme described above is known as the *classical* or *multi-tree* variant of BD, because the master problem is re-solved from scratch at every iteration and a new Branch-and-Bound tree is constructed each time. While conceptually clean, this approach has a significant practical drawback: all the enumeration effort (branching decisions, bounds, incumbent solutions) accumulated while solving the master problem at one iteration is discarded before the next, leading to substantial redundant computation.

Modern implementations overcome this limitation through a *single-tree* strategy commonly referred to as *Branch-and-Benders-Cut* (B&BC) [120]. The idea is to embed the Benders cut generation directly inside a single Branch-and-Cut tree for the master problem. In practice, this is realized through the *lazy-constraint callback* mechanism offered by state-of-the-art MILP solvers such as CPLEX, Gurobi, and SCIP: whenever the Branch-and-Cut process encounters an integer-feasible node during its exploration, it triggers the callback, which solves the Benders subproblem for the candidate $\hat{x}$ and, if necessary, adds the resulting optimality or feasibility cut as a *lazy constraint*—i.e., a constraint that is only checked and possibly generated when a new incumbent is found. This way the solver continues exploring the same enumeration tree, extending it with the newly added cuts, rather than restarting from scratch. The Branch-and-Cut framework also ensures that all general-purpose cutting planes (Gomory cuts, MIR cuts, etc.) are generated alongside the Benders-specific cuts, combining the benefits of both methodologies in a unified solution process. The result is typically a substantial reduction in computation time, especially for large-scale instances, and makes it possible to leverage the full algorithmic arsenal of modern MILP solvers [120].

Beyond the choice of single- versus multi-tree implementation, several *acceleration techniques* can further improve convergence. Among the most common are: warm-starting the master problem with cuts derived from heuristic solutions; selecting *Pareto-optimal cuts* [91], i.e., choosing among alternative dual optima the one that yields the deepest cut; *disaggregating* the subproblem when it has a block-diagonal structure (e.g., independent scenarios in stochastic programming), so as to generate one cut per block rather than a single weaker aggregated cut; and *regularization*, which adds penalty terms or trust-region constraints to the master problem to prevent large oscillations in the first-stage solution between successive iterations.

As explained so far, the classical BD framework relies on LP duality to derive the cuts. [64] generalized this paradigm to settings where the subproblem is a *convex* (possibly nonlinear) optimization problem, replacing LP duality with nonlinear duality theory. The resulting *Generalized Benders Decomposition* (GB) is relevant whenever the subproblem features nonlinear constraints or objectives, as occurs in certain classes of robust and bilevel optimization problems.

3.2 Bilevel Optimization

Bilevel optimization models hierarchical decision-making settings involving two levels of optimization: a *leader* (upper-level decision-maker) and one or more *followers* (lower-level decision-makers). The leader chooses first, anticipating the optimal response of the followers; the followers then optimize their own objectives, taking the leader's decision as given. This nested structure arises naturally in network design, pricing, toll setting, and supply chain coordination, where different agents operate at different levels of authority. Bilevel problems are notoriously hard to solve. [70] have shown that even linear bilevel problems, i.e., those with continuous variables, linear objective functions, and linear constraints, are strongly NP-hard in general. The prevailing strategy for solving bilevel problems consists in transforming them into equivalent single-level formulations, which—after possible additional reformulations—can be solved with general-purpose optimization solvers. Here we give a brief overview of bilevel optimization, focusing on the general formulation in Section 3.2.1 and on the KKT-based reformulation in Section 3.2.2, which are relevant for the methodologies used in this thesis. For a comprehensive treatment of bilevel optimization, we refer the reader to [52] and [48], as well as to the surveys [77] and [55].

3.2.1 General Formulation

A bilevel problem is an optimization problem of the form

$$\begin{aligned} \min_{x\in X} \quad & F(x,y) \\ \text{s.t.} \quad & G(x,y) \geq 0, \\ & y \in S(x), \end{aligned} \tag{3.9}$$

where $x \in X \subseteq \mathbb{R}^{n_x}$ are the *leader's (upper-level) variables*, $y \in Y \subseteq \mathbb{R}^{n_y}$ are the *follower's (lower-level) variables*, $F\colon \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \to \mathbb{R}$ and $f\colon \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \to \mathbb{R}$ are the upper- and lower-level objective functions, and $G\colon \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \to \mathbb{R}^l$ and $g\colon \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \to \mathbb{R}^m$ define the upper- and lower-level constraints, respectively. The set $S(x)$ is the *rational reaction set* of the follower, i.e., the set of optimal solutions to the *lower-level problem* parameterized by the leader's decision x:

$$S(x) = \arg\min_{y\in Y} \{ f(x,y) : g(x,y) \geq 0 \}. \tag{3.10}$$

The *bilevel feasible set* is the set of all pairs (x,y) that satisfy the upper-level constraints and for which y is an optimal response to x for the lower level:

$$\{(x,y) \in X \times Y : G(x,y) \geq 0,\ y \in S(x)\}. \tag{3.11}$$

Note that, for a solution (x,y) to be bilevel-feasible, it is not enough that y satisfies the upper-level constraints $G(x,y) \geq 0$: it must also be an optimal response to x for the lower level, which is a much stronger requirement. Differently, the *shared constraint set*

$$\Omega \ := \ \{(x,y) \in X \times Y : G(x,y) \geq 0,\ g(x,y) \geq 0\},$$

collects all pairs (x,y) satisfying both levels' constraints without requiring y to be optimal for the lower level. Its projection onto the leader's space, $\Omega_x := \{x \in X : \exists\, y \text{ s.t. } (x,y) \in \Omega\}$, represents the set of leader decisions that are jointly feasible.

Optimistic and pessimistic approaches. The problem in (3.9) could happen to be ill-posed when the rational reaction set $S(x)$ is not a singleton for some x: different follower choices within $S(x)$ may yield different upper-level objective values, yet the formulation does not specify which one the follower selects. Two standard conventions resolve this ambiguity. In the *optimistic* approach, the leader assumes cooperative behavior on the part of the follower: the follower selects, among all his optimal responses, the one that minimizes the leader's objective F. The resulting *optimistic bilevel problem* is

$$\min_{x \in X,\, y} \{F(x,y) : G(x,y) \geq 0,\ y \in S(x)\}, \tag{3.12}$$

where both x and y are treated as joint decision variables subject to the lower-level optimality constraint on y. Under the *pessimistic* approach, by contrast, the follower is assumed to choose among his optimal responses the one that is *least* favorable to the leader, who must hedge accordingly [52]. Throughout this thesis we adopt the optimistic convention, which is standard in the bilevel optimization literature.

3.2.2 Single-Level Reformulation via KKT Conditions

The KKT reformulation, pioneered by [61], converts the bilevel problem into a single-level one by replacing the lower-level optimality constraint $y \in S(x)$ in (3.12) with a finite system of constraints that fully characterize lower-level optimality.
This approach requires that the lower-level problem admit a compact optimality certificate. To this end, we consider the case in which the lower-level problem is convex: we assume that the objective $y \mapsto f(x,y)$ is convex in y and that the constraint functions $y \mapsto g_i(x,y)$ are concave for all $i \in \{1,\dots,m\}$. We further assume that all follower variables are continuous ($Y = \mathbb{R}^{n_y}$) and that f and g are continuously differentiable in y. Finally, we assume that *Slater's constraint qualification* holds for every $x \in \Omega_x$.
Under all of the above assumptions, the Karush–Kuhn–Tucker (KKT) conditions are both necessary and sufficient for lower-level optimality, and $y \in S(x)$ if and only if there exists a multiplier vector $\lambda \in \mathbb{R}^m_{\geq 0}$ satisfying the KKT system

$$\begin{aligned} \nabla_y \mathcal{L}(y,\lambda;\, x) &= 0, \\ \lambda_i\, g_i(x,y) &= 0, \quad i \in \{1,\dots,m\}, \\ g(x,y) &\geq 0, \quad y \in Y, \quad \lambda \geq 0, \end{aligned} \tag{3.13}$$

where the Lagrangian gradient of the lower-level problem is

$$\nabla_y \mathcal{L}(y,\lambda;\, x) \;=\; \nabla_y f(x,y) - \sum_{i=1}^{m} \lambda_i\, \nabla_y g_i(x,y).$$

Replacing $y \in S(x)$ in (3.12) with the KKT conditions (3.13) yields the single-level KKT reformulation, a mathematical program with complementarity constraints (MPCC):

$$\begin{aligned} \min_{x,y,\lambda} \quad & F(x,y) \\ \text{s.t.} \quad & G(x,y) \geq 0, \\ & g(x,y) \geq 0, \\ & \nabla_y \mathcal{L}(y,\lambda;\, x) = 0, \\ & \lambda_i\, g_i(x,y) = 0, \quad i \in \{1,\dots,m\}, \\ & x \in X, \quad y \in Y, \quad \lambda \geq 0. \end{aligned} \tag{3.14}$$

The KKT reformulation (3.14) and the optimistic bilevel problem (3.12) are globally equivalent—every global solution of (3.12) can be extended to a global solution of (3.14) and vice versa—under the above convexity and Slater conditions [52, 77]. The main challenge in solving (3.14) is the nonlinearity of the complementarity constraints $\lambda_i\, g_i(x, y) = 0$; standard approaches to handle them include big-M linearization and SOS1 constraints. In the big-M approach, for each complementarity pair (a_i, b_i), a binary variable $u_i \in \{0, 1\}$ is introduced and the nonlinear condition is replaced by the linear disjunction

$$a_i \leq M(1 - u_i), \quad b_i \leq M\, u_i, \quad u_i \in \{0, 1\},$$

where $M > 0$ is a sufficiently large constant [61]. This transforms the MPCC (3.14) into a mixed-integer linear program (MILP) solvable by standard MILP solvers. A considerable drawback of this approach is the need for valid big-M values: constants that are too large weaken the LP relaxation, while constants that are too small may incorrectly cut off lower-level optimal solutions, as demonstrated in [117]. Complexity questions related to finding and validating correct big-M constants have been further addressed in [37] and [78]. An alternative that avoids the sensitivity of big-M constants is the use of special ordered sets of type 1 (SOS1). A SOS1 constraint imposed on a pair $\{a_i, b_i\}$ enforces that at most one of the two variables is nonzero, directly encoding the complementarity condition without requiring any explicit constant bound. Modern MILP solvers handle SOS1 constraints natively through dedicated branching rules, making them less prone to the numerical issues of the big-M approach.

3.3 Robust Optimization

Robust optimization [20, 24] provides a framework for decision-making under uncertainty that does not require knowledge of probability distributions. Instead, uncertain parameters are assumed to belong to a deterministic *uncertainty set*, and the goal is to find a solution that performs well under *all* possible realizations within that set. This worst-case (min-max) perspective makes robust optimization particularly suitable when distributional information is scarce or unreliable, or when the decision-maker is risk-averse. An alternative paradigm is *stochastic programming* [30, 126], which models uncertainty through explicit probability distributions and minimizes expected cost over all scenarios. While stochastic programming can yield less conservative solutions when reliable distributional information is available, it may be sensitive to distributional misspecification; moreover, the requirement of enumerating a finite set of scenarios often leads to very large deterministic equivalents. Robust optimization, by contrast, requires no probabilistic assumptions, optimizes against the worst-case realization in $\mathcal{U}$, and typically preserves the tractability class of the nominal problem. The choice between the two approaches depends on the quality of available data, the risk attitude of the decision-maker, and the computational requirements of the specific application; a detailed comparison can be found in [20] and [30].

3.3.1 Static Robust Optimization

Consider the *nominal* linear program, in which all problem data are known exactly:

$$\min_{x \geq 0} \{c^\top x : Ax \leq b\}, \tag{3.15}$$

where $c \in \mathbb{R}^n$ is the cost vector, $A \in \mathbb{R}^{m\times n}$ is the constraint matrix, and $b \in \mathbb{R}^m$ is the right-hand side. In practice, one or more of these data elements may not be known precisely at the time the decision must be made, owing to measurement errors, demand variability, or other sources of uncertainty. We capture this by parametrizing the uncertain data through a vector $\xi \in \mathbb{R}^L$, where L is the total number of uncertain scalar parameters (e.g., individual entries of A or b that are not known with certainty). The data thus become functions $c(\xi)$, $A(\xi)$, and $b(\xi)$ evaluated at the true—but unobservable—realization of ξ, and the uncertain version of (3.15) reads

$$\min_{x\geq 0} \{c(\xi)^\top x : A(\xi)\, x \leq b(\xi)\}. \tag{3.16}$$

A solution that is feasible and optimal for one realization of ξ may be infeasible or highly suboptimal for another. Robust optimization addresses this by seeking a single decision x that remains *simultaneously* feasible and cost-effective for *every* realization ξ in a prescribed *uncertainty set* $\mathcal{U} \subseteq \mathbb{R}^L$, which encodes all plausible parameter values. The resulting *static robust counterpart* of the nominal problem (3.15) is:

$$\min_{x\geq 0} \Big\{\max_{\xi\in\mathcal{U}} \; c(\xi)^\top x \;:\; A(\xi)\, x \leq b(\xi),\ \forall\, \xi \in \mathcal{U}\Big\}. \tag{3.17}$$

The feasibility constraint $A(\xi)\, x \leq b(\xi)$ must hold for *all* scenarios in $\mathcal{U}$ simultaneously, while the objective minimizes the worst-case cost over $\mathcal{U}$. This formulation is inherently conservative: it protects against the most adverse scenario, at the potential cost of suboptimality for typical realizations. The degree of conservatism is governed by the size of $\mathcal{U}$: as $\mathcal{U}$ shrinks to the singleton $\{\bar{\xi}\}$ of nominal parameter values, the robust counterpart recovers the nominal problem (3.15).
The earliest robust formulation, due to [129], considers column-wise interval uncertainty and yields a highly conservative solution. Subsequent developments by [22, 20] and by [26] introduced more refined uncertainty sets that allow the decision-maker to control the trade-off between robustness and conservatism.

3.3.2 Two-Stage (Adjustable) Robust Optimization

In many practical settings, not all decisions need to be made before the uncertainty is revealed. *Two-stage* (or *adjustable*) robust optimization (ARO) [21, 147] distinguishes between *first-stage* (here-and-now) decisions x, which must be taken before any uncertain parameter is observed, and *second-stage* (wait-and-see, or *recourse*) decisions $y(\xi)$, which can adapt to the realized uncertainty ξ. Compared to the fully static robust counterpart (3.17), ARO reduces conservatism by deferring part of the decision until information becomes available, while still providing worst-case feasibility and cost guarantees over the entire uncertainty set $\mathcal{U}$.
The generic two-stage robust problem takes the min-max-min form

$$\min_{x\in X} \max_{\xi\in\mathcal{U}} \min_{y\in Y(x,\xi)} \{c^\top x + f^\top y\}, \tag{3.18}$$

where $X \subseteq \mathbb{Z}^{n_1} \times \mathbb{R}^{n_x}$ is the first-stage feasible set and $Y(x, \xi) = \{y \geq 0 : By \geq d(\xi) - Ax\}$ defines the set of feasible recourse decisions for given x and ξ. The structure of (3.18) captures the sequential nature of the decision process: the leader (first stage) commits to x to minimize the worst-case total cost, and the recourse decision y is then chosen optimally for the realized scenario. A fundamental requirement for the well-posedness of (3.18) is that

$Y(x,\xi) \neq \emptyset$ for every $x \in X$ and every $\xi \in \mathcal{U}$, a condition known as *relatively complete recourse*. When this condition fails, a worst-case realization of ξ may render the second-stage problem infeasible for a given first-stage decision x, causing the objective in (3.18) to become $+\infty$. In practice, relatively complete recourse can be ensured by including additional constraints on the first-stage decisions (feasibility cuts), by augmenting the second-stage problem with penalty variables for constraint violation, or by appropriately restricting the uncertainty set $\mathcal{U}$.

Robust Optimization and Benders Decomposition. The Benders decomposition framework of Section 3.1.3 can be adapted to the min-max-min structure of (3.18), yielding an exact solution algorithm for ARO with continuous recourse and polyhedral $\mathcal{U}$. The master problem optimizes over the first-stage variables $x \in X$ and an auxiliary variable η representing the worst-case recourse cost:

$$\min_{x \in X,\, \eta} \{c^\top x + \eta\}.$$

For a candidate first-stage solution $\hat{x}$ at a node of the Branch-and-Bound tree, the *Benders subproblem* evaluates the worst-case recourse cost by solving the adversarial problem

$$\max_{\xi \in \mathcal{U}} \min_{y \geq 0} \{f^\top y : By \geq d(\xi) - A\hat{x}\}.$$

Dualizing the inner minimization over y converts this to a bilinear maximization over (ξ, π), where $\pi \geq 0$ are the dual multipliers associated with the recourse constraints:

$$\max_{\xi \in \mathcal{U},\, \pi \geq 0} \{\pi^\top (d(\xi) - A\hat{x}) : B^\top \pi \leq f\}.$$

For polyhedral $\mathcal{U}$, this bilinear program can be solved exactly by enumerating the extreme points of $\mathcal{U}$ or via linearization. The optimal pair (ξ^*, π^*) yields an *optimality cut*

$$\eta \geq (\pi^*)^\top (d(\xi^*) - Ax),$$

which is appended to the master problem as a lazy constraint. If the subproblem is unbounded (relatively complete recourse fails), a *feasibility cut* is generated analogously.
In the *Branch-and-Benders-Cut* (B&BC) implementation [120], the cut-generation callback is embedded directly inside a single Branch-and-Cut tree for the master problem (see Section 3.1.3): whenever an integer-feasible node is encountered, the adversarial subproblem is solved and, if the current $\hat{\eta}$ underestimates the true worst-case cost, the optimality cut is injected as a lazy constraint and the search continues in the same tree. This single-tree strategy avoids the redundant enumeration of the multi-tree variant and fully exploits the cutting-plane and branching machinery of modern MILP solvers [120].

Robust Optimization and Bilevel Optimization. The robust counterpart (3.17) can be interpreted as a *bilevel optimization problem* (Section 3.2): the outer minimization over x represents the decisions of a *leader* who seeks to minimize cost, while the inner maximization over $\xi \in \mathcal{U}$ models an adversarial *follower* who selects the worst-case parameter realization in response. This bilevel interpretation is structural and holds for any choice of $\mathcal{U}$. The two-stage robust problem (3.18) extends this structure to a *three-level* (min-max-min) hierarchy, in which a third decision-maker—the recourse optimizer—responds optimally to the adversary's choice of ξ.

3.4 Agent-Based Simulation

When analytical models become intractable or insufficiently detailed to capture the complexity of real-world transportation and logistics systems, *simulation* provides a complementary approach. In particular, *agent-based simulation* models individual entities—vehicles, travelers, shipments—as autonomous *agents* whose local decisions and interactions produce emergent system-level dynamics such as congestion patterns, queue formation, and delivery bottlenecks [72]. Unlike equation-based models, agent-based simulation can represent heterogeneous behaviors, stochastic events, and spatiotemporal dynamics at an arbitrary level of detail.

3.4.1 The MATSim Framework

MATSim (*Multi-Agent Transport Simulation*) [72] is an open-source, activity-based, agent-based transport simulation framework designed for large-scale scenarios involving millions of agents. In MATSim, each agent represents an individual traveler who executes a *daily activity plan*—a sequence of activities (e.g., home, work, shopping) interleaved with trips specifying departure times, modes, and routes.
The core of MATSim is an iterative *co-evolutionary* loop consisting of three modules that are repeated over multiple iterations until the system reaches a relaxed equilibrium:

1. **Mobility simulation (mobsim).** All agents execute their current plans simultaneously on a shared network infrastructure. The traffic simulation is mesoscopic: each link has a finite flow capacity and a free-speed travel time, and vehicles interact through queue-based dynamics. Congestion arises endogenously when link capacity is exceeded.

2. **Scoring.** After the simulation, each executed plan is evaluated by a *scoring function* that quantifies the utility experienced by the agent. The default scoring function in MATSim, based on the Charypar–Nagel utility model [41], assigns positive utility for performing activities (increasing logarithmically with duration) and negative utility for traveling (proportional to travel time and, optionally, monetary costs):

$$S_{\text{plan}} = \sum_q S_{\text{act},q} + \sum_q S_{\text{trav},q}, \tag{3.19}$$

   where $S_{\text{act},q}$ is the utility of performing activity q and $S_{\text{trav},q}$ is the (dis)utility of traveling to activity q.

3. **Replanning.** A fraction of agents is selected to modify their plans through *innovation* strategies (e.g., rerouting via shortest-path computation, adjusting departure times, changing transport modes), while the remaining agents select among their previously memorized plans based on a logit model. Over iterations, poorly performing plans are discarded from each agent's memory.

3.4.2 Convergence and Equilibrium

Through repeated iterations of the co-evolutionary loop, the agent population converges toward a *stochastic user equilibrium*: agents have explored alternative plans and no agent can significantly improve their score by unilaterally changing their plan. This process mimics a day-to-day learning dynamic in which travelers adjust their behavior based on past experience. Convergence is typically assessed by monitoring the average population score and its variability across iterations [72].

Chapter 4

Multiple suppliers purchase planning problem under lead time uncertainty: an exact Benders-like approach and a horizon-split heuristic [1]

4.1 Introduction

Procurement planning is one of the most critical functions in supply chain management. In many industrial, retail, and healthcare settings, a buyer must periodically replenish a single product from a set of available suppliers over a finite planning horizon, deciding both which suppliers to activate in each period and how much to order from each of them, so as to satisfy the demands while minimising the total cost of purchases, inventory holding, and backlogging. This problem is known as the Multiple Suppliers Purchase Planning (MSPP) problem [133]. When no capacity constraints are present, it can be viewed as an uncapacitated single-item lot-sizing problem with multiple suppliers (MSLSP). The multi-supplier structure is practically motivated: sourcing from several suppliers simultaneously provides flexibility and resilience against supply disruptions, enables price competition among vendors, and allows the buyer to exploit volume discounts or delivery guarantees that differ across suppliers.

The main difficulty in solving MSPP instances arising from real applications is the presence of data uncertainty. In particular, we focus on lead time uncertainty: the interval between placing an order and receiving it is not known in advance and may vary due to transportation delays, congestion, or suppliers' reliability issues. When lead times are uncertain, orders placed in different periods may arrive out of sequence, complicating the management of inventory levels. As a consequence, the robust version of the problem is harder, both theoretically and computationally, than its deterministic counterpart (see the discussion in Section 4.2.4).

[1] The work presented in this chapter has been developed in collaboration with Claudio Gentile (IASI-CNR) and Sara Mattia (IASI-CNR)

In this work we focus on the Multiple Suppliers Purchase Planning problem with backlogging allowed and no capacity constraints. Inventory backlog refers to customer orders that have been received but not yet fulfilled or shipped, representing pending revenue and allocated stock. Allowing backlogging provides the buyer with additional flexibility to manage supply disruptions, as it can choose to delay fulfilling some demand rather than incurring high costs from expedited shipping or holding excessive inventory. However, it also introduces additional complexity into the problem, as the buyer must balance the cost of backlogging against the cost of holding inventory and purchasing from suppliers.

We formulate the problem as a two-stage robust optimization problem [21]. In the two-stage framework, the supplier activation and the order quantity decisions constitute the first-stage variables, committed before the lead times are revealed, while the inventory holding and backlogging levels form the second-stage variables, which can be adjusted after observing the realized delays. We assume unrestricted recourse, i.e., second-stage decisions can be adapted with no restrictions; this maximizes solution flexibility at the cost of increased computational difficulty. The uncertainty is modelled through a structured uncertainty set that captures both the budgeted number of delayed deliveries and the temporal correlation of delays from the same supplier, providing a flexible and interpretable representation of lead time variability.

We tackle the 2rMSPP with two complementary approaches. First, we develop an exact branch-and-cut algorithm based on a Benders-like reformulation of the problem. Benders decomposition [23] is a popular technique, that have been successfully used for many applications, such as telecommunication network design [7, 97], blackout prevention [29] and personnel scheduling [99]. We also study the polyhedral structure of the uncertainty set of the single supplier, and derive valid inequalities for the separation subproblem, we show that these inequalities are facet-defining and we test their effectiveness in computational experiments. Second, for large instances, we propose a horizon-split heuristic that applies the exact two-stage model to the near-term portion of the planning horizon while approximating the cost of the remaining periods through the robust counterpart formulation. The idea behind this heuristic is that the exact two-stage model provides the most value in the near term, where the uncertainty has a more immediate impact on inventory levels and backlogging costs, while a single-stage approximation may suffice for the more distant future, where the effect of current decisions is less direct. Moreover, in real-life applications, the reliability of the input data, even for the known demand, typically decreases as the time horizon extends. Thus, focusing on a shorter exact window can be more aligned with the practical decision-making process, but still considering the long-term implications of current decisions through the robust approximation. By adjusting the size of the exact window, the decision-maker can trade off solution quality against computational time.

We conduct an extensive computational study on a large set of instances, showing that the exact algorithm effectively solves all instances for medium-sized problems, while the heuristic allows to tackle larger instances efficiently. We also perform a robustness analysis studying the sensitivity of the optimal solutions to the two main characteristics of the uncertainty set, i.e., the uncertainty is limited by a budget on the total number of delayed deliveries and orders from the same supplier in consecutive periods will exhibit similar delays. The analysis reveals interesting structural properties. As the delay budget increases, the model exhibits two distinct procurement regimes: a diversification regime, where more orders are placed from a wider set of suppliers to hedge against delays, and a consolidation regime, where the model reverts to fewer, larger orders to save on fixed activation costs. This transition is economically intuitive: when the uncertainty budget is large, diversification loses effectiveness because even orders spread across many suppliers can be simultaneously delayed, making cost efficiency a more effective strategy than risk mitigation. When the delay budget is small, the correlation parameter has a stronger influence on the solution cost, since tighter correlation constraints limit the possibility of highly disruptive delay patterns when only few orders can be delayed. As we relax the correlation constraints, the cost increase is evident, since there are more ways in which the uncertainty can manifest in a disruptive way. Both aspects exhibit strong saturation effects: beyond moderate thresholds the solution cost plateaus, implying that the model is robust to parameter misspecification, a desirable property for practical applications where precise estimation of the uncertainty budget is difficult.

The rest of the chapter is organised as follows. Section 4.1.1 surveys the relevant literature on multi-supplier lot sizing under lead time uncertainty. Section 4.2 formally states the problem, introduces the two-stage robust formulation, and discusses the differences between the deterministic and robust settings. Section 4.3 defines the structured uncertainty set, characterises its polyhedral properties, and derives facet-defining valid inequalities for the separation subproblem. Section 4.4 develops the Benders-like reformulation and presents the exact branch-and-cut algorithm. Section 4.5 introduces the horizon-split heuristic, designed to tackle large-scale instances efficiently. Section 4.6 describes the computational setup and instance design, while Section 4.7 reports the numerical results, including a robustness analysis and managerial insights. Finally, Section 4.8 draws conclusions.

4.1.1 Literature Review

This section surveys the literature relevant to the problem studied in this contribution, namely the robust optimization of multi-supplier purchase planning under lead time uncertainty. We initially focus on the classical single-item single-supplier dynamic lot-sizing problem and its deterministic extensions, which provide the foundational framework for our work. We then review the multi-supplier generalisation of the lot-sizing problem in the deterministic setting, highlighting the additional complexities introduced by supplier selection decisions. Finally, we survey existing models of lot-sizing under uncertainty, distinguishing between demand and yield uncertainty on one hand and lead time uncertainty on the other.

The dynamic lot-sizing problem (DLSP) asks for an optimal ordering plan over a finite horizon T so as to satisfy known, time-varying demands while minimising the total cost of ordering setups and inventory holding. The foundational model is due to [139], who presented a dynamic programming (DP) that solves the single-item uncapacitated DLSP with deterministic demand and no backlogging in $O(T^2)$ time. Their analysis established the most important structural property of the problem: there exists an optimal solution in which an order is placed in period t only if the inventory entering period t is zero. This property is the basis of every exact DP algorithm for the uncapacitated problem. The polyhedral structure of the uncapacitated DLSP has been studied extensively. [118] provide a comprehensive treatment of MILP formulations and valid inequalities for lot-sizing problems, including the classical facility-location and shortest-path reformulations that yield exact LP relaxations for the uncapacitated case. Comprehensive surveys of the single-item DLSP and its extensions are provided by [145], [35] and, more recently, by [34]. [35] classify the literature along several dimensions—cost structure, capacity constraints, backlogging, time windows—and identify classes that admit polynomial-time algorithms. [34] update and extend this survey, covering remanufacturing, sustainability objectives, and stochastic demand. Together, these surveys document the evolution of the field from the classical DP algorithm presented in [139] to modern branch-and-cut methods.

One of the most practically important extensions of the basic model is the one including backlogging, i.e., the case in which unmet demand can be satisfied in a later period at a penalty cost; the uncapacitated backlogging variant remains polynomial [139, 148], whereas the capacitated version is NP-hard in general [31]. [148] also extends the zero-inventory ordering structural property of the case without backlogging, to this more general setting. He states that there exists an optimal solution in which, for each period t, either the demand d_t is satisfied entirely by orders arriving in t or it is served entirely by inventory/backlogging.

Another important extension is the multi-supplier case (MSLSP, or equivalently, MSPP). When multiple suppliers are available, the procurement problem requires the joint determination of (i) which suppliers to activate in each period and (ii) how much to order from each selected supplier. This combined decision introduces binary variables representing supplier activation, coupling the lot-sizing decisions across suppliers in a non-trivial way. Surveys of supplier selection and order allocation in this framework are provided by [4] and [104]. [4] review models combining supplier selection with lot sizing, classifying contributions by the number of items and suppliers, the presence of capacity constraints, and the solution method employed. The single-item MSLSP was formulated as a MILP by [14], who established that the problem is NP-hard and proposed an exact enumeration algorithm and a heuristic. The NP-hardness result distinguishes the MSLSP fundamentally from its single-supplier counterpart: while the uncapacitated DLSP with one supplier is polynomial, adding supplier selection variables makes the problem hard even without capacity constraints. [14] also establish an important structural property for the single-item uncapacitated MSLSP: there exists an optimal solution in which, for each period t, no product is ordered from two (or more) suppliers in the same period t. This result implies that, when a single supplier dominates across every period of the time horizon, i.e., he his cheaper than every other supplier in every period, the MSLSP reduces to a single-supplier problem, solvable by dynamic programming. However, in many practical settings, no supplier is dominant across all periods and items, and the optimal solution may involve complex patterns of supplier selection and order allocation. [149] studied the polyhedral structure of the single-item uncapacitated MSLSP, derived necessary and sufficient conditions for facet-defining inequalities, and gave a full description of the convex hull for special cases. [39] extended this polyhedral analysis to the multi-item case, showing that the multi-item uncapacitated MSLSP is NP-hard and proposing a facility-location extended formulation, valid inequalities, and an effective MIP heuristic. [65] provide a more recent survey covering multi-criteria supplier selection, resilience, and sustainability objectives; the paper contextualises the lot-sizing-based models within the broader supplier evaluation literature.

Finally, the last extension of the problem we aim to review is the introduction of data uncertainty. As we already mentioned, data uncertainty is very common in this type of problems, but nevertheless, the literature on lot-sizing under uncertainty is relatively limited, especially when compared to the vast body of work on deterministic lot-sizing. Data uncertainty in optimization problems is usually addressed by simulation [12, 122], stochastic programming

[30, 76] and robust optimization [20, 80]. These techniques have been applied to many real-life problems, like healthcare applications [1], network design [96], personnel scheduling [99], container logistics [10] and to the MSPP problem itself [133]. In particular, we focus our attention on robust optimization approaches. These approaches are particularly relevant in the context of supply chain management, where historical data may be limited, and the underlying probability distributions of uncertain parameters may be difficult to estimate accurately. Robust optimization provides a framework for making decisions that are protected against the worst-case realizations of uncertainty, without relying on specific distributional assumptions. The uncertainty is modelled by a set of possible values that the uncertain parameters can take, i.e., the uncertainty set, and the goal is to find a solution that is feasible for all realizations of the uncertain parameters within this set and optimizes the worst-case objective value. The uncertainty set can be defined using a general framework (see, e.g., [26]) or a model depending on the specific application (see, e.g., [59, 99, 133]).

In the context of lot-sizing problems, uncertainty can arise from various sources, such as demand variability, supply disruptions, and lead time fluctuations. Most of the present literature focuses on demand uncertainty [27] or yield uncertainty [101, 45]. In the context of robust optimization, [27] developed a robust inventory management framework in which demand uncertainty is modelled by a budgeted uncertainty set. They derive closed-form base-stock policies that are optimal for the robust problem and show that these policies are substantially less conservative than those obtained under box uncertainty. The result provides a theoretical foundation for applying budgeted uncertainty sets to inventory problems. [45] studied a lot-sizing problem with joint supply and demand uncertainty under budgeted uncertainty sets. Their model includes both demand variability and supplier yield uncertainty and establishes structural properties of the robust optimal solution. [2] considered a robust single-item lot-sizing problem with uncertain demands and proposed a row-and-column generation algorithm. Yield uncertainty—in which a random fraction of a received order is usable—is closely related to the supply disruption literature. [101] studied an adaptive robust lot-sizing problem under yield uncertainty, proposing a two-stage robust model where the second-stage ordering decisions respond to realised yields. They established complexity results and developed an exact branch-and-cut algorithm. While yield uncertainty is structurally different from lead time uncertainty, the two-stage modelling paradigm and the use of budgeted uncertainty sets are directly relevant to the present work. A survey of models under random supply and demand is given by [65].

Lead time uncertainty, while highly relevant in multi-sourcing environments, has received comparatively less attention [104]. Lead time is the interval between the placement of an order and its arrival. In many supply chain contexts, lead times exhibit variability due to congestion, transportation delays, and suppliers reliability issues. When lead times are uncertain, orders placed in different periods may arrive out of sequence, and the inventory balance constraints must account for all possible arrival patterns. Classical models treat lead time as a fixed and known parameter. For the single-supplier case, the extension to stochastic lead times was studied by [17], who considered a single-supplier replenishment planning problem with stochastic lead times drawn from a known probability distribution. They established structural properties of the optimal policy and provided analytical results for specific distribution families. In the robust optimization paradigm, [134] considered a single-item, single-supplier inventory problem with joint demand and lead time uncertainty modelled by a budgeted uncertainty set.

They proposed a Benders decomposition algorithm and demonstrated that the robust optimal policy can differ substantially from the stochastic optimal policy when the true distribution is unknown. The model of [134] does not include fixed ordering costs. [71] studied a discrete-scenario lead time uncertainty model for the single-item single-supplier lot-sizing problem and established NP-hardness. An important structural result is that the zero-inventory ordering property presented in the deterministic case [139, 148], i.e., that there exists an optimal solution in which for each period t, either the demand d_t is satisfied entirely by orders arriving in t or it is served entirely by inventory/backlogging, does not hold under lead time uncertainty: when lead times are uncertain, it may be optimal to maintain positive inventory even at the time of order placement, since early orders provide a buffer against late arrivals. This finding sharply distinguishes the robust lot-sizing problem from its deterministic counterpart and has implications for both solution algorithms and managerial policy. The combination of supplier selection decisions with lead time uncertainty is the setting most directly relevant to the present work. Surprisingly, this combination has received limited attention in the literature. Recently, [18] considered the multi-supplier settings under stochastic lead times, characterising optimal replenishment policies under distributional assumptions. The only work—to the best of our knowledge—that addresses multi-supplier lot sizing with lead time uncertainty modelled by polyhedral uncertainty sets is [132]. Their model includes per-period fixed activation costs for each supplier, unit purchase costs, inventory holding and backlogging costs. The objective is to minimise the worst-case total cost over all realisations of lead times consistent with the uncertainty set. The authors propose a row-and-column generation algorithm that iteratively adds adversarial scenarios to a master problem, and complement this with a fix-and-optimize matheuristic for large instances. We share the same problem but we model lead time uncertainty using a different structured uncertainty set and propose a different solution approach, based on a Benders-like reformulation and a horizon-split heuristic.

4.2 Problem statement

We study the Multiple Suppliers Purchase Planning (MSPP) problem under lead time uncertainty, which consists in determining a purchasing plan for a given time horizon T and a set of suppliers S, to satisfy the demand of each time period at minimum cost, while accounting for the possibility that some orders may be delayed. The cost of the purchasing plan includes the fixed and variable costs of the orders, as well as the inventory holding and backlogging costs. Here we present the two-stage model for the MSPP problem with and without lead time uncertainty. We will refer to the problem without uncertainty as to the *deterministic* problem, while we call *robust* problem the one with lead time uncertainty.

4.2.1 The notation

In what follows, we will use the parameters below.

T is a discrete time horizon and, for each time period $t \in T$:

- $d_t \geq 0$ is the demand of period t;
- $c_t^+ \geq 0$ the unit inventory cost for period t;
- $c_t^- \geq 0$ the unit backlogging cost for period t.

S is the set of the suppliers and, for each supplier $s \in S$:

$o^s \geq 0$ is the fixed cost of s for managing the orders, which is the same, independently of the time period where the order is placed and of the ordered amount;

$p_t^s \geq 0$ is the unit variable cost (cost depending on the ordered amount) of s at time t.

γ_{ht}^s is a binary parameter that takes value 1 if an order placed from supplier $s \in S$ in time period $h \in T$ arrives in time period $t \in T_h^s$ and 0 otherwise.

Parameters γ control the actual arrival time of the orders. If γ is fixed, there is no lead time uncertainty and the problem is deterministic (see Section 4.2.2), i.e., $|T_h^s| = 1$. We will later assume γ to be uncertain according to the uncertainty model in Section 4.3, with $T_h^s = \{t \in T : h + \ell^s \leq t \leq h + \mathcal{L}^s\}$, where:

$\ell^s \geq 0$ is the minimum lead time (time periods needed to deliver the order) of supplier s;

$\mathcal{L}^s \geq 0$ is the maximum lead time of supplier s.

We also define $\bar{T}^s = \{t \in T : t + \mathcal{L}^s \leq |T|\}$ as the set of time slots where it is possible to place an order from supplier s that can arrive within the time horizon T. The formulation of the deterministic model is reported in Section 4.2.2 and the one of the robust problem is illustrated in Section 4.2.3. We will use the following decision variables. Let:

y_t^s be a binary variable that takes value 1 if an order is placed in time period $t \in T$ from supplier $s \in S$ and 0 otherwise;

q_t^s be an integer variable representing the amount purchased from supplier $s \in S$ at time period $t \in T$;

I_t be a continuous variable corresponding to the inventory level at the end of period $t \in T$;

B_t be a continuous variable indicating the backlogging level at the end of period $t \in T$.

In the two-stage robust model, variables y and q are the first stage variables, whose value must be the same independently of the realization of the uncertain parameters γ, whereas variables I_t, B_t are the second stage ones, whose values can be changed according to the actual value of parameters γ.

4.2.2 The deterministic formulation

Assume that γ is a fixed (not uncertain) parameter. The deterministic MSPP problem can be formulated as below.

$$\text{dMSPP} \quad \min \sum_{t\in T}\sum_{s\in S}(o^s y_t^s + p_t^s q_t^s) + \sum_{t\in T}(c_t^+ I_t + c_t^- B_t) \tag{4.1}$$

$$\text{s.t.} \quad Ay + Dq \geq b \tag{4.2}$$

$$q_t^s \leq M y_t^s, \qquad t \in T, s \in S \tag{4.3}$$

$$I_t - B_t = \sum_{s\in S}\sum_{h=1}^{t}\sum_{k=h}^{t}\gamma_{hk}^s q_h^s - \sum_{h=1}^{t} d_h, \qquad t \in T \tag{4.4}$$

$$y \in \{0,1\}^{|T|\times|S|},\ q \in \mathbb{R}_+^{|T|\times|S|},\ I, B \in \mathbb{R}_+^{|T|}$$

The objective function (4.1) minimizes the sum of the purchase and the inventory/backlogging costs. Constraints (4.2) may be present or not, depending on the specific problem, and represent possible commercial agreements between the purchaser and the suppliers and other restrictions depending on the purchaser or on a supplier, such as: minimum and/or maximum amounts for the orders from a given supplier; minimum and/or maximum number of orders that a supplier can receive; minimum and/or maximum number of total orders that the purchaser can place; periods where orders cannot be placed and/or received, depending on the suppliers or the purchaser; maximum budget that can be used for placing orders from a given supplier; minimum and/or maximum number of suppliers that must be selected. Constraints (4.3) forbid to order any amount from supplier s at time t, if s has not been selected as an available suppliers at time t. Amount M can be set equal to $\sum_{t\in T} d_t$. Constraints (4.4) calculate backlogging and inventory levels of each time period.
A common assumption in the literature is that inventory and backlogging are empty at the beginning and at the end of the time horizon, which implies that all demands must be served within the time horizon T.

Assumption 1. *Inventory and backlogging are empty at the beginning and at the end of the time horizon.*

$$I_0 = B_0 = I_{|T|} = B_{|T|} = 0$$

This implies

$$q_h^s = 0 \quad \text{if } \gamma_{ht}^s = 0 \text{ for } h \le t \le T \tag{4.5}$$

that means that no order is placed in time period h from suppliers that cannot deliver it within the end of the considered time horizon. Assumption 1 also implies that the quantity ordered in the time horizon equals the total demand:

$$\sum_{t\in T}\sum_{s\in S} q_t^s = \sum_{t\in T} d_t \tag{4.6}$$

inequality (4.6) is valid and can be added to the model. Also note that the model contains the implicit assumption that only one order is placed from each supplier at each time period.

4.2.3 The robust model

We now consider the problem with uncertainty. The parameters γ are uncertain and take values within a given uncertainty set Γ. We make the following assumption:

Assumption 2. *The uncertainty set* Γ *is non-empty and bounded.*

The robust problem takes the form of a multilevel optimization program. In multilevel programming, each level corresponds to a distinct optimization subproblem whose feasible region or objective depends on the decisions made at the levels above it; the different levels are nested as follows. The outermost minimization selects the first-stage variables y and q—the purchasing plan—before the uncertainty is revealed. The intermediate maximization represents an *adversary* that, given the first-stage solution, selects the worst-case realization $\gamma \in \Gamma$ of the uncertain lead times. The innermost minimization then computes the optimal recourse, i.e., the values of the inventory I and backlogging B, for the given first-stage solution and the adversarially chosen scenario. Iand Bare the second-stage variables.

A critical modelling choice concerns the recourse policy adopted for the second-stage variables. In a *static* recourse policy, I and B would be fixed independently of the realization γ; however, since the balance constraints (4.4) are equality constraints, their right-hand side varies with γ, and a single fixed assignment cannot remain feasible across all scenarios simultaneously [21]. The second-stage variables must therefore be adapted to each realization, yielding an *adjustable* (or *dynamic*) recourse policy.
The resulting model is the following two-stage robust optimization problem, which we refer to as 2rMSPP.

$$\text{2rMSPP} \;\; \min \sum_{t\in T}\sum_{s\in S}(o_s y_t^s + p_s q_t^s) + \lambda \tag{4.7a}$$

$$\text{s.t.} \quad Ay + Dq \geq b \tag{4.7b}$$

$$\sum_{t\in T}\sum_{s\in S} q_t^s = \sum_{t\in T} d_t \tag{4.7c}$$

$$q_t^s \leq M y_t^s, \qquad s \in S, h \in T \tag{4.7d}$$

$$q_h^s = 0, \qquad s \in S, h \in T, h + \mathcal{L}^s > |T| \tag{4.7e}$$

$$y \in \{0,1\}^{|T|\times|S|}, q \in \mathbb{R}_+^{|T|\times|S|}, \lambda \in \mathbb{R}_+$$

$$\lambda = \max_{\gamma\in\Gamma} \min \sum_{t\in T}(c_t^+ I_t + c_t^- B_t) \tag{4.7f}$$

$$I_t - B_t = \sum_{s\in S}\sum_{h=1}^{t}\sum_{k=h}^{t} \gamma_{hk}^s q_h^s - \sum_{h=1}^{t} d_h, \qquad t \in T \tag{4.7g}$$

$$I, B \in \mathbb{R}_+^{|T|}$$

The objective function (4.7a) minimizes the fixed and variable costs of the orders plus the worst case costs of the inventory and the backlogging, whose amount is represented by auxiliary variable λ. These worst case costs depend on the realization of the uncertain parameters γ and on the first stage choices y, q that have been made. Value λ is computed by the inner maxmin problem, where in the max the worst case realization of γ is chosen and in the inner min the associated cost is computed. In the inner min, amounts y, q and γ are given and can be regarded as constant parameters, while the variables are only I and B. Observe that the inner min is always feasible and bounded for q and γ by (4.7e) and Assumption 2. Equalities (4.7e) are the generalization to the uncertain framework of the condition expressed by equality (4.5) and impose that we cannot place an order from suppliers s at time h if this order is not guaranteed to arrive within the end of the time horizon. We need to impose such a restriction to guarantee that all demands within the time horizon T can be potentially satisfied. In the literature, different ways have been used to model this requirement:

(a) Replace $B_{|T|} = 0$ with the relaxed condition $B_{|T|} \geq 0$.
This allows demand that cannot be fulfilled within the horizon to be recovered by orders arriving after period $|T|$, penalising unmet demand at the backlogging cost $c_{|T|}^-$.

(b) Order surplus quantities to ensure that sufficient supply arrives within the time horizon.
This requires relaxing (4.6) to $\sum_{t\in T}\sum_{s\in S} q_t^s \geq \sum_{t\in T} d_t$ (see [133]); however, it may yield

overly conservative—and therefore costly—purchasing plans, as the model may order substantially more than total demand to hedge against worst-case delays.

(c) Replace the maximum delay $\mathcal{L}^s$ with the period-dependent bound $\mathcal{L}^s_h = \min\{\mathcal{L}^s, |T| - h\}$, where h is the period in which the order is placed.
Although this ensures feasibility by construction, it distorts the uncertainty set: orders placed in later periods face a tighter delay bound, making them artificially appear less uncertain than earlier ones and introducing an asymmetry across time periods.

Some of these options have been compared in [133], where the results show that (c) leads to models that can be solved faster, with no significant cost increase. Hence, we decided to adopt this option and to model the requirement that all demands must be potentially satisfied by imposing the explicit constraints (4.7e) in the formulation. Nevertheless, the model we propose can be easily adapted to the other options.
The presented model is a two-stage robust optimization problem with adjustable recourse, which is known to be computationally challenging in general [21]. In particular, the robust MSPP is NP-hard, as it includes the robust problem with a single supplier, which is NP-hard already [71]. In Section 4.4.1 we show how to reformulate 2rMSPP using a Benders-like approach.

4.2.4 Deterministic vs robust

As discussed in Section 4.1.1, the deterministic problem has been extensively studied in the literature and admits two structural properties that are key to its efficient solution. First, for each time period t, any optimal solution satisfies the demand d_t exclusively from one of three sources: orders arriving in t, existing inventory, or backlogging; this property underlies the dynamic programming algorithms of [139, 148]. Second, in the multi-supplier setting, there always exists an optimal solution in which, for each period t, no product is ordered from two (or more) suppliers in the same period t [14]. As already discussed in Section 4.1.1, this result implies that, when a single supplier dominates across every period of the time horizon, i.e., he his cheaper than every other supplier in every period, the MSPP reduces to a single-supplier problem, solvable by dynamic programming. However, in many practical settings, no supplier is dominant across all periods and items, and the optimal solution may involve complex patterns of supplier selection and order allocation.
Neither property carries over to the robust setting.
We start showing in Example 1 that, in case of uncertainty, we may need to order from multiple suppliers in the same time period, even when there is one supplier that has strictly smaller costs than the others. This means that, in the robust case, we can not use the definiton of "dominant/dominated" supplier and this provides also a counterexample to what is stated in see Definition 3.1 in [133].

Example 1. *Consider a problem with two suppliers (*1 *and* 2*),* $T = \{1, 2\}$ *and demands* $d_1 < d_2$*. Assume that the backlogging costs* c_1^- *are much larger than the production/inventory costs. All other costs are negligible. Suppose that* Γ *is the following set: at most one order may be late and it can arrive either in the same period when the order is placed or in the next one. Because of constraints (4.7e), no order can be placed in* $t = 2$*. Hence, we are forced to order in* $t = 1$ *either from one supplier (and, then, from the cheapest one) or from both suppliers. The orders placed in time period* $t = 1$ *may arrive either in* $t = 1$ *or in* $t = 2$*. Since the largest* *costs are associated with the backlogging, the smaller the backlogging the better the solution. If we order only from one supplier, the worst case realization of the uncertainty is when the unique order is delayed and arrives in $t = 2$, so that we have to pay the backlogging amount $\lambda = c_1^- d_1$. If we order from both suppliers, the worst case realization is when one of the two orders is delayed and arrives in $t = 2$, but no backlogging is possible, because at most one order can be late and, hence, demand d_1 can be entirely served by orders arriving in $t = 1$. If we order from both suppliers amounts $q_1^1 \geq d_1$, $q_1^2 \geq d_1$ and $q_1^1 + q_1^2 = d_1 + d_2$, no backlogging is possible: at most one order can be late and demand d_1 can be entirely served by orders arriving in $t = 1$.*

Hence, under uncertain conditions, ordering from multiple suppliers in the same time period can be a way to hedge against the risk of delays, even when one supplier is cheaper than the others.
Note that the worst case realization for a given q is not necessarily the one where some orders are delayed, but can also be the one where every order is on time, as discussed in Example 2.

Example 2. *Consider again the problem in Example 1. Let q be the solution where, in time period $t = 1$, we order amount d_1 from supplier 1 and amount d_2 from supplier 2. In this case, the worst case realization is when both orders arrive on time (and, hence, in $t = 1$), so that we have to pay amount $\lambda = c_1^+ d_2$ to serve the demand of time period $t = 2$ by inventory. In fact: if the order from supplier 2 is late, we pay no inventory/backlogging costs; if the order from supplier 1 is late, we pay inventory cost $\lambda = c_1^+(d_2 - d_1)$.*

Finally, we show in Example 3 that neither the single-source property ([139], [148]) holds in the robust case, i.e., that it may be optimal to satisfy demand d_t in period t by a combination of orders arriving in t, inventory, and backlogging. A similar example is also provided in [71].

Example 3. *Consider again the problem in Example 1, but assume that $d_1 > d_2$. As we have already discussed, ordering from a single supplier at time $t = 1$ would imply to serve d_1 entirely by backlogging, paying cost $\lambda = c_1^- d_1$, because the worst case realization is when the unique order is late. If we order from both suppliers, differently to what done in Example 1, here we can no longer order at least d_1 from each supplier, since $2d_1 > d_1 + d_2$, which is the maximum total amount that can be ordered. Hence, d_1 can never be entirely served by orders arriving in $t = 1$, as in any worst case realization the largest order would be delayed to increase the backlogging cost. Hence, irrespective of the amounts that we order from the suppliers, any solution having $q_1^1 > 0$ and $q_1^2 > 0$ will be cheaper than any solution where a single order is placed and, in all those solutions, d_1 will be served partially by the order that is not delayed and arrives in $t = 1$ and partially by backlogging (the order that is delayed and arrives in $t = 2$). In follows that any optimal solution also has this property. The optimal solution is to order $(d_1 + d_2)/2$ from both suppliers and to pay backlogging cost $\lambda = c_1^-(d_1 - d_2)/2$.*

4.3 The uncertainty set

Robust optimization models are defined by the choice of the uncertainty set, which captures the decision-maker's assumptions on how the uncertain parameters may vary and interact. The models and algorithms described in the previous sections are independent of the specific uncertainty set, provided that it is non-empty and bounded (Assumption 2). Here we describe the set we are interested in.

Definition 1. *Set* Γ *is defined as follows.*

$$\Gamma = \Bigg\{ \gamma \in \{0,1\}^{|S|\times|T|\times|T|} : \tag{4.8}$$

$$\gamma^s_{ht} = 0, \qquad s \in S,\ h,t \in T : h + \mathcal{L}^s > |T| \tag{4.9}$$

$$\sum_{t=h+\ell^s}^{h+\mathcal{L}^s} \gamma^s_{ht} = 1, \qquad s \in S,\ h \in \bar{T} \tag{4.10}$$

$$\left| \sum_{t=h+\ell^s}^{h+\mathcal{L}^s} t\gamma^s_{ht} - \sum_{t=h-1+\ell^s}^{h-1+\mathcal{L}^s} t\gamma^s_{h-1t} \right| \le K, \qquad s \in S,\ h \in \bar{T} \setminus \{1\} \tag{4.11}$$

$$\sum_{s\in S} \sum_{h \in T} \sum_{t=h+\ell^s+1}^{h+\mathcal{L}^s} \gamma^s_{ht} \le H \Bigg\} \tag{4.12}$$

Inequality (4.12) imposes a maximum of H delayed orders over the whole time horizon, where an order placed at period h by supplier s is considered delayed if it does not arrive at the minimum lead time $h + \ell^s$. [133] use similar constraint, which corresponds to assume that the uncertainty is somehow bounded, as in the model by [26]. Constraints (4.9) set to zero γ^s_{ht} values corresponding to infeasible orders (i.e., orders placed at time h that arrive outside the interval $\{h+\ell^s, \dots, h+\mathcal{L}^s\}$ or orders placed at time h that are not allowed because they would not arrive within the planning horizon). Constraints (4.10) ensure that any order placed in $h \in \bar{T}^s$ arrives in the interval $\{h + \ell^s, \dots, h + \mathcal{L}^s\}$, and that it arrives only once. Inequalities (4.11) model correlation among time periods, imposing that orders placed from the same supplier at consecutive time slots must have similar delays. [99] use similar constraints to model correlation among the demands for a personnel scheduling problem. Note that, differently from the sets defined in [133], set Γ contains integer variables, modeling the restriction that the orders cannot be split and the whole amount of the order must be delivered in a unique occurrence. This avoids to split the delivery into multiple arrivals of small amounts, causing problems to the production process, which may not proceed until the whole quantity arrives.

4.3.1 The single supplier set Γ^s

Let us consider the set Γ^s, where only supplier s is considered and constraint (4.12) is removed.

$$\Gamma^s = \Bigg\{ \gamma^s \in \{0,1\}^{|T|\times|T|} :\ (4.9), (4.10), (4.11) \Bigg\} \tag{4.13}$$

The set $\Gamma \subseteq \prod_{s\in S} \Gamma^s$ and thus, any inequality that is valid for Γ^s is valid for Γ. We aim to analyze the polyhedral properties of Γ^s and then to derive valid inequalities for Γ. For the sake of simplicity, since in the following anlysis we focus on a single supplier s, we omit the superscript s. We define $P = conv(\Gamma^s)$.

4.3.2 Valid Inequalities

Theorem 1. *The inequalities (4.14) and (4.15) are valid for* P*.*

$$\sum_{t=h+\ell+K+w}^{h+\mathcal{L}} \gamma_{ht} \le \sum_{t=h+\ell+w}^{h-1+\mathcal{L}} \gamma_{h-1t} \qquad w \in \{0,\dots,W\},\ h \in \bar{T} \setminus \{1\} \tag{4.14}$$

$$\sum_{t=h+\ell+K+2+i}^{h+\mathcal{L}} \gamma_{ht} \leq \sum_{t=h+\ell+2+i}^{h+1+\mathcal{L}} \gamma_{h+1t} \quad i \in \{0,\ldots,I\},\ h, h{+}1 \in \bar{T} \tag{4.15}$$

$W = (h+\mathcal{L}) - (h+l+K) = \mathcal{L} - \ell - K$ and w is an integer. $I = (h+\mathcal{L}) - (h+l+K+2) = \mathcal{L} - \ell - K - 2$ and i is an integer.

Proof. We prove validity for (4.14); the proof for (4.15) is analogous. By contradiction, suppose there exists $\overline{\gamma} \in \Gamma^s$ violating (4.14) for some $\overline{h}$ and $\overline{w}$. Both sides of (4.14) sum $\overline{\gamma}$ variables over a subset of the arrival window of the respective period. By (4.10), exactly one variable in each full arrival window equals 1 and the rest are zero, so each side is either 0 or 1. For the inequality to be violated, the left-hand side must equal 1 and the right-hand side must equal 0. The left-hand side being 1 means that the order at time $\overline{h}$ arrives at some time $t_1 \geq \overline{h} + \ell + K + \overline{w}$. The right-hand side being 0 means that the order at time $\overline{h} - 1$ arrives outside the range $\{\overline{h} + \ell + \overline{w}, \ldots, \overline{h} - 1 + \mathcal{L}\}$, contradicting (4.11). □

4.3.3 The description of conv(Γ^s)

We prove that P is completely described by inequalities (4.14), (4.15), (4.10), (4.9), and non-negativity (Theorem 6). The proof proceeds in three steps.

(i) **Dimension** (Section 4.3.3). We prove that $\dim(P) = NR$. This result is a prerequisite for the next step: by definition, an inequality defines a *facet* of P if and only if the face it induces has dimension $\dim(P) - 1$, so the dimension of P must be known beforehand.

(ii) **Facet-defining inequalities** (Section 4.3.3). We prove that (4.14), (4.15), and non-negativity inequalities are facet-defining for P. This establishes that the description is *irredundant*: each inequality defines a maximal proper face of P and cannot be removed from a minimal description.

(iii) **Completeness** (Section 4.3.3). We prove that the above inequalities are also *sufficient* to describe P, i.e., no additional inequalities are needed. This is achieved in two parts:

  (a) We prove that the constraint matrix $\mathcal{M}$ of (4.14), (4.15), (4.10) is Totally Unimodular (TU).

  (b) We prove that (4.14) and (4.15) together imply (4.11), so no feasible point of Γ^s is lost by dropping (4.11).

Combining the results from the three steps above yields Theorem 6.

The dimension of P

We start by proving the dimension of P in the following theorem.

Theorem 2. *Let $N = |\bar{T}|$ and $R = \mathcal{L} - \ell$, then $dim(P) = NR$.*

Proof. Rename the second index of γ so that t ranges from 0 to $\mathcal{L} - \ell$. The variables are γ_{ht} with $h \in \{1, \ldots, N\}$ and $t \in \{0, \ldots, R\}$.
Since there are $T - \mathcal{L}$ equality constraints, then $dim(P) \leq NR$.
We now need to prove that there are no other equality constraints satisfied by every point in Γ^s that are linearly independent from (4.10).

We define two families of points in Γ^s:

1. p_τ for $\tau \in \{0, \dots, R\}$: $\gamma_{h\tau} = 1$ for every h, and $\gamma_{ht} = 0$ for $t \neq \tau$,

2. $q_{j\tau}$ for $j \in \{1, \dots, N\}$ and $\tau \in \{0, \dots, R{-}1\}$: $\gamma_{j,\tau+1} = 1$, $\gamma_{h\tau} = 1$ for $h \neq j$, and all other $\gamma_{ht} = 0$.

Points p_τ trivially satisfy the delay constraint since all orders arrive at the same time, and thus with the same delay; points $q_{j\tau}$ have a maximum delay of $1 \leq K$ between consecutive slots, and thus satisfy the delay constraint as well.

Let $\alpha\gamma = \alpha_0$ be any equality satisfied by every point in Γ^s.
The points p_τ and $q_{j\tau}$ differ only in period j: in p_τ an order placed in j arrives at τ, in $q_{j\tau}$ it arrives at $\tau + 1$. All the other periods are identical.
Since both points satisfy $\alpha\gamma = \alpha_0$, subtracting yields $\alpha_{j\tau} - \alpha_{j,\tau+1} = 0$, and thus $\alpha_{j\tau} = \alpha_{j,\tau+1}$. Since the argument holds for every period $j \in \{1, \dots, N\}$ and every $\tau \in \{0, \dots, R{-}1\}$, we obtain that, for each period h, all coefficients α_{ht} are equal:

$$\alpha_{h0} = \alpha_{h1} = \cdots = \alpha_{hR}.$$

Let $c_h = \alpha_{h0}$ denote this common value. Then $\alpha\gamma = \sum_{h=1}^{N} c_h \sum_{t=0}^{R} \gamma_{ht}$. Since every point in Γ^s satisfies $\sum_{t=0}^{R} \gamma_{ht} = 1$ for each h, the expression above evaluates to $\sum_{h=1}^{N} c_h = \alpha_0$ for every feasible point.
Hence $\alpha\gamma = \alpha_0$ is just a linear combination of the N equality constraints (4.10), and no valid equality can be linearly independent from them. Therefore $\dim(P) = NR$. □

Facet-defining inequalities

We now aim to prove that inequalities (4.14), (4.15) and non-negativity inequalities are facet defining for P. In the following, we report the proof for (4.14); the proof for (4.15) and non-negativity inequalities is analogous and thus omitted.

Theorem 3. *Inequalities* (4.14) *are facet defining for* P.

Proof. Let us suppose that an inequality (4.14) for a pair h and w is not facet defining for P, then there exists an inequality $\alpha\gamma \leq \alpha_0$, which defines a facet containing the solutions that are tight for that inequality (4.14).
We consider two types of tight solutions for (4.14):

- the set $\mathcal{S}^0$ of solutions such that both sides evaluate zero,

- the set $\mathcal{S}^1$ of solutions such that both sides evaluate one.

Let $T(h-1) = \{h-1+\ell, \dots, h-1+\mathcal{L}\}$ and $T(h) = \{h+\ell, \dots, h+\mathcal{L}\}$. The solutions in $\gamma \in \mathcal{S}^0$ are such that $\gamma_{h-1\,\tau_{h-1}} = 1$ with $\tau_{h-1} \in \{h-1+\ell, \dots, h+\ell+w-1\}$, $\gamma_{h\tau_h} = 1$ with $\tau_h \in \{h+\ell, \dots, h+\ell+K+w-1\}$, and $|\tau_h - \tau_{h-1}| \leq K$.
If we consider $\bar{\gamma} \in \mathcal{S}^0$ such that $\bar{\gamma}_{h-1\,\tau_{h-1}} = \bar{\gamma}_{h\tau_{h-1}+1} = 1$ for $\tau_{h-1} \in \{h-1+\ell, \dots, h-1+\ell+w-1\}$, and then $\bar{\bar{\gamma}} \in \mathcal{S}^0$ such that $\bar{\bar{\gamma}}_{h-1,\tau_{h-1}+1} = \bar{\bar{\gamma}}_{h\tau_{h-1}+1} = 1$, and $\bar{\gamma}_{h't'} = \bar{\bar{\gamma}}_{h't'}$ for $h' \neq h, h-1$, we then get $\alpha_{h-1,\tau_{h-1}} = \alpha_{h-1,\tau_{h-1}+1}$.
So, all the coefficients for $\alpha_{h-1,\tau}$ with $\tau \in \{h-1+\ell, \dots, h+\ell+w-1\}$ are equal one another.

If we consider $\bar{\gamma}, \bar{\bar{\gamma}} \in \mathcal{S}^0$ such that

$$\begin{array}{ll} \bar{\gamma}_{h\tau_h} = \bar{\gamma}_{h-1,\tau_h} = 1 & \text{for } \tau_h \in \{h+\ell, \ldots, h+\ell+w\} \\ \bar{\bar{\gamma}}_{h,\tau_h+1} = \bar{\bar{\gamma}}_{h-1,\tau_h} = 1 & \\ \bar{\gamma}_{h't'} = \bar{\bar{\gamma}}_{h't'} & \text{for } h' \neq h, h-1 \end{array}$$

then we get $\alpha_{h\tau_h} = \alpha_{h,\tau_h+1}$.
If we consider $\bar{\gamma}, \bar{\bar{\gamma}} \in \mathcal{S}^0$ such that

$$\begin{array}{ll} \bar{\gamma}_{h\tau_h} = \bar{\gamma}_{h-1,\tau_h-i} = 1 & \text{for } \tau_h = h+\ell+w-1+i \text{ and } i \in \{1, \ldots, K\} \\ \bar{\bar{\gamma}}_{h,\tau_h-1} = \bar{\bar{\gamma}}_{h-1,\tau_h-i} = 1 & \\ \bar{\gamma}_{h't'} = \bar{\bar{\gamma}}_{h't'} & \text{for } h' \neq h, h-1 \end{array}$$

then we get $\alpha_{h\tau_h} = \alpha_{h,\tau_h-1}$. So, all the coefficients $\alpha_{h\tau}$ with $\tau \in \{h+\ell, \ldots, h+\ell+K+w-1\}$ are equal one another.
Now, let us consider $\bar{\gamma}, \bar{\bar{\gamma}} \in \mathcal{S}^1$ such that

$$\begin{array}{ll} \bar{\gamma}_{h-1,\tau_{h-1}} = \bar{\gamma}_{h,\tau_h} = 1 & \text{for } \tau_{h-1} \in \{h+\ell+w, \ldots, h-1+\mathcal{L}-1\} \\ & \text{and } \tau_h = \min\{\tau_{h-1}+K, h+\mathcal{L}\} \\ \bar{\gamma}_{h-1,\tau_{h-1}+1} = \bar{\gamma}_{h,\tau_h} = 1 & \\ \bar{\gamma}_{h't'} = \bar{\bar{\gamma}}_{h't'} & \text{for } h' \neq h, h-1 \end{array}$$

then we get $\alpha_{h-1,\tau_{h-1}} = \alpha_{h-1,\tau_{h-1}+1}$. So, all the coefficients $\tau \in \{h+\ell+w, h-1+\mathcal{L}\}$ are equal one another.
If we get $\bar{\gamma}, \bar{\bar{\gamma}} \in \mathcal{S}^1$ such that

$$\begin{array}{ll} \bar{\gamma}_{h,\tau_h} = \bar{\gamma}_{h-1,\tau_h-K} = 1 & \text{for } \tau_h \in \{h+\ell+K+w, \ldots, h+\mathcal{L}-1\} \\ \bar{\gamma}_{h-,\tau_h+1} = \bar{\gamma}_{h-1,\tau_h-K} = 1 & \\ \bar{\gamma}_{h't'} = \bar{\bar{\gamma}}_{h't'} & \text{for } h' \neq h, h-1 \end{array}$$

then we get $\alpha_{h,\tau_h} = \alpha_{h,\tau_h+1}$. So, all the coefficients $\tau \in \{h+\ell+K+w, \ldots, h+\mathcal{L}\}$ are equal one another.

For $h' \neq h$ and $h' \neq h-1$ we can take a solution in $\mathcal{S}^0$ and complete it in two different ways for (h', t) and $(h', t+1)$, thus obtaining $\alpha_{h',t} = \alpha_{h',t+1}$.

As the $dim(P) = NR = N(R+1) - N$, we can choose $N+1$ coefficients:

$$\begin{array}{ll} \alpha_0 = 0 & \\ \alpha_{h',t} = 0 & \text{for } h' \neq h, h' \neq h-1, t \in \{h'+\ell, \ldots, h'+\mathcal{L}\} \\ \alpha_{h-1,t} = 0 & \text{for } t \in \{h-1+\ell, \ldots, h+\ell+w-1\} \\ \alpha_{h,t} = 1 & \text{for } t \in \{h+\ell+K+w, \ldots, h+\mathcal{L}\}. \end{array}$$

Finally, if we consider a solution $\bar{\gamma} \in \mathcal{S}^1$, we get $\alpha\bar{\gamma} = \alpha_{h-1,t_1} + \alpha_{h,t_2} = \alpha_0 = 0$, that is, $\alpha_{h-1,t_1} = -\alpha_{h,t_2} = -1$; if we consider a solution $\bar{\gamma} \in \mathcal{S}^0$, we get $\alpha\bar{\gamma} = \alpha_{h-1,t_1} + \alpha_{h,t_2} = \alpha_0 = 0$, that is, $\alpha_{h,t_2} = -\alpha_{h-1,t_1} = 0$; thus $\alpha\gamma \leq \alpha_0$ is proved to be exactly as the selected inequality (4.14). □

So, we can conclude that inequalities (4.14), (4.15), and non-negativity inequalities are facet defining for P.

Completeness

Total Unimodularity of $\mathcal{M}$ Let's now define $\mathcal{M}$ as the constraint matrix defined by the inequalities (4.14), (4.15), (4.10). We prove that $\mathcal{M}$ is Totally Unimodular (TU). Before proving that, we need to recall some preliminaries.

Definition 2. *Let $G = (V, A)$ be a directed graph and let $\mathcal{T} = (V, A_{\mathcal{T}})$ be a directed spanning tree of G, i.e. $A_{\mathcal{T}} \subseteq A$. For each $a \in A$ let the a-path be the unique path in $\mathcal{T}$ with extreme v and w where $a = (v, w)$. A network matrix $N \in \mathbb{R}^{|A_{\mathcal{T}}| \times |A|}$ generated by $\mathcal{T}$ and G is defined as follows:*

$$N_{a'a} = \begin{cases} +1, & \textit{if the unique a-path in } \mathcal{T} \textit{ traverses } a' \textit{ forward,} \\ -1, & \textit{if the unique a-path in } \mathcal{T} \textit{ traverses } a' \textit{ backward,} \\ 0, & \textit{if the unique a-path in } \mathcal{T} \textit{ does not traverse } a'. \end{cases}$$

We can now recall the following result by [137]:

Theorem 4. *A network matrix is TU.*

A simpler proof is present in [125], Volume A, Section 13.3.

Definition 3. *A matrix is a co-network matrix if it is the transpose of a network matrix.*

Remark 1. *A matrix is TU if and only if its transpose is TU.*

We can now proceed to prove that the matrix $\mathcal{M}$ is TU.

Theorem 5. *The matrix $\mathcal{M}$ defined by inequalities* (4.14)*,* (4.15)*,* (4.10)*, and non-negativity inequalities is TU.*

Proof. By Theorem 4, Remark 1, and Definition 3, a co-network matrix is the transpose of a network matrix, and thus it is TU. In the following, we show that the matrix $\mathcal{M}$ is a co-network matrix, and thus it is TU. In particular, we will show that $[\mathcal{M}^T|I]$ is a network matrix. We proceed by showing that there exists a directed graph $G = (V, A)$ and a directed spanning tree $\mathcal{T}$, such that $[M^T|I]$ corresponds to the network matrix generated by $\mathcal{T}$ and G.

Let $\ell_h = h + \ell$ and $\mathcal{L}_h = h + \mathcal{L}$ for each h such that $h \in \bar{T}$.

We define the graph as follows.

- The set of nodes is

$$V = \{v\} \cup \{v_{ht} \mid h \in \bar{T}, t \in T_h\}$$

- The set of arcs A is partitioned into different subsets:

$$A = A_n \cup A_0 \cup A_1 \cup A_Q \cup A_B$$

  where:
  $A_n = \{(v, v_{h,\mathcal{L}_h}) \mid h \in \bar{T}\}$;
  $A_0 = \{(v_{ht}, v_{ht-1}) \mid h \in \bar{T},\ t \in \{\ell_h + 1, \dots, \mathcal{L}_h\}\}$;
  $A_1 = \{(v, v_{h,\ell_h}) \mid h \in \bar{T}\}$;
  $A_Q = \{(v_{h-1,t_1}, v_{h,t_2}) \mid h \in \bar{T} \setminus \{1\}, t_1 = h + \ell + w, t_2 = h + \ell + K + w, w \in \{0, \dots, W\}\}$;
  $A_B = \{(v_{h+1,t_1}, v_{h,t_2}) \mid h \in \bar{T} \setminus \{|\bar{T}|\}, t_1 = h+\ell+2+i, t_2 = h+\ell+K+2+i, i \in \{0, \dots, I\}\}$.

We define the directed spanning tree $\mathcal{T} = (V, A_\mathcal{T})$ where $A_\mathcal{T} = A_n \cup A_0$.

Each row of the matrix $\mathcal{M}$ associated with a constraint (4.14) for $h \in \bar{T} \setminus \{1\}$ and $w \in \{0, \dots, W\}$ corresponds to the path in $\mathcal{T}$ linking the extremes of the arc $(v_{h-1,h+\ell+w}, v_{h,h+\ell+K+w}) \in A_Q$, indeed the subpath from $v_{h-1,h+\ell+w}$ to v is traversed backward and the subpath from v to $v_{h,h+\ell+K+w}$ is traversed forward.
Each row of the matrix $\mathcal{M}$ associated with a constraint (4.15) for $h \in \bar{T}\setminus\{|\bar{T}|\}$ and $i \in \{0, \dots, I\}$ corresponds to the path in $\mathcal{T}$ linking the extremes of the arc $(v_{h+1,h+\ell+2+i}, v_{h,h+\ell+K+2+i}) \in A_B$, indeed the subpath from $v_{h+1,h+\ell+2+i}$ to v is traversed backward and the subpath from v to $v_{h+\ell+K+2+i}$ is traversed forward.
Each row of the matrix $\mathcal{M}$ associated with a constraint (4.10) corresponds to the path in $\mathcal{T}$ linking the extremes of the arc $(v, v_{h+\ell}) \in A_1$ that is completely traversed forward.

In conclusion, $N = [\mathcal{M}^T|I]$ is a network matrix and therefore it $\mathcal{M}$ is TU. $\square$

An example is given below.

Example 4. *The problem is defined by the following parameters:* $T = 10$, $\ell = 2$, $\mathcal{L} = 7$, $K = 2$. *Let's consider a reduced example for* $h \in \{2, 3\}$.
The matrix $\mathcal{M}$ *is shown below.*

	$h=1$						$h=2$						$h=3$					
t	3	4	5	6	7	8	4	5	6	7	8	9	5	6	7	8	9	10
(4.14) (h=2, w=0)	0	−1	−1	−1	−1	−1	0	0	1	1	1	1	0	0	0	0	0	0
(4.14) (h=2, w=1)	0	0	−1	−1	−1	−1	0	0	0	1	1	1	0	0	0	0	0	0
(4.14) (h=2, w=2)	0	0	0	−1	−1	−1	0	0	0	0	1	1	0	0	0	0	0	0
(4.14) (h=2, w=3)	0	0	0	0	−1	−1	0	0	0	0	0	1	0	0	0	0	0	0
(4.14) (h=3, w=0)	0	0	0	0	0	0	0	−1	−1	−1	−1	−1	0	0	1	1	1	1
(4.14) (h=3, w=1)	0	0	0	0	0	0	0	0	−1	−1	−1	−1	0	0	0	1	1	1
(4.14) (h=3, w=2)	0	0	0	0	0	0	0	0	0	−1	−1	−1	0	0	0	0	1	1
(4.14) (h=3, w=3)	0	0	0	0	0	0	0	0	0	0	−1	−1	0	0	0	0	0	1
(4.15) (h=1, i=0)	0	0	0	0	1	1	0	−1	−1	−1	−1	−1	0	0	0	0	0	0
(4.15) (h=1, i=1)	0	0	0	0	0	1	0	0	−1	−1	−1	−1	0	0	0	0	0	0
(4.15) (h=2, i=0)	0	0	0	0	0	0	0	0	0	0	1	1	0	−1	−1	−1	−1	−1
(4.15) (h=2, i=1)	0	0	0	0	0	0	0	0	0	0	0	1	0	0	−1	−1	−1	−1
(??) (h=1)	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
(??) (h=2)	0	0	0	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0
(??) (h=3)	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Table 4.1. Matrix $\mathcal{M}$ for the illustrative example with $T = 10$, $\ell = 2$, $\mathcal{L} = 7$, $K = 2$.

The header contains the first index h *and the second index* t *for variables* γ_{ht}. *The rows correspond to: (i) the constraints* (4.14) *for* $h \in \{2, 3\}$ *and* $w \in \{0, 1, 2, 3\}$*; (ii) the constraints* (4.15) *for* $h \in \{1, 2\}$ *and* $i \in \{0, 1\}$*; (iii) the constraints* (4.10) *for* $h \in \{1, 2, 3\}$.
The graph illustrated in Fig. 4.1 corresponds to the matrix $[\mathcal{M}^\mathcal{T}|\mathcal{I}]$*, where the tree arcs (solid red) correspond to the columns of* $\mathcal{M}$*, while the dashed blue arcs correspond to the rows of* $\mathcal{M}$ *associated with constraints* (4.14)*, the dotted teal arcs correspond to the rows of* $\mathcal{M}$ *associated with constraints* (4.15)*, and the dash-dot purple arcs correspond to the rows of* $\mathcal{M}$ *associated with constraints* (4.10).

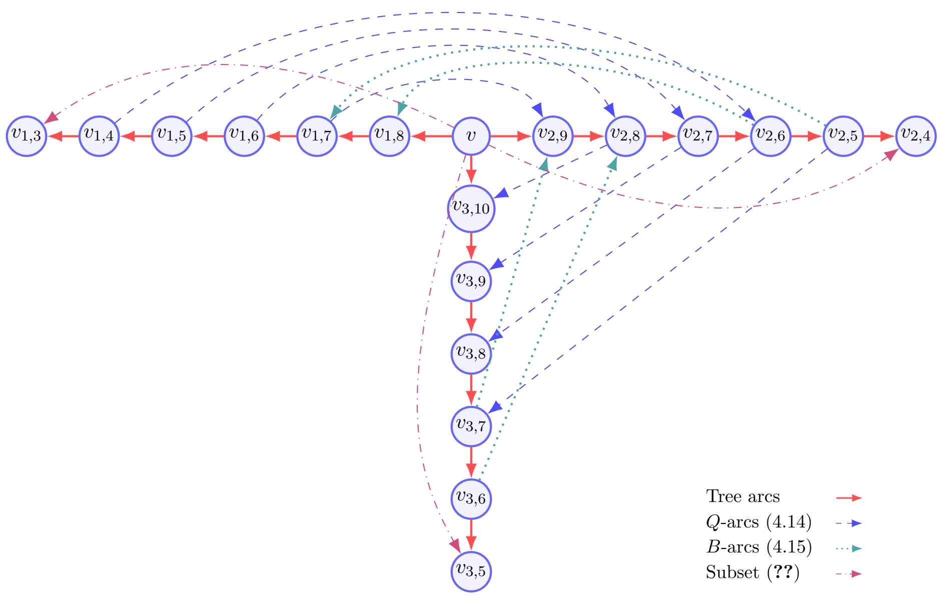


Figure 4.1. Graph representation $G(V, A)$ corresponding to matrix $[\mathcal{M}^{\mathcal{T}}|\mathcal{I}]$ for the illustrative example.

where:

- $A_n = \{g, p, z\}$
- $A_0 = \{a, b, \dots, f\} \cup \{h, i, \dots, o\} \cup \{q, r, \dots, v\}$
- $A_1 = \{\alpha, \beta, \gamma\}$
- $A_Q = \{1, 2, \dots, 5\} \cup \{9, 10, \dots, 13\}$
- $A_B = \{6, 7, 8\} \cup \{14, 15, 16\}$

The directed tree identified is $\mathcal{T} = (\mathcal{V}, \mathcal{A}_{\mathcal{T}})$ *where* $A_{\mathcal{T}} = A_N \cup A_0 = \{a, b, \dots, z\}$.
It can be easily shown that the network matrix represented by G *and* $\mathcal{T}$ *is equivalent to* $[\mathcal{M}^{\mathcal{T}}|\mathcal{I}]$.

Sufficiency Finally, we need to prove that constraints (4.14), (4.15), (4.10), (4.9), and non-negativity inequalities are sufficient to satisfy constraints (4.11). This is done by the following two propositions.

Proposition 1. *If all inequalities* (4.14) *are satisfied, then the following inequalities are satisfied for each* $h \in \bar{T}$*:*

$$\sum_{t=h+\ell}^{h+\mathcal{L}} t\gamma_{ht} - \sum_{t=h-1+\ell}^{h-1+\mathcal{L}} t\gamma_{h-1t} \leq K. \tag{4.16}$$

Proof. Suppose, for contradiction, that there exists $\bar{h}$ such that an inequality of type (4.16) is violated. If we sum all the inequalities (4.14) for $h = \bar{h}$ and all $w \in \{0, \dots, W\}$, we obtain:

$$\sum_{\tau=1}^{\mathcal{L}-\ell-K+1} \tau\gamma_{\bar{h},\,\bar{h}+\ell+K-1+\tau} - \left[\sum_{\tau=1}^{\mathcal{L}-\ell-K+1} \tau\gamma_{\bar{h}-1,\,\bar{h}-1+\ell+\tau} + \sum_{\tau=1}^{\mathcal{L}-K} (\mathcal{L}-\ell-K+1)\gamma_{\bar{h}-1,\,\bar{h}-K+\tau}\right] \leq 0 \tag{4.17}$$

is satisfied. By subtracting (4.17) to (4.16) for $h = \bar{h}$ and recalling that the last one has been supposed to be violated, we obtain the following:

$$\sum_{t=\bar{h}+\ell}^{\bar{h}+\ell+K-2} t\gamma_{\bar{h}t} + (\bar{h}+\ell+K-1)\sum_{t=\bar{h}+\ell+K-1}^{\bar{h}+\mathcal{L}} \gamma_{\bar{h}t}$$
$$-\left[(\bar{h}-1+\ell)\sum_{t=h-1+\ell}^{\bar{h}+\mathcal{L}-K} \gamma_{\bar{h}-1t} + \sum_{t=\bar{h}+\mathcal{L}-K+1}^{\bar{h}-1+\mathcal{L}} (t-\mathcal{L}+\ell+K-1)\gamma_{\bar{h}-1\,t}\right] > K.$$

We can rewrite the above as follows:

$$\sum_{i=0}^{K-2} (\bar{h}+\ell+i)\gamma_{\bar{h},\,\bar{h}+\ell+i} + (\bar{h}+\ell+K-1)\sum_{i=K-1}^{\mathcal{L}-\ell} \gamma_{\bar{h},\,\bar{h}+\ell+i}$$
$$-\left[(\bar{h}-1+\ell)\sum_{j=0}^{\mathcal{L}-K-\ell} \gamma_{\bar{h}-1,\,\bar{h}-1+j} + \sum_{i=1}^{K}(\bar{h}-1+\ell+i)\gamma_{\bar{h}-1,\,\bar{h}-1+\mathcal{L}-K+i}\right] > K.$$

By using equalities (4.10), we can further rewrite the above as follows:

$$\sum_{i=0}^{K-2} (\bar{h}+\ell+i)\gamma_{\bar{h},\,\bar{h}+\ell+i} + (\bar{h}+\ell+K-1)(1-\sum_{i=0}^{K-2} \gamma_{\bar{h},\,\bar{h}+\ell+i})$$
$$-\left[(\bar{h}-1+\ell)(1-\sum_{i=1}^{K}\gamma_{\bar{h}-1,\,\bar{h}-1+\mathcal{L}-K+i}) + \sum_{i=1}^{K}(\bar{h}-1+\ell+i)\gamma_{\bar{h}-1,\,\bar{h}-1+\mathcal{L}-K+i}\right] > K.$$

And finally, we get

$$K - \sum_{i=0}^{K-2}(K-1-i)\gamma_{\bar{h},\,\bar{h}+\ell+i} - \sum_{i=1}^{K} i\gamma_{\bar{h}-1,\,\bar{h}-1+\mathcal{L}-K+i} > K,$$

and, hence, we get a contradiction. □

Similarly, we can prove the following proposition:

Proposition 2. *If all inequalities* (4.15) *are satisfied, then the following inequalities are satisfied for each* $h \in \bar{T}$*:*

$$-\sum_{t=h+\ell}^{h+\mathcal{L}} t\gamma_{ht} + \sum_{t=h-1+\ell}^{h-1+\mathcal{L}} t\gamma_{h-1t} \leq K. \tag{4.18}$$

Proof is analogous to the one of the previous proposition and thus omitted.

Theorem 6. $P = \mathrm{conv}(\Gamma^s)$ *is completely described by inequalities* (4.14)*,* (4.15)*,* (4.10)*,* (4.9)*, and non-negativity inequalities.*

Proof. By Proposition 1 and Proposition 2, inequalities (4.14) and (4.15) together imply (4.11). Hence Γ^s is equivalent to the set

$$\{\gamma \in \{0,1\}^{|T|\times|T|} : (4.10), (4.9), (4.14), (4.15)\}. \tag{4.19}$$

By Theorem 5, the extreme points of the LP relaxation of the set defined by (4.14), (4.15), (4.10), (4.9), and non-negativity inequalities have $\{0,1\}$ coordinates. Therefore those inequalities completely describe P. □

4.4 An exact algorithm

We present in this section an exact approach for the Multiple Supplier Production Planning problem. In Section 4.4.1 we reformulate the model 2rMSPP[4.7] through a Benders-like approach and in Section 4.4.2 we present the branch-and-cut algorithm to solve the reformulated model, along with some techniques to speed up the solution process.

4.4.1 The Benders-like reformulation

We can now derive a Benders-like reformulation of model 2rMSPP[4.7], using an argument similar to the one in [99]. Note that the derivation and the final formulation is different from model (23)–(28) in [133], as we introduce Benders-like inequalities based on the values of dual variables, as we discuss below.

Observe that the inner min in the maxmin that computes λ is a linear programming problem. Below we report the problem and its dual. Recall that q and γ can be regarded as fixed constants when this problem is solved.

$$P(q,\gamma) \quad \min \sum_{t\in T}(c_t^+ I_t + c_t^- B_t) \tag{4.20a}$$

$$\text{s.t. } (z_t) \quad I_t - B_t = \sum_{s\in S}\sum_{h=1}^{t}\sum_{k=h}^{t} \gamma_{hk}^s q_h^s - \sum_{h=1}^{t} d_h \qquad t \in T \tag{4.20b}$$

$$I, B \in \mathbb{R}_+^{|T|} \tag{4.20c}$$

$$D(q,\gamma) \quad \max \sum_{t\in T}\left(\sum_{s\in S}\sum_{h=1}^{t}\sum_{k=h}^{t} \gamma_{hk}^s q_h^s - \sum_{h=1}^{t} d_h\right) z_t \tag{4.21a}$$

$$\text{s.t. } (I_t) \quad z_t \le c_t^+, \qquad t \in T \tag{4.21b}$$

$$(B_t) \quad -z_t \le c_t^-, \qquad t \in T \tag{4.21c}$$

$$z \in \mathbb{R}^{|T|} \tag{4.21d}$$

We observe that the primal problem is feasible and bounded. It is always feasible, as we can adapt the values of I and B to satisfy the constraints. Also, it is bounded, as the objective function is linear and the variables are non-negative. Therefore, by linear duality theory, the dual problem is also feasible and bounded, and the optimal values of the primal and dual problems are equal.
We can show that both the primal and the dual problem can be solved in closed form.

Theorem 7. *The optimal solution (I, B) of $P(q,\gamma)$ can be computed in closed form as follows.*

$$I_t = \max\left\{0, \sum_{s\in S}\sum_{h=1}^{t}\sum_{k=h}^{t} \gamma_{hk}^s q_h^s - \sum_{h=1}^{t} d_h\right\} \qquad t \in T \tag{4.22}$$

$$B_t = \max\left\{0, -\sum_{s\in S}\sum_{h=1}^{t}\sum_{k=h}^{t} \gamma_{hk}^s q_h^s + \sum_{h=1}^{t} d_h\right\} \qquad t \in T \tag{4.23}$$

The optimal solution z of $D(q,\gamma)$ can be computed as below.

$$z_t = \begin{cases} c^+ & \text{if } \sum_{s\in S}\sum_{h=1}^{t}\sum_{k=h}^{t}\gamma_{hk}^s q_h^s - \sum_{h=1}^{t} d_h \geq 0 \\ -c^- & \text{otherwise} \end{cases} \qquad t \in T \tag{4.24}$$

Proof. Let (σ^I, σ^B) be a primal solution and suppose that there exists $u \in T$ with $\sigma_u^I > 0$ and $\sigma_u^B > 0$. For the sake of simplicity let u be the only time period with this property and assume that $\sigma_u^I \geq \sigma_u^B$. Consider solution (r^I, r^B) defined as: $r_t^I = \sigma_t^I$, $r_t^B = \sigma_t^B$, for $t \in T \setminus \{u\}$; $r_u^B = 0, r_u^I = \sigma_u^I - \sigma_u^B$. Solution (r^I, r^B) is feasible and it has an objective cost smaller than (σ^I, σ^B). Then, any optimal primal solution (I, B) has the property that $I_t B_t = 0$ for all $t \in T$. It follows that the optimal solution is unique and that it is computed by (4.22)–(4.23). By the complementarity conditions of the linear duality theory, if $I_t > 0$, then the corresponding dual constraint must be satisfied with equality. The same happens if $B_t > 0$. This implies that the optimal dual solution is computed by (4.24). □

Denote by V the set of the vertices (extreme points) of the polyhedron that defines the feasible region of the dual problem and note that it does not depend on q or γ.

Theorem 8. *Model 2rMSPP can be reformulated as follows.*

$$\begin{aligned}
bMSPP \quad & \min \sum_{t\in T}\sum_{s\in S}(o^s y_t^s + p_t^s q_t^s) + \lambda \\
s.t. \quad & Ay + Dq \geq b \\
& \sum_{t\in T}\sum_{s\in S} q_t^s = \sum_{t\in T} d_t \\
& q_t^s \leq M y_t^s && t \in T, s \in S \\
& q_h^s = 0 && s \in S, h \in T, h + \mathcal{L}^s > |T| \\
& \lambda \geq \sum_{t\in T} v_t\Big(\sum_{s\in S}\sum_{h=1}^{t}\sum_{k=h}^{t}\gamma_{hk}^s q_h^s - \sum_{h=1}^{t} d_h\Big) && v \in V, \gamma \in \Gamma \qquad (4.25) \\
& y \in \{0,1\}^{|T|\times|S|}, q \in \mathbb{R}_+^{|T|\times|S|}, \lambda \in \mathbb{R}_+
\end{aligned}$$

Proof. Recall that the optimal solution of a linear programming problem, if any, corresponds to a vertex of the feasible region. As we discussed above, $P(q,\gamma)$ is always feasible and bounded and, by the linear duality theory, the same applies to $D(q,\gamma)$. Consider a vertex $v \in V$. By the weak duality theory, any v provides a lower bound on the optimal primal value and, hence, inequalities (4.25) are valid. By the strong duality theory, for any γ and q, there exists a vertex $v^{q\gamma} \in V$ that exactly provides the optimal value of problem $P(q,\gamma)$. Hence, inequalities (4.25) implement a correct formulation of the 2rMSPP problem. □

We refer to inequalities (4.25) as to the benders cuts. Note that, although exponential in number, the Benders cuts are finitely many, as we must consider one inequality for any vertex of the Γ polytope (provided that we know the convex hull of the integer feasible solutions) and any vertex of V. Model bMSPP can be solved by a branch-and-cut approach.

4.4.2 Branch-and-cut

The Benders-like formulation bMSPP can be solved by starting with a formulation without Benders cuts, which are added iteratively. Two approaches are possible: either adding a

Algorithm 1 Processing a node of the branch-and-bound tree

```
1: solve the linear relaxation of the problem at the node,
   let (y, q, λ) be the current solution and Csol its value,
   let Bsol be the current best upper bound
2: if Csol > Bsol then
3:     prune the node and exit
4: else
5:     if y is integer or we are at the root node then
6:         solve the separation problem
7:         if there exists a violated inequality (4.25) then
8:             add it to the current formulation and go back to step 1
9:         else
10:            exit
11:    else
12:        branch and exit
```

Benders-cut only after computing an integer solution or adding them during a unique branch-and-bound exploration, that is, using branch-and-cut. We decided to use this second policy. At each iteration we solve the linear relaxation of the current formulation, if we are in the root node or the solution is integer, we look for a violated Benders cut by solving a *separation* problem. If a violated cut has been found, the corresponding inequality is added to the current model, otherwise we either branch (if the solution was fractional) or we have found a feasible solution (if it was integer). The procedure we use at each node of the branch-and-bound tree is summarized in Algorithm 1.

The separation problem consists of solving the inner maxmin of formulation 2rMSPP[4.7]. Although the inner min can be solved in closed form (see Theorem 7), we can use an argument similar to the one in [133], to prove that the separation problem remains hard.

There exists two different ways to solve the separation problem. The first is to solve problem *psep*(q) below, which is equivalent to problem (17)–(22) in [133]. However, differently from what done in [133], we then use the optimal solution of *psep*(q) to compute the corresponding optimal dual vertex $v^{q\gamma}$, to be used to define the Benders cut (see the proof of Theorem 8).

$$
\begin{aligned}
psep(q) \quad & \max \sum_{t\in T}(c_t^+ I_t + c_t^- B_t) & \\
& I_t - B_t = \sum_{h=1}^{t}\sum_{s\in S}\sum_{k=h}^{t}\gamma_{hk}^s q_h^s - \sum_{h=1}^{t} d_h & t \in T \\
& I_t \leq M\alpha_t & t \in T \\
& B_t \leq M(1-\alpha_t) & t \in T \\
& I, B \in \mathbb{R}_+^{|T|}, \alpha \in \{0,1\}^{|T|}, \gamma \in \Gamma &
\end{aligned}
$$

Recall that q can be considered a constant parameter when the separation problem is solved. M can be set to $\sum_{h=1}^{t} d_h$. Binary variables α are used to replace the inner min of the maxmin problem, as they enforce the optimality condition $I_t B_t = 0$ for all $t \in T$ discussed in Theorem 7. Once the worst case realization $\gamma \in \Gamma$ is known by the optimal solution of *sep*(q), the optimal dual vertex $v^{q\gamma}$ to be used for the Benders cut is computed by (4.24).

Another possibility is to compute the worst case realization γ and the dual vertex $v^{q\gamma}$ directly, solving problem *dsep*(q).

$$
\begin{aligned}
dsep(q) \quad & \max \sum_{t\in T}\Big(\sum_{s\in S}\sum_{h=1}^{t}\sum_{k=h}^{t}\gamma_{hk}^{s}q_h^s - \sum_{h=1}^{t} d_h\Big)v_t \\
& v_t \le c_t^+ && t \in T \\
& -v_t \le c_t^- && t \in T \\
& v \in \mathbb{R}^{|T|}, \gamma \in \Gamma
\end{aligned}
$$

This problem has a quadratic objective function. However, since γ is binary, the quadratic product $\gamma_{hk}^s v_t$ can be linearized as follows. Replace v_t by $v_t^I - v_t^B$ and let $\eta_{hk}^{st} = v_t^I \gamma_{hk}^s$ and $\mu_{hk}^{st} = v_t^B \gamma_{hk}^s$. The constraints of the problem are replaced by the ones below.

$$
\begin{aligned}
& 3\eta_{hk}^{st} \le M_t^I \gamma_{hk}^s && s \in S, h, k, t \in T \\
& \mu_{hk}^{st} \le M_t^B \gamma_{hk}^s && s \in S, h, k, t \in T \\
& \eta_{hk}^{st} \le v_t^I && s \in S, h, k, t \in T \\
& \mu_{hk}^{st} \ge v_t^B - M_t^I(1-\gamma_{hk}^s) && s \in S, h, k, t \in T \\
& v_t^I \le M_t^I \chi_t && t \in T \\
& v_t^B \le M_t^B(1-\chi_t) && t \in T \\
& v^I, v^B \in \mathbb{R}_+^{|T|}, \mu, \eta \in \mathbb{R}^{|S|\times|T|^3}, \chi \in \{0,1\}^{|T|}, \gamma \in \Gamma
\end{aligned}
$$

where M_t^I can be set to c_t^+ and M_t^B can be set to c_t^-. Some preliminary computational tests showed that solving *psep*(q) is faster than solving *dsep*(q), thus we decided to adopt the first approach for the separation problem.

4.4.3 Speeding up the convergence

Some techniques have been adopted to speed up the convergence of the branch-and-cut algorithm.

Valid Inequalities. In the separation problem, constraints can be replaced by the valid inequalities (4.11) and (4.12) derived in Section 4.3.2, as discussed. These inequalities strengthen the polyhedral structure of the uncertainty set and improve the performance of the algorithm, as shown in the computational results in Section 4.7.1.

Cut Separation Strategy. To avoid generating a large number of Benders cuts, we adopt a cut separation strategy that generates cuts only for integer solutions or at the root node, as illustrated in Algorithm 1. In particular,

- Separation at integer solutions: Invoked at every integer solution to ensure feasibility. The separation problem *sep*(q) is always solved for integer solutions to add violated Benders cuts. This guarantees that all integer solutions satisfy the two-stage robust constraints.

- Separation at fractional solutions: This is more computationally expensive and may generate many cuts that are not effective in pruning the search tree. Thus, we limit the separation at fractional nodes to only the root node, where it can significantly improve the lower bound early in the search. We compare four different strategies for invoking the separation problem at fractional nodes: (i) at every node (for all fractional solutions), (ii) only at the root node (for all fractional solutions at the root), (iii) at the root node every 5 iterations, and (iv) never (no user cuts for fractional solutions). Some computational tests show that the strategy (iii) provides a good balance between computational overhead and solution quality and achieves the best performance. This is the strategy we adopt for all experiments reported in Section 4.7.1.

Feasible Solutions. At each node, whenever a new incumbent solution is found for the master problem variables (x, y, λ), the separation problem $sep(q)$ is solved to identify any violated Benders cuts. Thus, putting together the incumbent solution from the master problem and the solution of the separation problem, we obtain a complete feasible solution for the original two-stage robust problem and we submit it to the solver, using it to help identify high-quality feasible solutions early in the search process.

Priority Branching. To guide the branching strategy, we assign higher branching priority to the first-stage binary variables y (supplier selection and ordering decisions). This ensures that the solver explores the search tree by first fixing the supplier selection decisions, which are the most critical decisions from a modeling perspective, before branching on other variables.

Lower bounds

Since the Benders cuts are generated iteratively, the initial formulation of the master problem is a relaxation of the original problem and, hence, it may provide a very loose lower bound on the optimal value for the λ variable, which represents the worst-case inventory cost. To obtain tighter lower bounds on λ throughout the branch-and-cut search, we derive a valid inequality that exploits the structure of the inventory-backlogging cost as a function of when the first and the last order are placed.
Let $L_{\min} = \min_{s \in S} \ell^s$ and $L_{\max} = \max_{s \in S} \mathcal{L}^s$ denote the shortest and longest possible lead times across all suppliers. Define cumulative demand quantities $D_u^{\leq} = \sum_{\tau=1}^{u} d_\tau$ and $D_u^{>} = \sum_{\tau=u+1}^{T} d_\tau$.
We introduce four families of auxiliary binary variables:

- $p_t \in \{0, 1\}$, equal to 1 if at least one order has been placed in some period $\tau \leq t$;
- $n_t \in \{0, 1\}$, equal to 1 if at least one order will be placed in some period $\tau \geq t$;
- $b_t \in \{0, 1\}$, equal to 1 if t is the *first* period in which an order is placed ($b_t = p_t - p_{t-1}$);
- $f_t \in \{0, 1\}$, equal to 1 if t is the *last* period in which an order is placed ($f_t = n_t - n_{t+1}$).

The variables p_t and n_t are enforced to be non-decreasing and non-increasing, respectively, by the monotonicity constraints

$$
\begin{aligned}
p_t &\geq p_{t-1} && \forall\, t = 2, \ldots, T, \\
n_t &\geq n_{t+1} && \forall\, t = 1, \ldots, T-1.
\end{aligned}
$$

Their activation is governed by the ordering decisions through

$$p_t \geq y_t^s, \quad n_t \geq y_t^s \qquad \forall\, t,\, s \in S,$$

while cumulative upper bounds ensure that p_t (resp. n_t) can be 1 only if at least one order has actually been placed up to t (resp. from t onwards):

$$\sum_{\tau=1}^{t} \sum_{s\in S} y_\tau^s \geq p_t \qquad \forall\, t,$$
$$\sum_{\tau=t}^{T} \sum_{s\in S} y_\tau^s \geq n_t \qquad \forall\, t.$$

The "marker" variables b_t and f_t are linearised as

$$b_t \geq p_t - p_{t-1}, \quad b_t \leq 1 - p_{t-1}, \quad b_t \leq p_t \qquad \forall\, t,$$
$$f_t \geq n_t - n_{t+1}, \quad f_t \leq 1 - n_{t+1}, \quad f_t \leq n_t \qquad \forall\, t,$$

which together force $b_t = 1$ if and only if t is the unique period where p switches from 0 to 1, and $f_t = 1$ if and only if t is the unique period where n switches from 1 to 0.
The lower bound on λ is then derived from two structural observations. If the first order is placed in period t (i.e. $b_t = 1$), no goods can arrive before period $t + \ell_{\min}$; hence, in every period $\tau \leq t + \ell_{\min}$ the system faces a backlog of at least $D_\tau^{\leq}$. Symmetrically, if the last order is placed in period t (i.e. $f_t = 1$), all orders have been received by period $t + L_{\max}$ at the latest; hence, in every period $\tau \geq t + L_{\max}$ the inventory is at least $D_\tau^{>}$, i.e. the residual demand that has not yet occurred. This yields the valid inequality

$$\lambda \;\geq\; \sum_{t=1}^{T} c_t^-\, b_t \sum_{\tau=1}^{t+\ell_{\min}} D_\tau^{\leq} \;+\; \sum_{t=1}^{T} c_t^+\, f_t \sum_{\tau=t+L_{\max}}^{T} D_\tau^{>}\,, \tag{4.26}$$

which is active only when $b_t = 1$ (first order period) or $f_t = 1$ (last order period), respectively. Inequality (4.26) is a valid lower bound since it accounts for the unavoidable inventory costs at the two ends of the planning horizon, regardless of the specific scenario $\gamma \in \Gamma$.
Nevertheless, it is not tight in general, as it does not consider the inventory costs incurred in the intermediate periods between the first and last order. Some preliminary computational tests showed that adding inequality (4.26) to the formulation does not speed up the convergence of the branch-and-cut algorithm significantly. For this reason, we decided not to include it in the final formulation.

4.5 A heuristic approach

Although the exact algorithm of Section 4.4 solves 2rMSPP [4.7] to optimality, its computational cost may become prohibitive on instances with many time periods. To address this limitation, we propose a hybrid heuristic grounded in a simple but effective planning principle: decisions concerning near-term periods have an immediate operational impact and should therefore be based on an exact evaluation of the worst-case cost; decisions concerning the medium-to-long term, by contrast, have a less immediate impact—and, by the time they are actually executed, the underlying data (even those considered "known" at the time of planning,

such as demands, costs, lead times) may have changed. It is therefore acceptable, for these periods, to base the decisions on a conservative approximation of the cost rather than on its exact worst-case value.
This principle motivates a partition of T into two complementary sub-horizons, $T^1 \subseteq T$ and $T^2 = T \setminus T^1$. The former contains the first $\beta \in \{0, 1, \cdots, |T|\}$ time periods, which are modelled through the formulation 2rMSPP [4.7] and solved exactly via the branch-and-cut algorithm of Section 4.4; the latter contains the remaining periods, which are modelled through the robust counterpart formulations sMSPP [4.27] and included in the master problem of the branch-and-cut algorithm as a single-stage approximation of the original two-stage problem. Robust counterpart is computationally lighter and provides a conservative upper bound on the inventory/backlogging costs. The resulting formulation, which we call hMSPP, is described in detail in Section 4.5.2.

4.5.1 The robust counterpart model

In this section we present the robust counterpart model sMSPP [4.27], which is used as a building block for the horizon-split heuristic. The formulation is based on the one proposed by [133]. The key difference between 2rMSPP[4.7] and sMSPP [4.27] lies in how the adversary is modelled. In 2rMSPP, a single scenario $\gamma \in \Gamma$ governs all time periods simultaneously, so the worst-case inventory/backlogging cost is evaluated jointly across T. The robust counterpart relaxes this coupling: each period $t \in T$ is allowed to face its own, independent worst-case realization of γ, yielding a separate maximization problem per period. This period-by-period decoupling is a relaxation of 2rMSPPand therefore provides an upper bound on its optimal value.

$$\text{sMSPP} \quad \min \sum_{t\in T}\sum_{s\in S}(o^s y_t^s + p_t^s q_t^s) + \lambda \tag{4.27a}$$

$$\text{s.t.} \quad Ay + \mathbb{R}q \geq b \tag{4.27b}$$

$$q_t^s \leq M y_t^s \qquad t \in T, s \in S \tag{4.27c}$$

$$\lambda \geq \sum_{t\in T} c_t \tag{4.27d}$$

$$c_t \geq \max_{\gamma\in\Gamma}\left\{c_t^+\left(\sum_{s\in S}\sum_{h=1}^{t}\sum_{k=h}^{t}\gamma_{hk}^s q_h^s - \sum_{h=1}^{t} d_h\right)\right\} \quad t \in T \tag{4.27e}$$

$$c_t \geq \max_{\gamma\in\Gamma}\left\{c_t^-\left(\sum_{h=1}^{t} d_h - \sum_{s\in S}\sum_{h=1}^{t}\sum_{k=h}^{t}\gamma_{hk}^s q_h^s\right)\right\} \quad t \in T \tag{4.27f}$$

$$y \in \{0,1\}^{|T|\times|S|}, \quad q \in \mathbb{R}_+^{|T|\times|S|} \tag{4.27g}$$

$$I, B \in \mathbb{R}_+^{|T|}, \quad \lambda \in \mathbb{R}_+ \tag{4.27h}$$

Contraint 4.27d computes the total inventory/backlogging cost λ as the sum of the costs c_t for each period $t \in T$. (4.27e) (resp. (4.27f)) computes the worst-case backlogging (resp. inventory) cost for period t by maximizing over all possible realizations of the uncertainty $\gamma \in \Gamma$. The optimal λ value provides an upper bound on the inventory/backlogging costs for optimal values $(y*, q*)$. Here the computed (I, B) does not have the property that, for a single period t, at most one of I_t and B_t is positive ($I_t B_t = 0$ for $t \in T$), as it happens in 2rMSPP. In fact, the worst case inventory/backlogging is computed independently for any $t \in T$, so it may happen that, for a given t, the worst case for inventory is obtained for a different γ than the one for backlogging, yielding $I_t > 0$ and $B_t > 0$ at the same time. This is a consequence of the decoupling of the periods in the robust counterpart, which allows each period to see a different realization of the uncertainty. In 2rMSPPinstead, all the time periods see the same realization at the same time, so the worst case inventory/backlogging is computed jointly across T.

To derive a compact reformulation of this problem using linear duality, we have to relax the constraint the γ must be binary. Any solution of the relaxed problem also provides an upper bound for the problem where γ is binary. Consider the max problem in constraints (4.27e) and (4.27f). Once the integrality of γ is relaxed, they are linear programming problems in the continuous variables γ, while q can be regarded as a constant. Since those problems are feasible and bounded, it is well known that they can be replaced by their duals to obtain a compact reformulation [26]. In fact, while the max of the primal version is not redundant, the min of the dual is redundant and it can be removed, obtaining a compact single level reformulation of the problem. We omit the details of the reformulation, that can be obtained using the same procedure illustrated in [26, 133].

4.5.2 The horizon-split heuristic

In this section we present the horizon-split heuristic hMSPP, which combines the exact two-stage formulation 2rMSPP [4.7] on a near-term sub-horizon with the single-stage robust counterpart sMSPP [4.27] on the remaining periods. Formally, given a parameter $\beta \in \{0, 1, \dots, |T|\}$, we partition T into $T^1 \subseteq T$ containing the first β periods ($T^1 = \{1, \dots, \beta\}$) and $T^2 \subseteq T$ containing the periods from $\beta + 1$ on, if any ($T^2 = \{\beta + 1, \dots, |T|\}$). On T^1, we solve the exact two-stage model to obtain precise cost estimates; on T^2, we use the robust counterpart to compute an upper bound on inventory and backlogging costs. The splitting parameter β controls the trade-off between solution quality and running time: a larger β implies more periods treated exactly, improving the bound at the cost of increased computation. The two limiting cases recover existing methods: when $T^1 = T$ (i.e. $\beta = |T|$), the heuristic reduces to the two-stage formulation 2rMSPP [4.7]; when $T^2 = T$ (i.e. $\beta = 0$), it reduces to the robust counterpart sMSPP [4.27]. When both sub-horizons are non-empty, we obtain the horizon-split heuristic hMSPP [4.28].

Note that, since T^1 does not extend to the end of the horizon, the terminal condition $I_{|T|} = B_{|T|} = 0$ imposed in the exact model (see Section 4.2) must be relaxed for the two-stage subproblem on T^1; otherwise, the subproblem may become infeasible. Another way to ensure feasibility of the two-stage subproblem on T^1 is by imposing $\beta \geq max_{s \in S} \mathcal{L}$, to guarantee that at least one order can be placed and delivered within T^1.

We choose the first option for full flexibility, allowing β to take any value in $\{0, 1, \dots, |T|\}$ and letting the model determine the optimal inventory/backlogging levels at the end of T^1. The resulting formulation is below.

$$\text{hMSPP} \quad \min \sum_{t \in T} \sum_{s \in S} (o^s y_t^s + p_t^s q_t^s) + \lambda \tag{4.28a}$$

$$
\begin{aligned}
\text{s.t.}\quad & Ay + \mathbb{R}q \geq b && && (4.28b)\\
& q_t^s \leq M y_t^s && t \in T, s \in S && (4.28c)\\
& \lambda = \lambda^1 + \lambda^2 && && (4.28d)\\
& \lambda \geq \sum_{t \in T^2} c_t && && (4.28e)\\
& c_t \geq \max_{\gamma \in \Gamma} \left\{ c_t^+ \left(\sum_{s \in S} \sum_{h=1}^{t} \sum_{k=h}^{t} \gamma_{hk}^s q_h^s - \sum_{h=1}^{t} d_h \right) \right\} && t \in T^2 && (4.28f)\\
& c_t \geq \max_{\gamma \in \Gamma} \left\{ c_t^- \left(\sum_{h=1}^{t} d_h - \sum_{s \in S} \sum_{h=1}^{t} \sum_{k=h}^{t} \gamma_{hk}^s q_h^s \right) \right\} && t \in T^2 && (4.28g)\\
& y \in \{0,1\}^{|T| \times |S|}, \quad q \in \mathbb{R}_+^{|T| \times |S|} && && (4.28h)\\
& I, B \in \mathbb{R}_+^{|T^2|}, \quad \lambda, \lambda^1 \lambda^2 \in \mathbb{R}_+ && && (4.28i)\\
& \lambda^1 = \max_{\gamma \in \Gamma} \min \sum_{t \in T^1} (c_t^+ I_t + c_t^- I_t) && && (4.28j)\\
& \qquad I_t - B_t = \sum_{s \in S} \sum_{h=\beta+1}^{t} \sum_{k=h}^{t} \gamma_{hk}^s q_h^s - \sum_{h=1}^{t} d_h && t \in T^1 && (4.28k)\\
& \qquad I, B \in \mathbb{R}_+^{|T^1|} && && (4.28l)
\end{aligned}
$$

The inner maxmin in 4.28j computes the worst-case inventory/backlogging cost λ^1 for the first β periods, while constraints 4.28f-4.28g compute an upper bound on the inventory/backlogging cost λ^2 for the remaining periods. The value of λ is computed as the sum of the two contributions in constraint 4.28d.
The inner maxmin in 4.28j can be replaced by Benders cuts as we do in 2rMSPP, while the max problems in 4.28f-4.28g can be relaxed and replaced by their dual as in sMSPP.
One could decide, in principle, to decompose the original problem into more than two problems. However, the more you decompose, the more expensive the combined solution becomes. Hence, we chose a binary decomposition.

4.6 Computational Setup and Experimental Design

In this section, we discuss the setup and design of our computationalstudy. In Section 4.6.1, we elaborate on the hardware, the software, and the solver used in our computational study. Afterward, in Section 4.6.2, we describe our test instances.

4.6.1 Computational Setup

All experiments were performed on a 64-bit Linux system with an Intel Core i9-13900K (3 GHz) and 128 GB RAM using 4 threads. The models and algorithms were implemented in Python 3.10.15 and solved using IBM ILOG CPLEX 22.1.1. We set a time limit of 600 seconds per instance and use a relative optimality gap tolerance of 10^{-3}. To guide the branching strategy, we assign higher branching priority to the first-stage binary variables y. All the

other parameters are the their default values. We employ two types of callbacks in CPLEX to dynamically generate Benders cuts: the Lazy constraints callback and the User cuts callback. We also use the Heuristic Callback to pass to the solver the best solution obtained combining (y, q) provided by the master and λ obtained in the separation phase (see Section 4.4.3).
Code and instances are publicly available at
https://github.com/fragiancola/mspp-under-uncertainty.

4.6.2 Instances

The test instances were generated taking inspiration from [133], but adapted to our setting with some modifications to account for the specific characteristics of our two-stage robust model and uncertainty set formulation.
We consider five different time horizons $|T| \in \{20, 30, 40, 50, 60\}$ and four different sets of suppliers $|S| \in \{5, 10, 15, 20\}$. For each combination of time horizon and number of suppliers, ten instances were generated, yielding a total of 200 instances. The performance of the algorithm is evaluated by averaging the results over the various instances in each group.
Demand for each period is drawn from a uniform distribution in $[20, 40]$ starting from the first period. For each supplier $s \in S$, the base value for the unit purchasing cost and for the fixed ordering cost are generated randomly in $[5, 20]$ and in $[12, 14]$ respectively. For each time period $t \in T$, this base value of cost is then varied within $\pm 50\%$ to account for time-dependent cost fluctuations. Similarly, inventory and back-ordering costs also vary across time within $\pm 50\%$ of their base values $c^+ = 0.2$ and $c^- = 2.4$, respectively, making it much more expensive to backlog than storing inventory.
For each supplier $s \in S$, lead time intervals $[\ell^s, \mathcal{L}^s]$ are generated by first drawing a mean lead time L_{mean} from $\{2, 4, 6\}$, a lower deviation L_{low} from $\{0, 1, 2, 3\}$, and an upper deviation L_{up} from $\{0, 1, 2, 3\}$. Then, $\ell^s = \max\{0, L_{\text{mean}} - L_{\text{low}}\}$ and $\mathcal{L}^s = \min\{L_{\text{mean}} + L_{\text{up}}, T\}$. This procedure generates a variety of lead time profiles, including some rare cases with zero variability ($\ell^s = \mathcal{L}^s$) and some with larger variability (width of 5 or 6), for an average interval width of 3 periods.
Regarding the uncertainty set, the parameter K, representing the maximum delay difference between consecutive orders from the same supplier, is randomly sampled from $[2, 5]$ for each instance.
The parameter H, which bounds the total delay across all suppliers, is set based on practical considerations. Specifically, we assume that at most 20% of the total orders placed may experience delays. Accordingly, for each planning horizon T, we compute the average number of orders in the optimal deterministic solution and use this value to calibrate H.

4.7 Computational Results and Discussion

4.7.1 Exact Algorithm Performance

Scalability Analysis We analyze the scalability of the exact branch-and-cut algorithm with respect to the dimensions of the problem, focusing on the two most relevant parameters: the time horizon $|T|$ and the number of suppliers $|S|$. Figure 4.2 displays the success rate as a heatmap over the $(|T|, |S|)$ grid. The success rate is computed as the percentage of instances solved to optimality within the time limit. For $|T| \leq 40$, all 120 instances are solved to optimality. When $|T|$ increases to 50, the success rate drops to 80% for $|S| = 5$ and to 20%

for $|S| = 20$. For $|T| = 60$ and $|T| = 80$, only 7 and 1 instances are solved, respectively. This transition between $|T| = 40$ and $|T| = 50$ is sharp: below the threshold, all instances are tractable and easily solved; above it, the algorithm fails on the majority of cases. Thus, we can identify a first region ($|T| \leq 40$), where all the problems are easily solved, and a second region ($|T| \geq 50$), where this is not always the case.

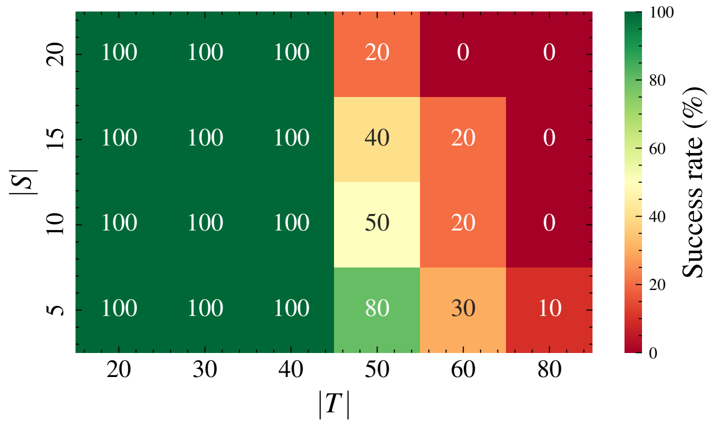


Figure 4.2. Success rate (%) for each ($|T|, |S|$) configuration, where T is the time horizon and S is the number of suppliers. The success rate is computed as the percentage of instances solved to optimality within the time limit.

Table 4.2. Average execution time for solved instances ($|T| \leq 40$) and average optimality gap for unsolved instances ($|T| \geq 50$).

		Avg. Time (s)			Avg. Gap (%)		
$\lvert S\rvert$	$\lvert T\rvert$	20	30	40	50	60	80
5		0.3	3.1	15.7	0.9	2.5	4.3
10		0.6	8.4	63.3	3.1	3.3	10.7
15		1.3	21.9	130.1	7.4	7.4	19.3
20		1.9	29.3	222.7	7.8	12.8	16.2

Within the second region, $|S|$ becomes a critical factor in determining tractability. We further analyze the impact of $|T|$ and $|S|$ on execution times (for the instances solved to optimality, with $|T| \leq 40$) and optimality gaps (for the instances not solved to optimality, with $|T| \geq 50$). Table 4.2 presents a detailed view of these metrics across the different configurations ($|T|$,$|S|$). For the solved instances, execution time increases with both $|T|$ and $|S|$. Moreover, the impact of the two parameters is correlated: a larger number of suppliers has a more evident effect on execution times as the length of the time horizon increases. This is evident when comparing the time increase due to $|S|$ for the same values of $|T|$. Similarly, for the unsolved instances, the optimality gap at termination increases with both $|T|$ and $|S|$. However, when the number of suppliers is small, the gap remains below 5%. As $|S|$ increases, the gap grows: it exceeds 10% for $|S| \geq 15$ and reaches 19.3% for $|T| = 80$ and $|S| = 15$.

Impact of Valid Inequalities We compare the performance of the branch-and-cut algorithm with and without the inequalities defined in Section 4.3. Table 4.3 reports the computational results aggregated by $|T|$. Valid inequalities provide a modest but persistent computational benefit. On the 145 instances solved by both formulations, the use of valid inequalities yields an average speed-up of 1.23×, with improvements ranging from 1.07× when $|T| = 20$ to 1.45× if $|T| = 30$ and 1.36× at $|T|$=50.

Table 4.3. The number of instances solved to optimality ("#opt"), the average execution time ("time", in s) and the average optimality gap ("Gap", in %) are reported for the two-stage formulations for different values of $|T|$. Additionally, the speed-up ratio ("speed-up"), and the difference in the optimality gap ("ΔGap") are reported to compare the two formulations.

	With VI			Without VI			Comparison	
$\lvert T\rvert$	#opt	time(s)	Gap(%)	#opt	time(s)	Gap(%)	speed-up	ΔGap
20	40/40	1.0	–	40/40	1.1	–	1.07	–
30	40/40	15.7	–	40/40	22.8	–	1.45	–
40	40/40	101.3	–	39/40	128.6	–	1.27	–
50	19/40	133.3	9.5	19/40	181.7	11.1	1.36	-1.6
60	7/40	82.6	8.4	9/40	75.8	11.1	0.92	-2.7
80	1/40	113.9	13.2	1/40	98.7	14.7	0.87	-1.6
Tot.	147/240	74.63	10.7	148/240	84.78	12.6	1.23	-1.9

For the 90 instances unsolved by both formulations, the version with valid inequalities yields lower optimality gaps, with a reduction of 1.9 percentage points on average. The gap improvement is more evident for $|T|$=60 (-2.7%).

Time Distribution In this paragraph, we provide detailed insights into the algorithmic behavior of the branch-and-cut method, analyzing how computational effort is distributed across different components. Figure 4.3 shows how computational time is distributed across different algorithmic components for each $(|T|, |S|)$ configuration. Figure 4.3 (a) displays the share of time spent at the root node versus the rest of the branch-and-bound tree. Figure 4.3 (b) shows the relative time spent on the separation subproblem versus the master problem.
The figure reveals two key patterns. First, the time spent at the root node decreases as instances become larger. Second, the separation consumes most of the total execution time, especially for larger instances.

4.7.2 Heuristic Algorithm Performance

Recall that the heuristic is based on a *hybrid* approach controlled by the parameter $\beta \in [0, |T|]$, which determines the extent of exact two-stage optimization versus robust-counterpart approximation.

Solution Quality Analysis We consider instances with $|T| \in \{20, 30, 40\}$ and $|S| \in \{5, 10, 15, 20\}$. Out of the 120 instances, we restrict our analysis to the 112 instances solved to optimality within the time limit for every value of β, to avoid bias due to incomplete data. These instances include all 40 instances with $|T| = 20$, 39 out of 40 instances with $|T| = 30$, and 32 out

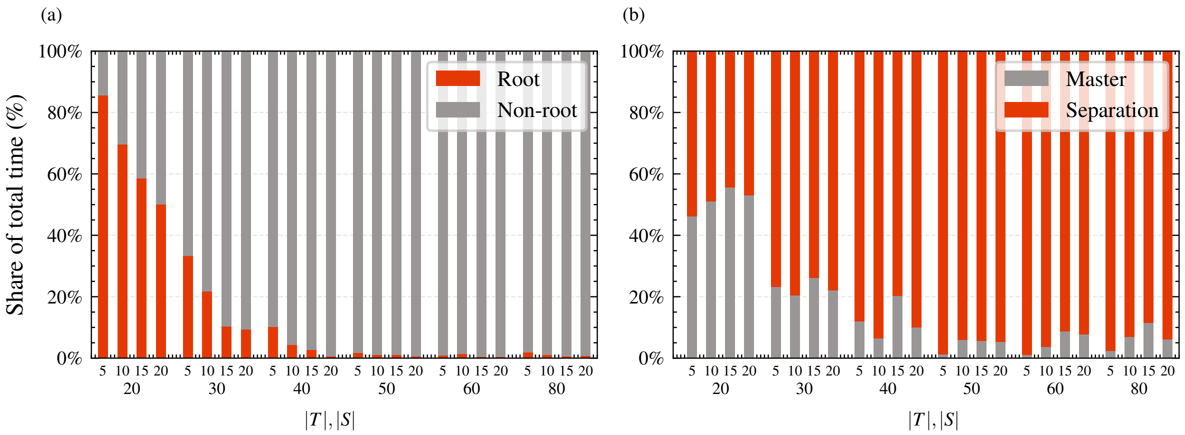


Figure 4.3. Time breakdown across ($|T|, |S|$) configurations. (a): share of total time spent at the root node versus the branch-and-bound tree. (b): share of combined master and separation time spent on the master problem versus the separation subproblem.

of 40 instances with $|T| = 40$. For each of these instances, we compare the heuristic objective value (for varying β) against the optimal value obtained by the exact method. Figure 4.4(a) reports the deviation from the optimal objective value as a function of $\beta/|T|$, with one curve per horizon length ($|T| \in 20, 30, 40$). Each point on a curve is obtained by averaging over all instances sharing the same $|T|$ and β, across all values of $|S|$. The deviation from optimality decreases steadily as β increases, for all values of T. The rate of decrease is steeper for small β/T and gradually moderates as β/T approaches 1. Although the change in slope is mild, three regions can be identified in the plot: when $\beta/T \in [0, 0.25]$, the decrease in deviation is steepest, meaning that small increases in β yield significant improvements in solution quality; when $\beta/T \in [0.25, 0.75]$, the deviation continues to decrease, but the improvement per unit increase is more moderate; when $\beta/T \in [0.75, 1]$, the marginal gain becomes less pronounced, indicating diminishing returns as the solution approaches the exact two-stage formulation.
Figure 4.4(b) disaggregates the results according to $|S|$. The deviation when $\beta = 0$ ranges from 3.5% ($|T| = 20$, $|S| = 5$) to 8.0% ($|T| = 30$, $|S| = 20$), confirming that the robust counterpart leaves a non-negligible gap with respect to the exact solution. Increasing β reduces this gap. When $\beta = T/2$, the residual deviation is already below half of its initial value. Overall, these results indicate that a moderate two-stage window covering approximately half the planning horizon captures a substantial portion of the benefit of the full exact formulation.

Scalability We now analyze how the heuristic scales with instance size. Figure 4.5 shows the success rate (percentage of instances solved within the 600-second time limit) for $|T| \in \{50, 60, 80\}$ and $|S| \leq 20$ as a function of β.
The results show that the heuristic enables solving instances that would be intractable with the exact approach. For $|T| = 50$, the heuristic solves the majority of instances to optimality for $\beta \leq 20$, with success rates above 75%, while the exact algorithm ($\beta = 50$) solves only about 40% of instances. For $|T| = 60$, where the exact approach struggles further (solving only 22% of instances), the heuristic achieves success rates around 55% for $\beta \leq 20$. For $|T| = 80$, the instances become impossible to solve exactly within the time limit. The heuristic still provides value for smaller numbers of suppliers: for $|S| = 5$ and $\beta \leq 30$, the success rate reaches 90%,

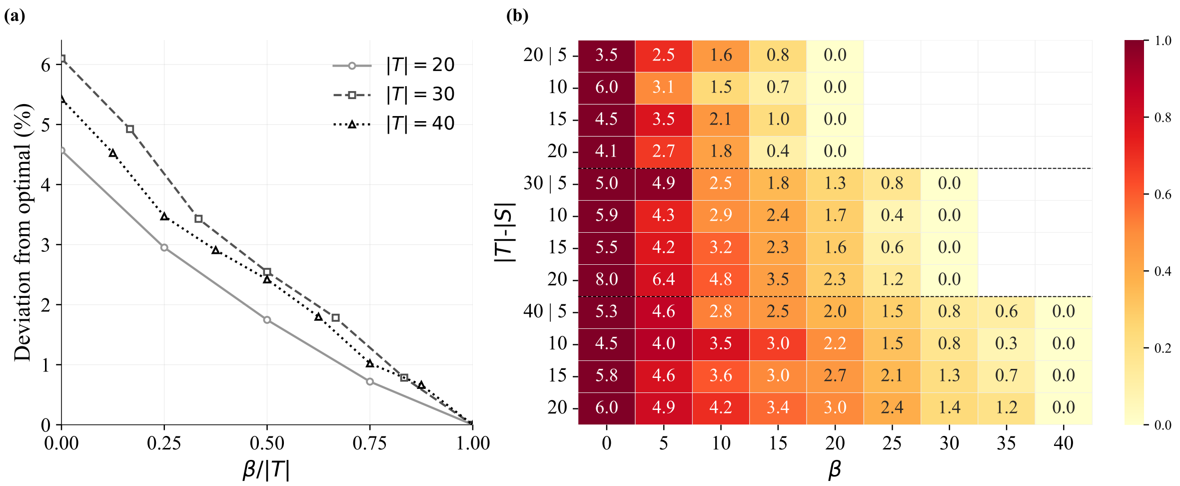


Figure 4.4. Effect of β on solution quality. (a) Deviation from optimality as a function of $\beta/|T|$ for different values of $|T|$. (b) Deviation from optimality as a function of β for different combinations of $|T|$ and $|S|$.

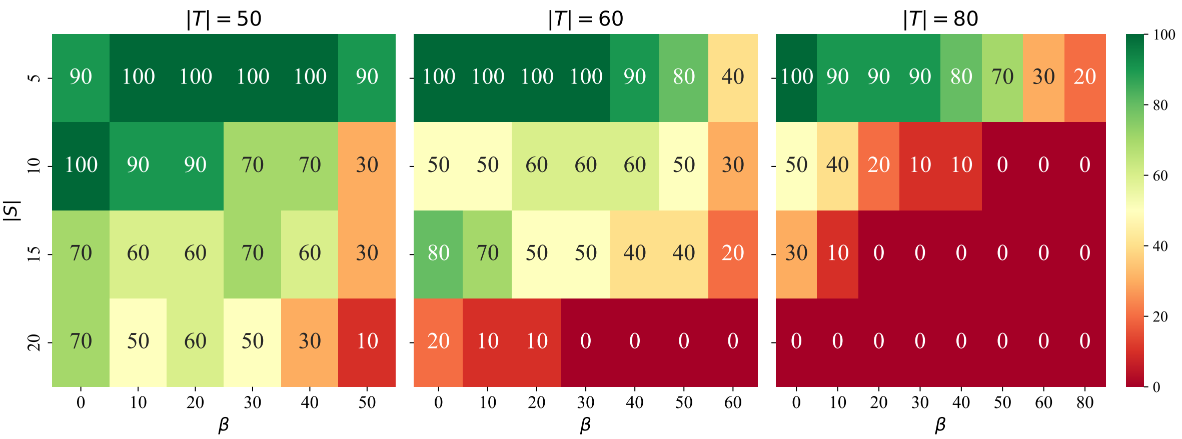


Figure 4.5. Success rate of the heuristic.

compared to only 20% for the exact algorithm. As $|S|$ increases, both the exact and heuristic approaches face greater difficulty. The heuristic nonetheless enables scaling along the time dimension, which is the most relevant parameter from a practical standpoint.

4.7.3 Robustness Analysis

We now investigate how the uncertainty parameters H and K affect the purchaising choices and provide suggestions on the policies to be adopted. We recall that H represents the maximum number of late orders across the entire planning horizon, while K controls the maximum delay difference between consecutive orders from the same supplier. For this analysis, we restrict our attention to a subset of instances with $|T| = 20$ and $|S| \in \{5, 10\}$, and systematically vary $H \in \{1, 2, \dots, 10, 30\}$ and $K \in \{1, 2, 3, 4, 5, 20\}$. For each (H, K) configuration, 20 instances (10 per value of $|S|$) are solved to optimality. All results are compared against the *deterministic case*, which serves as a benchmark for the cost components and solution structure. With *deterministic case* we refer to the solution of the problem without uncertainty, i.e., assuming that all orders are delivered on time (at ℓ^s).

Parameter H Parameter H controls the maximum number of late orders over the entire planning horizon. As H increases, a wider range of delay scenarios must be considered. Figures 4.6 and 4.7 provide a graphical representation of the effect of H on cost components and solution structure. The deterministic case is included as a reference point, indicated by the dashed horizontal lines in the plots. For this analysis, to isolate the effect of H we fix $K = 20$, thus removing the correlation constraint.

The behavior is qualitatively identical for $|S| = 5$ and $|S| = 10$: while the absolute volumes differ due to the different number of suppliers, the trends remain nearly invariant. This suggests that the effect of H on solution structure is driven by the nature of the uncertainty rather than by the size of the supplier pool. The detailed numerical results, disaggregated by $|S|$, are reported in the Appendix.

Figure 4.6(a) shows that total cost increases monotonically with H, but with diminishing returns, indicating that the first few units of additional uncertainty have the major impact on solution cost, while further increases yield progressively smaller effects. The increase is driven primarily by backlog costs; nevertheless, purchasing costs also rise, meaning that hedging against uncertainty forces the model to deviate from the cost-optimal plan by using more expensive suppliers, less favorable times, or suboptimal quantities.
Figure 4.7 shows how the structural characteristics of the solution—number of orders placed during the planning horizon, number of suppliers from which orders are placed, and number of periods in which orders are placed, and finally, the number of late orders— evolve with H. The solution structure reveals two distinct regimes. For $H \leq 4$, the model diversifies: for $|S| = 10$ orders rise from 6.5 to 8.5, active suppliers from 2.4 to 3.2. This diversification strategy—placing more, smaller orders from a wider set of suppliers—reduces the exposure to any single supplier's delays. Note that while the number of ordering periods increases, the growth is less pronounced than that of total orders, indicating that some of the additional orders are placed in the same periods already identified as cost-effective by the model. For $H > 5$, a reversal occurs: orders drop to 4 (below the deterministic level of 4.3), suppliers to 2.2, and ordering periods to 3.8. We move from a *diversification* strategy to a *consolidation*

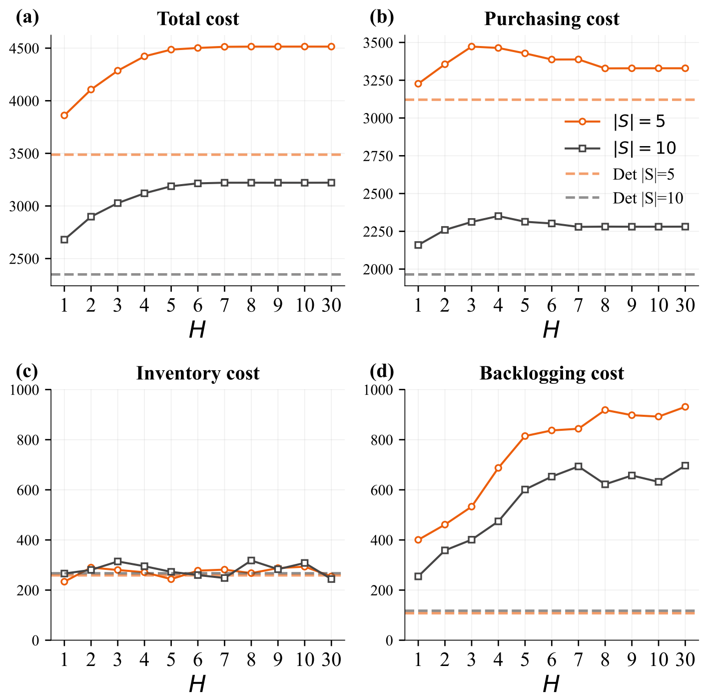


Figure 4.6. Effect of H on cost components. Dashed horizontal lines indicate the values for the deterministic case as a benchmark for each value of $|S|$.

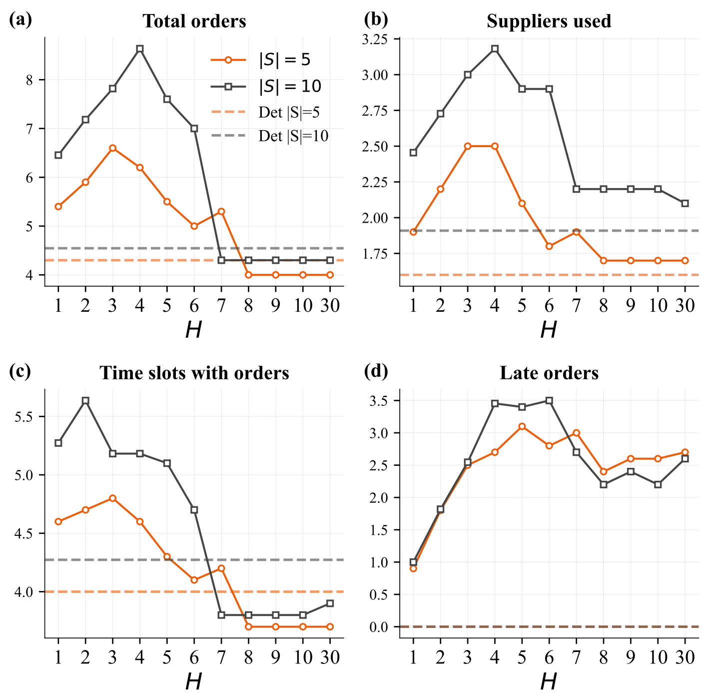


Figure 4.7. Effect of H on solution structure.

strategy, where fewer, larger orders are placed from a smaller set of suppliers. This could seem counter-intuitive at first, but the behavior admits an explanation based on the nature of the uncertainty and the model's response to it. When H is large, a large fraction of the orders can be delayed: diversification no longer provides effective protection, because even orders spread across many suppliers can all be simultaneously delayed. The model therefore switches to a different strategy: placing fewer, larger orders to save on fixed ordering costs, prioritizing cost efficiency over risk mitigation. Figure 4.7(d) shows the number of late orders as a function of H. The number of late orders increases with H, reaching a maximum of 3.5 at $H = 6$, before declining slightly to 2.1 for $H \geq 8$. This peak coincides with the transition between the two regimes described above. For $H \leq 6$, the model still places many orders (thus exposing more orders to potential delays), so the number of late deliveries grows. Nevertheless, the number of late orders remains relatively low with respect to the total number of orders. This confirms one of the greater difficulties of this problem: the worst-case scenarios are not those in which all orders are delayed, but rather those in which a small number of strategically chosen orders are delayed, which makes the problem more complex and less intuitive. For $H = 30$ (all orders can be delayed), the results are nearly identical to $H = 10$, indicating saturation beyond $H \approx 8$.

Parameter K Parameter K controls the maximum absolute difference in delay between consecutive orders from the same supplier. As K increases, the delay patterns can become more diverse and less predictable across consecutive orders from the same supplier, thus increasing the complexity of the problem. As before, the deterministic case is included as a reference

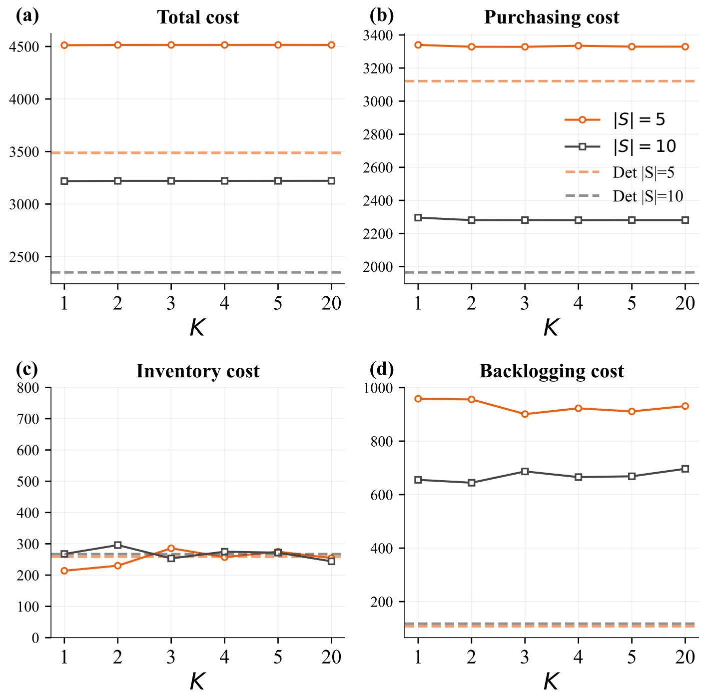


Figure 4.8. Effect of K on cost components. Dashed horizontal lines indicate the deterministic benchmark for each value of $|S|$.

point, indicated by the dashed horizontal lines in the plots. For this analysis, to isolate the effect of K we fix $H = 30$, thus removing the budget constraint. Figure **??** and Figure **??**

show the effect of K on cost components and solution structure, respectively. The effect of K on cost is more limited than that of H, with total cost increasing by about 5% as K increases from 1 to 5, and then stabilizing, while the effect on solution structure is even more limited, with the number of orders, suppliers, and ordering periods remaining relatively stable across different values of K. Thus, what emerges is that, when H is unconstrained, K has a negligible effect on both cost and solution structure. This shows that when all the orders can be delayed, the temporal correlation constraint becomes redundant.

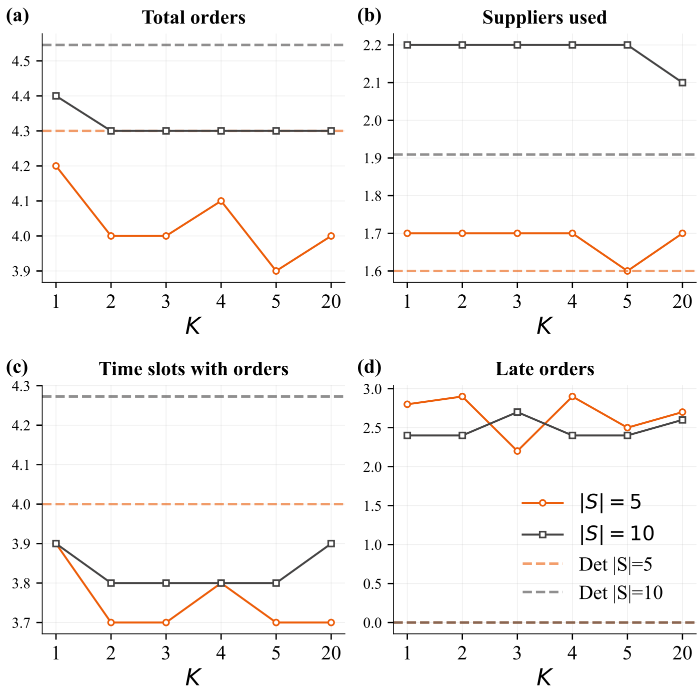


Figure 4.9. Effect of K on solution structure.

Joint effect of H and K After analyzing the effects of H and K separately, we now examine their joint impact on cost and solution structure. Table 4.4 reports the percentage cost increase over the deterministic baseline for each (H, K) pair. The table reveals a clear saturation pattern.. In the low-uncertainty region ($H \leq 3$, $K \leq 2$), costs increase steeply with both parameters, with the cost increase ranging from 4.1% to 19.1%. In the high-uncertainty region ($H \geq 7$, $K \geq 3$), the cost increase saturates at approximately 32.8%, and further increases in either H or K have negligible effect. This saturation corresponds to a situation where the uncertainty set is large enough that the worst-case delay pattern is already close to the unconstrained maximum.

Figures 4.10 and 4.11 provide a detailed view of the combined effect of H and K on cost components and solution structure, respectively. In both figures, the x-axis enumerates all (H, K) configurations with K nested within each value of H, and the deterministic baseline is reported as the leftmost point. The interaction between H and K is critical: for small to moderate values of H, the parameter K becomes a major driver of the cost of robustness. Notably, the temporal correlation of delays can have a significant impact on the cost of robustness, contrary to what emerged in the previous paragraph, for the case where H is unconstrained.

$K\backslash H$	1	2	3	4	5	6	7	8	9	10
1	4.1	9.9	13.8	17.8	20.4	24.7	26.5	27.9	28.9	29.5
2	7.4	13.9	19.1	23.3	27.2	29.7	31.0	31.9	32.2	32.5
3	9.9	17.4	22.6	26.9	29.7	31.4	32.3	32.6	32.7	32.8
4	11.2	19.2	24.7	28.4	31.0	32.2	32.6	32.7	32.8	32.8
5	11.9	20.0	25.4	29.2	31.4	32.4	32.7	32.8	32.8	32.8

Table 4.4. Percentage increase in total cost over the deterministic solution.

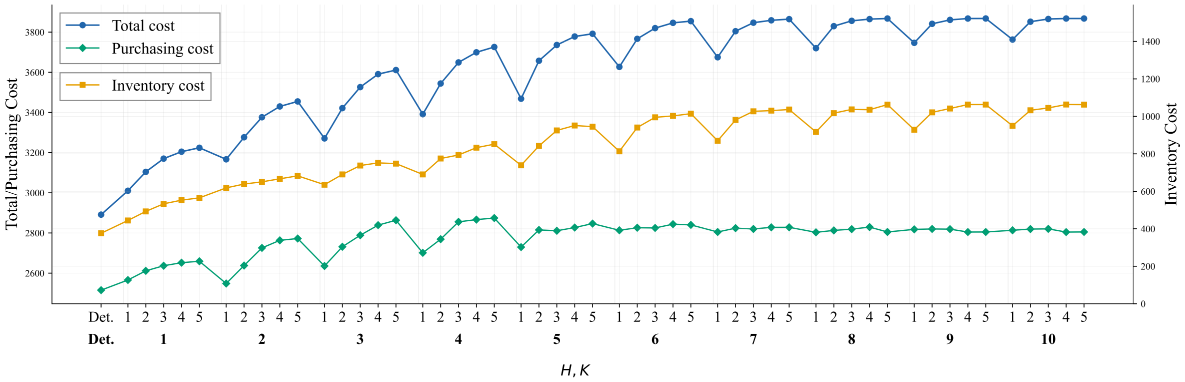


Figure 4.10. Effect of (H, K) on cost components.

Within each H group, for moderate values of H, increasing K produces a substantial cost increase. This increase is driven especially by purchasing costs and backlogging costs. In particular, for $H < 5$, the purchasing cost is the main driver of the total cost increase, while for $H \geq 5$, the backlogging cost becomes more significant. This suggests that relaxing the temporal correlation constraint forces the model to alter its procurement plan (ordering more often, from more expensive suppliers or at less favorable times) to hedge against more diverse delay patterns. Nevertheless, when the number of potential late orders is large, the temporal correlation constraint becomes less relevant, as the adversary already has enough flexibility to construct worst-case scenarios regardless of K.

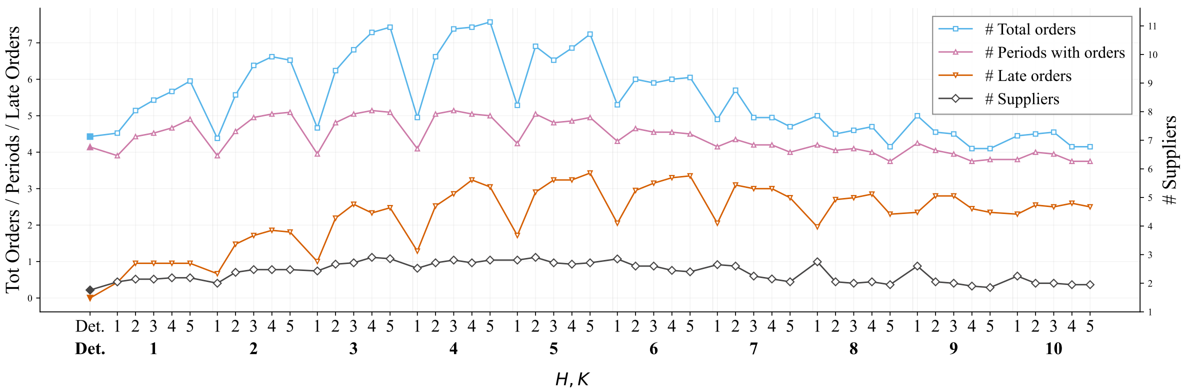


Figure 4.11. Effect of (H, K) on solution structure.

Figure 4.11 showing the joint effect on solution structure, confirms the previous insights. For moderate values of H ($3 \leq H \leq 5$), increasing K strongly increases the number of orders: at $H = 3$, orders rise from 4.7 ($K = 1$) to 7.4 ($K = 5$), a span of 2.8, comparable to the full range across all H values for fixed K. For $H \geq 7$, the effect of K on solution structure becomes negligible: the model has already shifted to the consolidation regime regardless of K. Regarding the number of suppliers, the solutions consistently select between 2.0 and 3.3 suppliers on average, even when $|S| = 10$ are available (the maximum in any single instance is 5). This implies that scalability with respect to $|S|$ is less critical in practice, as the model naturally selects a small subset of suppliers. Finally, both figures exhibit a clear saturation effect: in the high-uncertainty region ($H \geq 7$, $K \geq 3$), cost and structural metrics stabilize, indicating that the uncertainty set has become large enough for the worst-case delay pattern to be effectively unconstrained.

4.7.4 Managerial Insights

The computational results presented in the previous sections yield several actionable insights for practitioners involved in robust procurement planning under supplier uncertainty.

The choice of optimization method should be driven primarily by the length of the planning horizon $|T|$ and the size of the supplier pool $|S|$. For horizons up to 40 periods, the exact algorithm reliably finds optimal solutions within acceptable runtimes, and decision-makers can confidently rely on it. Beyond this threshold, tractability deteriorates sharply, and the heuristic becomes the method of choice. Importantly, the heuristic offers a flexible lever: by tuning the parameter β, practitioners can control how many periods are solved to optimality before switching to robust approximations. Setting β to cover roughly half the planning horizon already captures a large share of the exact solution's quality, and smaller values of β can be used when computational resources are limited. The size of the supplier pool $|S|$ also affects tractability: instances with a large number of candidate suppliers can be challenging even for moderate horizons, and practitioners should be mindful of this when specifying the input.

A key observation from the computational experiments is that the optimal solution almost never activates more than five suppliers simultaneously, regardless of how large $|S|$ is. This structural regularity suggests a natural two-phase workflow for practitioners. In the *first phase*, the exact algorithm is run on an instance with the full candidate set (e.g., $|S| = 20$) and a sufficiently long horizon (but still tractable, e.g., $|T| = 40$), with the goal of identifying which suppliers are selected most frequently across the optimal solution. This step is run only once, producing a small preferred subset of suppliers. In the *second phase*, the heuristic is applied with this restricted supplier set and a long planning horizon (up to $|T| = 80$), allowing to solve instances that would be intractable with the exact method while still capturing the most relevant suppliers identified in the first phase. This two-phase approach leverages the strengths of both methods and provides a practical pathway to robust procurement planning in real-world settings with long horizons.

The robustness analysis reveals that the model responds to increasing uncertainty through two qualitatively different strategies, depending on the regime of H. In the *low-uncertainty regime* (small H), the model adopts a *diversification* strategy: orders are spread across multiple suppliers and multiple periods so that if one delivery is delayed, alternative sources and orders can compensate, reducing the impact of any single disruption. As H increases and the worst-case scenario becomes more severe, diversification alone is no longer sufficient: splitting orders across many suppliers exposes more deliveries to potential delays, increasing vulnerability. The model therefore transitions to a *consolidation* strategy, concentrating orders on a smaller number of highly reliable suppliers and adjusting order timing to build safety buffers. The idea is that, if risk cannot be mitigated after all, it is better to focus on cost efficiency and save money on purchasing and ordering costs.

4.8 Conclusions

This chapter studied the Multiple Suppliers Purchase Planning problem under lead time uncertainty, formulated as a two-stage robust optimization problem. In this setting, a buyer must decide which suppliers to activate and how much to order from each, before the actual lead times are revealed, so as to minimize the worst-case total cost—comprising purchase prices, fixed ordering fees, inventory holding, and backlogging penalties—over all realizations of delays consistent with a prescribed uncertainty set. We design of a structured uncertainty set Γ that captures two practically motivated properties of lead time variability: a budget H bounding the total number of late deliveries across the entire planning horizon, and a correlation parameter K limiting the maximum absolute difference in delay between consecutive orders from the same supplier. Unlike continuous uncertainty sets used in prior work, Γ contains integer variables, reflecting the requirement that each order arrives in its entirety at a single point in time. We studied the polyhedral structure of the single-supplier projection Γ^s, identifying two families of valid inequalities that are facet-defining for Γ^s and can be used to strengthen the formulation. We also derived some polyhedral properties and found the complete description of the convex hull of Γ^s. We then developed an exact Benders-like branch-and-cut algorithm to solve the two-stage robust model and a horizon-split heuristic to approximately solve larger instances that are intractable for the exact approach. We observe a sharp tractability transition for the exact algorithm around 50 periods, while the heuristic enables the solution of instances with up to 80 periods. Finally, we conducted a comprehensive computational study to evaluate the performance of the proposed algorithms and to analyze the model's robustness properties and managerial implications. In particular, we identify two distinct regimes of behavior as the uncertainty parameters H, i.e., the maximum total number of late deliveries, and K, i.e., the maximum absolute difference in delay between consecutive orders from the same supplier, vary: a diversification strategy that spreads orders across many suppliers and periods when uncertainty is limited, and a consolidation strategy that concentrates orders on fewer suppliers when uncertainty is pervasive. We also observe strong saturation effects, indicating that the model's behavior stabilizes beyond certain thresholds of H and K, which has practical implications for parameter calibration. Also, we find that the cost of robustness—i.e., the increase in total cost compared to a deterministic benchmark with no uncertainty—can be significant, especially when H is large, highlighting the importance of accounting for lead time variability in procurement planning. Finally, we propose a two-phase decision-making framework to guide practitioners in solving larger instances. We suggest using the exact algorithm to solve a smaller instance with a reduced number of periods, but a larger set of suppliers, to choose the most promising subset of suppliers to consider in the full-size instances. As a second step, we recommend using the heuristic approach to solve a full-size instance with larger number of periods, considering the selected subset of suppliers. The present work focuses on a single-item, uncapacitated setting with backlogging allowed and lead time as the sole source of uncertainty. A natural direction for future research is the multi-item generalization, where demand interactions across products—shared suppliers, joint ordering economies, or substitution effects—would enrich both the modeling and the algorithmic challenges. On the uncertainty modeling side, it would be interesting to consider joint uncertainty in both lead times and demand, which would require a more intricate uncertainty set and a three-stage or rolling-horizon formulation. Finally, a data-driven approach to calibrate the uncertainty set parameters H and K from historical delay records could strengthen the connection between the theoretical model and real-world procurement practice.

Chapter 5

Modeling Network Congestion under Demand Uncertainty Using Wardrop Principles [1]

5.1 Introduction

Congestion is one of the central challenges in urban transportation, leading to increased travel times, fuel consumption, and emissions. To mitigate these effects, effective traffic management requires anticipating how travelers respond to changing network conditions, especially variations in travel demand arising from, e.g., daily fluctuations, seasonal trends, or unforeseen events. Because such variations are inevitable and often difficult to predict accurately, understanding the potential worst-case congestion in a traffic network is crucial.

In this chapter, we study the problem of determining the worst-case congestion in a multi-commodity traffic network subject to demand uncertainty. Effectively, we stress-test a given network by identifying demand realizations and corresponding travelers' route choices that maximize congestion. To this end, we propose a novel bilevel formulation in which a traffic planner acts as the leader and the users of the traffic network act as the followers. In the leader's problem, we explicitly model the uncertain travel demand against which the traffic planner wants to hedge. For this purpose, we exploit ideas from robust optimization [129, 20, 24] so that demand realizations are drawn from a predefined uncertainty set. We use both ellipsoidal and polyhedral uncertainty sets, including the hose polyhedron [59, 57] and the budgeted uncertainty set [25, 26, 127]. To model the travelers' route choices, we consider the two Wardrop principles—the user equilibrium and the system optimum [141, 140]. In a user equilibrium (UE), travelers select routes that minimize their individual travel costs so that no traveler can improve their travel time by unilaterally changing routes. In contrast, under the system optimum (SO), a central planner coordinates or assigns traffic such as to minimize the total travel time across the network. Overall, we thus consider a single-leader multi-follower bilevel problem in which Wardrop principles govern the followers' decisions. An accessible overview of the two Wardrop principles can, e.g., be found in [50]. Moreover, we refer to [52, 13, 53], and the surveys in [48, 47, 77] for a general overview of bilevel optimization. Let us further mention that bilevel problems are notoriously hard to solve. [70] have shown that even linear

[1] The work presented in this chapter has been developed in collaboration with Yasmine Beck (ESSEC Business School), Ivana Ljubić (ESSEC Business School), and Sara Mattia (IASI-CNR).

bilevel problems, i.e., those with continuous variables, linear objective functions, and linear constraints, are strongly NP-hard in general.

Building on the modeling choices described above and acknowledging the computational challenges of bilevel optimization, the congestion models studied in this work are reformulated as mixed-integer nonlinear problems that exploit binary variables and big-M constants, which can be tackled using state-of-the-art general-purpose solvers. We prove the existence of optimal solutions to these problems, derive valid big-Ms, and propose several enhancement techniques that exploit the network topology and structural properties of optimal flows to further strengthen the formulations. To measure congestion, we consider three different functions: the maximum utilization function, which focuses on the most congested arc; the total utilization function, which accounts for congestion on all arcs of the network; and the Bureau of Public Roads (BPR) function, which penalizes arcs with very high congestion levels. To compare the different variants of the congestion models and to assess the effectiveness of the proposed enhancement techniques, we conduct an extensive computational study on instances of the Sioux Falls network [84] and instances from the SNDlib [113, 112]. We analyze both user equilibrium and system optimum formulations in terms of congestion levels and computational performance. Moreover, we compare different problem variants arising from the choice of congestion measures and uncertainty models, as well as from different travel cost functions, including linear, quadratic, and BPR functions. Our computational study reveals several key insights. First, the proposed enhancement techniques significantly improve computational performance, enabling the solution of larger instances that would otherwise remain unsolved. Second, the choice of the travel cost function has a greater impact on the computational tractability of the congestion models than the choice of the congestion measure. Third, the UE formulation can be solved considerably faster than the SO formulation, while congestion levels under the UE are typically close to those under the SO. Fourth and finally, we observe a trade-off between worst-case congestion and computational cost across different uncertainty models, as congestion estimates over larger uncertainty sets are typically obtained at increased computational effort.

Related Literature

Many applications of bilevel optimization arise in the context of transportation; see, e.g., [102] for a general overview. In particular, bilevel models have been studied in network design [83, 94, 19, 60, 121], network pricing [81, 36, 75, 54, 56, 16], or pricing with routing [40]. Among these contributions, however, only a few works consider bilevel formulations in which the lower level is modeled using Wardrop principles. To the best of our knowledge, [16, 54, 121] are the closest related studies in this context. In [121], the authors study the discrete network design problem in which travelers' act according to the Wardrop user equilibrium principle. To solve the resulting bilevel model, the authors present a branch-and-price-and-cut approach. In [54], the authors consider a bilevel toll-setting problem in which travelers again follow the Wardrop user equilibrium principle. In their work, the authors focus primarily on the theoretical properties of the model, but a closely related toll-setting problem is studied in [16] from a computational point of view. In addition, [16] incorporate robust Wardrop equilibria to model travelers' route choices subject to uncertain travel costs. Overall, our work thus differs from the above contributions in two main aspects. First, while [121] address network design and [54, 16] focus on network pricing, we study the problem of determining the worst-case congestion that can arise in a given traffic network. Second, we explicitly account for uncertainty in the travel demand, whereas [16] consider uncertainty in the travel costs and [54, 121] study deterministic settings. To the best of our knowledge, no existing work considers the problem of stress-testing a traffic network under uncertain travel demand using a bilevel formulation with Wardrop principles governing the travelers' behavior. Although we focus on existing traffic networks, let us mention that the insights provided by our models may also be useful in network design contexts, for example to identify which parts of a network are most susceptible to congestion under uncertain travel demand.

Outline

The remainder of this chapter is organized as follows. In Section 5.2, we define the congestion models under the user equilibrium and the system optimum. In Section 5.3, we present mixed-integer nonlinear reformulations of these problems that exploit binary variables and big-M constants and prove the existence of valid big-Ms and optimal solutions. In Section 5.4, we propose several enhancement techniques to strengthen the formulations of the congestion models. In Section 5.5, we discuss further practical modeling techniques and elaborate on our specific choice of congestion measures, travel cost functions, and uncertainty models. In Section 5.6, we describe the experimental setup and the design of the computational study. The results of this study are discussed in Section 5.7. Finally, we conclude in Section 5.8.

5.2 Problem Statement

We study the problem of determining the worst-case congestion in a traffic network under uncertain travel demand. To this end, we first introduce the network model and discuss how to account for congestion and demand uncertainty in Section 5.2.1. Afterward, in Section 5.2.2, we elaborate on the behavior of the network users, who are assumed to act according to one of the two Wardrop principles—the user equilibrium or the system optimum.

5.2.1 Congestion Model

To model the multi-commodity traffic network, we consider a directed graph $G = (V, A)$ with vertex set V and arc set $A \subseteq V \times V$. Moreover, vertex subsets $S \subseteq V$ and $T \subseteq V$ denote the sets of origin and destination vertices, respectively. The set of all commodities to be routed through the network is given by $K \subseteq S \times T$ and each commodity $k \in K$ has a fixed demand $d_k \in \mathbb{R}_{\geq 0}$ to travel from its origin to its destination. For the ease of presentation, we consider a single commodity for each origin-destination (OD) pair and all other pairs of vertices are assumed to have zero demand. For the remainder of this chapter, we make the following connectivity assumption, which is standard in the transportation literature; see, e.g., Assumption 2.A in [115].

Assumption 3. *For every commodity $k = (s_k, t_k) \in K$, there exists at least one dipath that connects s_k and t_k in G.*

Let $f = (f_a)_{a \in A} \in \mathbb{R}^{|A|}$ denote the vector of all arc flows used to model traffic. To assess the network under adverse conditions, we consider the worst-possible realization of the travel demand and the corresponding travelers' route choices. This leads us to considering the problem

$$\max_{d,f} \quad L(f) \quad \text{s.t.} \quad d \in \mathcal{U},\ f \in F(d). \tag{5.1}$$

Here, we assume that the travel demand $d = (d_k)_{k \in K}$ is not known exactly but that it takes values within a given uncertainty set $\mathcal{U}$. Moreover, we use $F(d)$ to denote the set of traffic flows corresponding to the optimal travelers' responses for a given demand realization d. For the uncertainty set, we impose the following throughout the remainder of this chapter.

Assumption 4. *The set of origin-destination pairs K is fixed across all realizations of the uncertain demand. The uncertainty set $\mathcal{U}$ is non-empty, convex, compact, and contained in the nonnegative orthant $\mathbb{R}^{|K|}_{\geq 0}$.*

We assume that the uncertainty set $\mathcal{U}$ is non-empty and compact to ensure the existence of optimal solutions to the congestion models. Moreover, we assume convexity to obtain formulations that are more computationally tractable and amenable to state-of-the-art general-purpose solvers. In Assumption 4, we further require that $\mathcal{U}$ is contained in the nonnegative orthant and that the set of commodities is fixed, which guarantees the existence of optimal flows for all demand realizations. Note that, if the travel demand is perfectly known, i.e., if the uncertainty set $\mathcal{U}$ is a singleton, Problem (5.1) reduces to a deterministic traffic assignment problem aimed at determining the worst-case congestion. Throughout this chapter, we refer to this setting as the *deterministic case* and use it as a baseline for assessing the impact of demand uncertainty on network congestion.
For the latency function $L(f)$ used to measure congestion, we make the following assumption.

Assumption 5. *The latency function $L : \mathbb{R}^{|A|}_{\geq 0} \to \mathbb{R}_{\geq 0}$, $f \mapsto L(f)$ is continuous.*

Next, we elaborate on the Wardrop principles used to model traffic flow for a given $d \in \mathcal{U}$.

5.2.2 Wardrop Principles

For computational tractability, we adopt a fully aggregated node-arc formulation in which all commodities with the same origin are treated as a unique commodity with a single source and multiple destinations. In the transportation literature, it is well known that such an aggregation preserves the set of optimal flows, provided there are no commodity-dependent costs; see, e.g., Section 5.2.3 in [33], [28], and [43] in which similar settings are studied. In what follows, we model aggregated commodity flows using variables $x = (x^s)_{s \in S}$ with $x^s = (x^s_a)_{a \in A} \in \mathbb{R}^{|A|}_{\geq 0}$ for all $s \in S$. The overall arc flows are then given by

$$f = \sum_{s \in S} x^s. \tag{5.2}$$

Moreover, for every source $s \in S$, we define the set of commodities that originate in s as

$$K_s := \{k = (s_k, t_k) \in K \colon s_k = s\}.$$

In what follows, we use $\delta^{\text{in}}(v)$ and $\delta^{\text{out}}(v)$ to denote the sets of in- and outgoing arcs of vertex $v \in V$, respectively. Flow conservation can then be stated as

$$\sum_{a \in \delta^{\text{out}}(v)} x^s_a - \sum_{a \in \delta^{\text{in}}(v)} x^s_a = d^s_v, \quad v \in V,\ s \in S, \tag{5.3}$$

with

$$d^s_v = \begin{cases} +\sum_{k \in K_s} d_k, & v = s, \\ -d_k, & k = (s, v) \in K_s, \\ 0, & \text{otherwise.} \end{cases}$$

With the constraints in (5.2) and (5.3) at hand, which characterize feasible flows in the traffic network, we now extend the model to incorporate travel costs. These costs determine how users choose their routes and, consequently, how flows distribute across the network. For each arc $a \in A$, we define a travel cost function

$$c_a : \mathbb{R}^{|A|}_{\geq 0} \to \mathbb{R}_{>0}, \quad f \mapsto c_a(f_a),$$

which captures the time required to traverse the arc as well as additional expenses such as fuel consumption or congestion-induced delays. In line with standard terminology, we use the terms travel time and travel cost interchangeably. Moreover, the following will be a standing assumption for the remainder of this chapter.

Assumption 6. *For every arc $a \in A$, the function $c_a : \mathbb{R}^{|A|}_{\geq 0} \to \mathbb{R}_{>0}$, $f \mapsto c_a(f_a)$ is convex, non-decreasing, and continuously differentiable in f.*

In particular, we assume that travel costs are separable, i.e., the cost of traversing an arc $a \in A$ only depends on the flow on that arc but not on the flows on other arcs of the network. Assumption 6 is satisfied by many travel cost functions studied in the literature such as, e.g., the BPR function; see also Section 1.5 in [115] for further discussions. Let us now formalize the behavior of users in the traffic network. To this end, we follow the Wardrop principles [141, 140] and distinguish between the user equilibrium and the system optimum. For both settings, we assume that travelers have complete knowledge of available paths and that traffic flows remain stable over time.

User Equilibrium

In the user equilibrium, travelers act selfishly by selecting their routes such as to minimize their own travel times. This behavior leads to a stable state in which no individual user can reduce their travel costs by unilaterally changing routes. Consequently, in a Wardrop user equilibrium, all paths used by a given commodity have equal travel costs. For separable travel cost functions, the UE can be obtained as an optimal solution to the problem

$$\min_{f,x \geq 0} \quad \sum_{a \in A} \int_0^{f_a} c_a(\xi) \, \mathrm{d}\xi \quad \text{s.t.} \quad (5.2), (5.3). \tag{5.4}$$

Hence, under the user equilibrium, the set $F(d)$ corresponds to the set of optimal solutions to Problem (5.4). A similar problem is, e.g., considered in Section 3.6.2 in [58].

System Optimum

The system optimum seeks to minimize the total travel time across the network by coordinating the flow of all users. This corresponds to a situation in which the marginal travel times are the same on all routes used by a given commodity. Otherwise, shifting flow from a higher- to a lower-cost route would reduce the overall travel time. The SO can be obtained as an optimal solution to the problem

$$\min_{f,x \geq 0} \quad \sum_{a \in A} c_a(f_a) f_a \quad \text{s.t.} \quad (5.2), (5.3). \tag{5.5}$$

Hence, under the system optimum, the set $F(d)$ corresponds to the set of optimal solutions to Problem (5.5). For further discussion of the SO, see also Section 2.4 in [115].

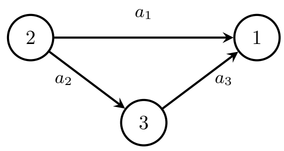


Figure 5.1. The traffic network considered in Example 5.

To sum up, for each demand realization $d \in \mathcal{U}$, both the UE and the SO can be obtained by solving appropriately chosen nonlinear optimization problems (NLPs). The two models share the same linear feasibility constraints, i.e., the only nonlinearities arise in the respective objective functions. As a result, the Abadie constraint qualification is satisfied for all feasible points. Moreover, under Assumption 6, Problems (5.4) and (5.5) are convex NLPs, so that the Karush–Kuhn–Tucker (KKT) conditions are both necessary and sufficient optimality conditions. Finally, let us mention that the main difference between the UE and SO models lies in their behavioral assumptions. The UE reflects decentralized, selfish route choices by individual travelers, whereas the SO represents a centralized assignment of traffic flows that minimizes total travel time.

5.2.3 Congestion Ratio

To compare the congestion levels under the user equilibrium and the system optimum, we introduce the so-called congestion ratio (CR), which we define as

$$\mathrm{CR} = \frac{L(f^{\mathrm{UE}})}{L(f^{\mathrm{SO}})}.$$

Here, f^{UE} and f^{SO} denote optimal arc flows in the UE and the SO formulation, respectively. A congestion ratio of 1 indicates that the user equilibrium solution achieves the same overall congestion as the system optimum. Values greater than 1 indicate higher congestion under the UE, whereas values lower than 1 indicate higher congestion under the SO.
Note that the congestion ratio differs from the price of anarchy [79], a widely used metric in the transportation literature for comparing UE and SO solutions. The price of anarchy measures the efficiency loss caused by selfish routing by comparing aggregated travel costs under a given demand realization and is, by definition, always at least 1. In contrast, the CR focuses on congestion rather than total travel cost and may compare UE and SO flows corresponding to different demand realizations in our framework. As a result, the CR captures worst-case congestion levels and, unlike the price of anarchy, can take values below 1. We illustrate such a behavior with the following example.

Example 5 (CR < 1)**.** *We consider the network depicted in Figure 5.1 with a single commodity $k = (2, 1)$ whose demand is uncertain and takes values in the interval $[30, 50]$. For all arcs $a \in A = \{a_1, a_2, a_3\}$, we use the travel cost function $c_a(f_a) = 1 + 0.15(f_a/20)^4$ and measure congestion as $L(f) = \sum_{a \in A} f_a$, which captures the overall traffic load. In this network, there are two paths from the origin vertex 2 to the destination vertex 1, namely $2 \to 1$ and $2 \to 3 \to 1$.*
Under the UE formulation, the worst-case demand is $d^{UE} = 50$ with optimal flows $f_{a_1}^{UE} = 33.2$ and $f_{a_2}^{UE} = f_{a_3}^{UE} = 16.8$, which results in a congestion value of $L(f^{UE}) = 3.34$. Under the SO

formulation, the worst-case demand is still $d^{SO} = 50$ *but the optimal flows are* $f_{a_1}^{SO} = 28.4$ *and* $f_{a_2}^{SO} = f_{a_3}^{SO} = 21.6$*, which yields* $L(f^{SO}) = 3.58$*. Hence, the congestion ratio is*

$$CR = \frac{L(f^{UE})}{L(f^{SO})} = \frac{3.34}{3.58} = 0.93 < 1,$$

which shows that the UE solution can achieve lower worst-case congestion than the SO solution, a behavior that cannot occur with the price of anarchy. Note that, in this example, both formulations yield the same worst-case demand realization, but this does not have to be the case in general.

5.3 Single-Level Reformulations

Because of its nested structure, the congestion model (5.1) is intrinsically hard to solve. Motivated by common approaches in bilevel optimization, we thus reformulate it as a single-level problem, which can then be tackled using state-of-the-art general-purpose solvers. Under Assumption 6, the lower-level problems (5.4) and (5.5) are convex d-parameterized NLPs, which allows to use their KKT conditions to obtain equivalent single-level reformulations. We present the single-level reformulations of the congestion models under the user equilibrium and the system optimum in Sections 5.3.1 and 5.3.2, respectively. As the resulting problems are mathematical problems with equilibrium constraints (MPECs)—see, e.g., [89] for a general overview—we further reformulate them as mixed-integer nonlinear problems (MINLPs), following the same ideas as in [61]. To this end, we introduce auxiliary binary variables and sufficiently large big-M constants, which we derive in Section 5.3.3. Finally, we prove the existence of optimal solutions to the congestion models in Section 5.3.4.

5.3.1 Congestion Model under the User Equilibrium

For a given demand realization $d \in \mathcal{U}$, the KKT conditions of Problem (5.4) can be stated as

$$f = \sum_{s \in S} x^s, \tag{5.6a}$$

$$\sum_{a \in \delta^{\text{out}}(v)} x_a^s - \sum_{a \in \delta^{\text{in}}(v)} x_a^s = d_v^s, \qquad v \in V,\ s \in S, \tag{5.6b}$$

$$0 \le c_a(f_a) + \tau_i^s - \tau_j^s \perp x_a^s \ge 0, \qquad a = (i,j) \in A,\ s \in S. \tag{5.6c}$$

Here, the variables $\tau = (\tau^s)_{s \in S}$ with $\tau^s = (\tau_v^s)_{v \in V} \in \mathbb{R}^{|V|}$ correspond to the Lagrangian multipliers associated with the flow conservation constraints in (5.3). Here the notation $0 \le a \perp b \ge 0$ is used for the complementarity condition meaning that $a \ge 0$, $b \ge 0$, and $a \cdot b = 0$. Because the KKT conditions are necessary and sufficient for Problem (5.4), we can replace Problem (5.4) with System (5.6) to obtain the single-level reformulation of the congestion model (5.1) given by

$$\max_{d,f,x,\tau} \quad L(f) \quad \text{s.t.} \quad d \in \mathcal{U},\ (5.6a)\text{–}(5.6c). \tag{5.7}$$

Problem (5.7) is an MPEC because of the complementary constraints in (5.6c). Nevertheless, we can exploit the disjunctive nature of these constraints to derive an MINLP reformulation of the overall congestion model. To this end, we introduce auxiliary binary variables $y_a^s \in \{0,1\}$

and sufficiently large constants $M_a^s, N_a^s \in \mathbb{R}_{\geq 0}$ for all arcs $a \in A$ and all origin vertices $s \in S$ so that we can equivalently re-write the constraints in (5.6c) as

$$0 \leq c_a(f_a) + \tau_i^s - \tau_j^s \leq M_a^s(1 - y_a^s), \qquad a = (i,j) \in A,\ s \in S, \tag{5.8a}$$
$$0 \leq x_a^s \leq N_a^s y_a^s, \qquad a \in A,\ s \in S. \tag{5.8b}$$

For $a \in A$ and $s \in S$, the binary variable y_a^s indicates whether any flow originating in s uses arc a. The MINLP reformulation of Problem (5.1) then reads

$$\max_{d,f,x,\tau,y} \quad L(f) \quad \text{s.t.} \quad d \in \mathcal{U},\ (5.6a),\ (5.6b),\ (5.8),\ y_a^s \in \{0,1\},\ a \in A,\ s \in S. \tag{5.9}$$

In general, Problem (5.9) is a nonconvex MINLP because the constraints in (5.8a) are nonlinear and nonconvex for general convex travel cost functions; cf. Assumption 6. Nevertheless, we emphasize that the structural properties of Problem (5.9) depend strongly on the specific choice of travel cost and latency functions. For instance, if both are linear, Problem (5.9) is a convex mixed-integer linear problem (MILP). We elaborate on problem variants in Section 5.5, where we study different choices of travel cost and latency functions in more detail.

5.3.2 Congestion Model under the System Optimum

For a given demand realization $d \in \mathcal{U}$, the KKT conditions of Problem (5.5) can be stated as

$$f = \sum_{s \in S} x^s, \tag{5.10a}$$
$$\sum_{a \in \delta^{\text{out}}(v)} x_a^s - \sum_{a \in \delta^{\text{in}}(v)} x_a^s = d_v^s, \qquad v \in V,\ s \in S, \tag{5.10b}$$
$$0 \leq c_a(f_a) + f_a c_a'(f_a) + \tau_i^s - \tau_j^s \perp x_a^s \geq 0, \qquad a = (i,j) \in A,\ s \in S. \tag{5.10c}$$

Here, $c_a'(f_a)$ denotes the derivative of the travel cost function $c_a(f_a)$ with $a \in A$. Because the UE formulation (5.4) and the SO model (5.5) only differ in their objective functions, the corresponding KKT conditions only differ in (5.6c) and (5.10c). As before, we obtain a single-level reformulation of the congestion model (5.1) by replacing Problem (5.5) with its necessary and sufficient KKT conditions. This yields

$$\max_{d,f,x,\tau} \quad L(f) \quad \text{s.t.} \quad d \in \mathcal{U},\ (5.10a)\text{–}(5.10c). \tag{5.11}$$

Again, we introduce auxiliary binary variables $y_a^s \in \{0,1\}$ and sufficiently large constants $M_a^s, N_a^s \in \mathbb{R}_{\geq 0}$ for all $a \in A$ and $s \in S$ to obtain an equivalent MINLP reformulation of (5.7), which is given by

$$\max_{d,f,x,\tau,y} \quad L(f) \tag{5.12a}$$
$$\text{s.t.} \quad d \in \mathcal{U},\ (5.10a),\ (5.10b), \tag{5.12b}$$
$$0 \leq c_a(f_a) + f_a c_a'(f_a) + \tau_i^s - \tau_j^s \leq M_a^s(1 - y_a^s), \qquad a \in A,\ s \in S, \tag{5.12c}$$
$$0 \leq x_a^s \leq N_a^s y_a^s, \qquad a \in A,\ s \in S, \tag{5.12d}$$
$$y_a^s \in \{0,1\}, \qquad a \in A,\ s \in S. \tag{5.12e}$$

5.3.3 Variable Bounds and Big-Ms

Note that the equivalence of Problems (5.9) and (5.12) to the original congestion models under the UE and the SO relies on choosing appropriate values for the constants M_a^s, N_a^s for all $a \in A$ and $s \in S$. In this section, we show that such sufficiently large constants exist by proving valid bounds for the variables f, x, and τ.

Proposition 3. *Let $d \in \mathcal{U}$ be given arbitrarily. Then, under Assumptions 3, 4, and 6, Problems* (5.4) *and* (5.5) *each admit an optimal solution (f, x) that satisfies*

$$0 \leq x_a^s \leq \sum_{k \in K_s} d_k, \quad a \in A,\, s \in S,$$

as well as

$$0 \leq f_a \leq \sum_{s \in S} \sum_{k \in K_s} d_k, \quad a \in A.$$

Proof. Because the travel cost functions are positive and non-decreasing under Assumption 6, any positive flow on a cycle would increase the objective function value and therefore cannot occur in an optimal solution; cf., e.g., Theorem 3.8 in [3]. Hence, we can, w.l.o.g., add the constraints $0 \leq x_a^s \leq \sum_{k \in K_s} d_k$ for all $a \in A$ and $s \in S$ to Problems (5.4) and (5.5) without affecting their sets of optimal solutions. As the variables x are linearly coupled to the arc flows f via (5.2), this directly implies finite bounds for f, i.e., all flow variables are bounded. The feasibility of Problems (5.4) and (5.5) follows from Assumptions 3 and 4. Moreover, as the feasible sets are described by finitely many linear constraints, they are compact. Finally, because the objective functions of the UE and SO problems are continuous, the Weierstrass theorem yields the existence of optimal solutions to Problems (5.4) and (5.5); see, e.g., Theorem 2.4 in [115] for similar arguments. □

Proposition 4. *Let $d \in \mathcal{U}$ be given arbitrarily and suppose that Assumptions 3, 4, and 6 hold. Then, for every $(f, x) \in F(d)$, there exists τ such that $0 \leq \tau_v^s < \infty$ holds for all $v \in V$ and $s \in S$, and (f, x, τ) solves* (5.6) *or* (5.10) *under the user equilibrium or the system optimum, respectively.*

Proof. The claim can be shown in analogy to Remark 1 and the proof of Proposition 3 in [16]. □

Finally, we mention that sufficiently large constants M_a^s and N_a^s, $a \in A$, $s \in S$, to be used in Problems (5.9) and (5.12) can be obtained by exploiting Propositions 3 and 4 together with Assumption 6 and the specific structure of the travel cost functions; see, e.g., [16] for further details.

5.3.4 Existence of Solutions

To conclude this section, we now prove the existence of optimal solutions to the congestion model (5.1) under the UE and the SO.

Theorem 9. *Under Assumptions 3–6, the congestion model* (5.1) *admits an optimal solution under both the user equilibrium and the system optimum.*

Proof. The function L is continuous because of Assumption 5. Moreover, the set $\mathcal{U}$ is non-empty and compact due to Assumption 4 and, by Propositions 3 and 4, the set $F(d)$ is non-empty and bounded for all $d \in \mathcal{U}$. In addition, the graph of the set-valued map F is closed as it is described by finitely many continuous equality and inequality constraints. Applying the Weierstraß theorem completes the proof. □

5.4 Strengthened Formulations

We now describe the main strategies used to improve the computational tractability of the congestion model (5.1). These techniques reduce the problem size and strengthen the formulation by exploiting the network topology and structural properties of optimal flows. In Section 5.4.1, we discuss graph partitioning techniques based on cut vertices and blocks that can be used to derive tighter bounds for the flow variables f and x, which is what we do in Section 5.4.2. In Section 5.4.3, we present valid inequalities that eliminate cycle flows in the traffic network.

5.4.1 Graph Partitioning

Given the directed graph $G = (V, A)$, we denote its underlying simple undirected graph as $\underline{G} = (V, E)$, in which each arc (i, j) is replaced by edge $\{i, j\}$ and no multiple edges are allowed. Before proceeding, we introduce some basic graph-theoretic concepts that will be used throughout this section; see, e.g., Chapter 4 of [143] for further details.

Definition 4 (Connected Component)**.** *A* connected component *of $\underline{G}$ is a maximal subset $C \subseteq V$ such that, for all $u, v \in C$, there exists a path connecting u and v.*

Definition 5 (Cut Vertex)**.** *A* Cut vertex *(or* Articulation vertex*) is a vertex whose removal increases the number of connected components of $\underline{G}$.*

Definition 6 (Bridge)**.** *An edge $e \in E$ is a* bridge *(or* cut edge*) if the graph $(V, E \setminus \{e\})$ has more connected components than $\underline{G}$.*

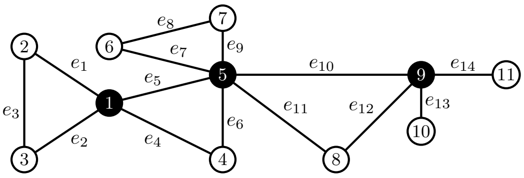
Definition 7 (2-Connected Graph)**.** *An undirected graph $\underline{G}$ is said to be* 2-connected *(or* vertex-biconnected*) if it is connected, has at least three vertices, and contains no cut vertex.*

Definition 8 (Block)**.** *A* block *$B = (V_B, E_B)$ of $\underline{G}$ is a maximal connected subgraph of $\underline{G}$ that has no cut vertex. If a block has a single edge, it is called a* trivial block *or* bridge*, otherwise it is called a* non-trivial block.

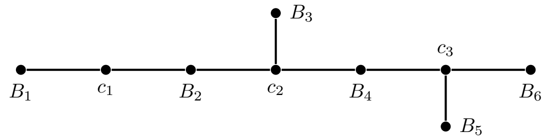
Note that $\underline{G}$ itself is a block if it is connected and has no cut vertex. Moreover, we emphasize that every edge of $\underline{G}$ belongs to exactly one block and that two blocks share at most one vertex, which must be an articulation vertex.

Definition 9 (Block-Cut Tree)**.** *The* block-cut tree *of a connected graph $\underline{G}$ is a tree T whose vertex set consists of the blocks and articulation vertices of $\underline{G}$, with an edge between a block B and an articulation vertex v whenever $v \in B$.*

For a set of vertices $S \subseteq V$, we denote by $\underline{G}[S]$ and $G[S]$ the subgraphs of $\underline{G}$ and G induced by S, respectively. In the following, we consider graph decomposition techniques based on blocks and articulation vertices. Such decompositions are not only theoretically appealing but also relevant for real-world networks. Empirical evidence indicates that many practical networks contain



(a) Graph $\underline{G}$



(b) Block-cut tree T

Figure 5.2. The graph considered in Example 6 with articulation vertices shown in black (a) and its block-cut tree (b).

a non-negligible fraction of articulation points, e.g., [135] document substantial articulation structure in U.S. road networks, underscoring the practical relevance of such decomposition techniques.

Example 6. *We consider the undirected graph* $\underline{G} = (V, E)$ *with vertex set* $V = \{1, \ldots, 11\}$ *and edge set* $E = \{e_1, \ldots, e_{14}\}$ *illustrated in Figure 5.2 (a). This graph has three articulation vertices,* $c_1 = 1$*,* $c_2 = 5$*, and* $c_3 = 9$*, and six blocks:*

$$B_1 = (\{1,2,3\}, \{e_1, e_2, e_3\}),\ B_2 = (\{1,4,5\}, \{e_4, e_5, e_6\}),$$
$$B_3 = (\{5,6,7\}, \{e_7, e_8, e_9\}),\ B_4 = (\{5,8,9\}, \{e_{10}, e_{11}, e_{12}\}),$$
$$B_5 = (\{9,10\}, \{e_{13}\}),\ B_6 = (\{9,11\}, \{e_{14}\}).$$

Here, B_1*,* B_2*,* B_3*, and* B_4 *are non-trivial blocks (2-connected subgraphs), whereas* B_5 *and* B_6 *are trivial blocks (bridges). The corresponding block-cut tree is shown in Figure 5.2 (b).*

Under Assumption 6, UE and SO solutions cannot have positive flow on directed cycles; see, e.g., Theorem 3.8 in [3]. This property also holds for the source-aggregated formulation in Problems (5.4) and (5.5); see, e.g., Section 5.2.3 in [33] for further details. The absence of directed cycle flows allows us to strengthen the formulation of the congestion models by fixing the values of certain variables in a preprocessing step. Specifically, any dipath that exits a block through an articulation vertex and re-enters it through the same vertex forms a directed cycle. Hence, all optimal flows with origin and destination vertices in the same block can be restricted to dipaths entirely contained in that block.

Proposition 5. *Suppose that Assumption 6 holds. Let* $B = (V_B, E_B) \subseteq \underline{G}$ *be a block and let* $u, v \in V_B$ *be such that there exists a dipath from* u *to* v *in* G*. Then, the following statements hold.*

(i) There exists at least one dipath from u *to* v *in* G *that is entirely contained in* $G[V_B]$*.*

(ii) Any dipath from u *to* v *used in an optimal UE or SO solution is entirely contained in* $G[V_B]$*.*

Proof. We first recall a key property of block-cut trees that will be used in the proof. To this end, let T be a block-cut tree of $\underline{G}$, i.e., a block B is a vertex of T and its neighbors in T are precisely the cut vertices of $\underline{G}$ contained in B. If a path in $\underline{G}$ leaves B through a vertex $w \in V_B$, then w must be a cut vertex. This is because the edge connecting w to a vertex outside of V_B belongs to a different block B' and $w \in B \cap B'$ implies that w is a cut vertex. Once the path leaves B through w, it continues within the subtree of $T - B$ that contains w. Because T is a tree, the only way for the path to return to B is through vertex w. Because every arc in G corresponds to an edge in $\underline{G}$, the same property holds for dipaths in G: any dipath that exits $G[V_B]$ must re-enter through the same cut vertex w, thereby forming a directed cycle through w.
We now prove the two statements. By assumption, there exists a dipath from u to v in G. As shown above, each time this dipath leaves $G[V_B]$, it must re-enter through the same cut vertex, forming a directed cycle. Removing all such cycles yields a dipath from u to v that is entirely contained in $G[V_B]$, which proves (i). Under Assumption 6, optimal UE and SO solutions do not carry flow on directed cycles. Due to (i), any dipath that leaves and later re-enters $G[V_B]$ forms a directed cycle. Hence, all dipaths carrying positive flow in an optimal solution must be entirely contained in $G[V_B]$, which proves (ii). □

By Part (ii) of Proposition 5, we can a priori fix certain variables of the congestion model (5.1) to zero. More formally, we have the following result.

Corollary 1. *Suppose that Assumptions 3–6 hold. Let (d, f, x, τ, y) be an optimal solution to either Problem* (5.9) *or Problem* (5.12)*. Further, let $s \in S$ be given arbitrarily and let B_s denote a block of $\underline{G}$ containing s. For each $k = (s, t_k) \in K_s$, let B_{t_k} be a block of $\underline{G}$ containing t_k and let $\mathcal{B}_k$ be the set of blocks on the unique path from B_s to B_{t_k} in the block-cut tree T of $\underline{G}$. Moreover, let*

$$\mathcal{B}_s = \bigcup_{k \in K_s} \mathcal{B}_k \quad \textit{and} \quad A_{\mathcal{B}_s} = \{(i,j) \in A \colon \{i,j\} \in E_B \textit{ for some } B \in \mathcal{B}_s\}.$$

Then, it holds

$$x_a^s = 0,\, y_a^s = 0, \quad a \notin A_{\mathcal{B}_s}.$$

We can also fix additional variables that would take the value of zero in any optimal solution. For instance, we can a priori fix the variables x_a^s and y_a^s for all arcs $a \in A$ that are not reachable from an origin $s \in S$.

Lemma 1. *Let $d \in \mathcal{U}$ be given arbitrarily. For each $s \in S$, let the set of vertices that are reachable from source s be given by*

$$V_s = \{v \in V \colon \textit{there exists a } s\text{–}v \textit{ dipath}\}.$$

Then, for any $(f, x) \geq 0$ that satisfies (5.2) *and* (5.3)*, it holds $x_a^s = 0$ for all $s \in S$ and $a = (i,j) \in A$ with $i \notin V_s$.*

Proof. The claim immediately follows from the definition of the set V_s, $s \in S$. □

We use Lemma 1 to fix some of the binary variables y_a^s in Problems (5.9) and (5.12), which indicate whether arc $a \in A$ is used by any commodity with origin $s \in S$.

Corollary 2. *Suppose that Assumptions 3–6 hold. Let* (d, f, x, τ, y) *be feasible for either Problem* (5.9) *or Problem* (5.12). *For each* $s \in S$*, let the set of vertices* V_s *reachable from source* s *be defined as in Lemma 1. Then,* $y_a^s = 0$ *holds for all* $s \in S$ *and* $a = (i, j) \in A$ *with* $i \notin V_s$.

Lemma 1 also allows us to reduce the number of constraints in the model by removing the redundant flow conservation constraints

$$\sum_{a \in \delta^{\text{out}}(v)} x_a^s - \sum_{a \in \delta^{\text{in}}(v)} x_a^s = 0, \quad s \in S,\, v \notin V_s,$$

from the original congestion model.

5.4.2 Improved Variable Bounds

Building on the results of Section 5.4.1, we now compute tighter bounds for the flow variables f and x, which can be used to obtain smaller values for the big-M constants in Problems (5.9) and (5.12).

Proposition 6. *Suppose that Assumptions 3, 4, and 6 hold, and let* $d \in \mathcal{U}$ *and* $(f, x) \in F(d)$ *be given arbitrarily. For each* $s \in S$ *and each* $k \in K_s$*, let* $\mathcal{B}_k$ *be defined as in Corollary 1, and define*

$$A_{\mathcal{B}_k} := \{(i, j) \in A \colon \{i, j\} \in E_B \text{ for some } B \in \mathcal{B}_k\}.$$

Moreover, let the set of vertices V_s *reachable from source* s *be defined as in Lemma 1. Then, for all* $a = (i, j) \in A$ *and* $s \in S$*, the flows satisfy*

$$0 \le x_a^s \le \bar{x}_a^s := \sum_{k \in K_s} d_k \mathbf{1}_{\{a \in A_{\mathcal{B}_k}\}},$$

as well as

$$0 \le f_a \le \bar{f}_a := \sum_{s \in S} \sum_{k \in K_s} d_k \mathbf{1}_{\{a \in A_{\mathcal{B}_k}\}},$$

where $\mathbf{1}_{\{a \in A_{\mathcal{B}_k}\}}$ *is an indicator function that equals* 1 *if* $a \in A_{\mathcal{B}_k}$*, and* 0 *otherwise.*

Proof. The claim immediately follows from Proposition 3, Corollary 1, Lemma 1, and the constraints in (5.2). □

5.4.3 Cycle-Elimination Constraints

Because UE and SO solutions cannot carry positive flow on cycles, we can leverage this property to further strengthen the MINLP reformulations of the congestion models by adding cycle-elimination constraints. To this end, let $\mathcal{R}$ denote the set of all simple directed cycles in the network. Then, for all $r \in \mathcal{R}$ and $s \in S$, the associated cycle-elimination constraint is given by

$$\sum_{a \in r} y_a^s \le |r| - 1. \tag{5.13}$$

Here, y_a^s is the binary variable indicating whether arc $a \in r$ is used by any commodity originating in $s \in S$ and $|r|$ is the number of arcs in (or the length of) the cycle $r \in \mathcal{R}$. Constraint (5.13) ensures that at most $|r| - 1$ arcs can carry positive flow, thereby preventing any directed cycles in an optimal solution. To reduce the number of cycle-elimination constraints added to the model, we exploit Corollary 1 so that Constraint (5.13) only needs to be imposed for cycles contained within the blocks associated with a source node s.

Proposition 7. *Suppose that Assumptions 3–6 hold. Let $s \in S$ be given arbitrarily and let the set $\mathcal{B}_s$ be the set of blocks associated with s as defined in Corollary 1. For each $B \in \mathcal{B}_s$, let V_B denote the set of nodes in block B. Further, let*

$$\mathcal{R}_s = \{r \in \mathcal{R} \colon V(r) \subseteq V_B \text{ for some } B \in \mathcal{B}_s\}, \tag{5.14}$$

where $V(r)$ denotes the set of vertices visited by cycle $r \in \mathcal{R}$. Then, the inequalities

$$\sum_{a \in r} y_a^s \leq |r| - 1, \quad r \in \mathcal{R}_s \tag{5.15}$$

are valid for Problems (5.9) *and* (5.12)*.*

Proof. We prove the claim for the UE as the SO case can be shown analogously. Let (d, f, x, τ, y) be feasible for Problem (5.9). We prove the claim by contradiction. To this end, suppose that there exist $s \in S$ and $r \in \mathcal{R}_s$ with

$$\sum_{a \in r} y_a^s > |r| - 1.$$

Because $y^s \in \{0,1\}^{|A|}$, this implies $y_a^s = 1$ for all $a \in r$. By constraints (5.8a), we then have $c_a(f_a) + \tau_i^s - \tau_j^s = 0$ for all $a = (i,j) \in r$. Summing these equalities over all arcs $a = (i,j) \in r$ yields

$$\sum_{a \in r} c_a(f_a) = 0,$$

which contradicts Assumption 6. This concludes the proof. □

5.5 Congestion Measures, Travel Cost Functions and Demand Uncertainty Modeling

In the previous sections, we have presented a general framework for modeling and solving the congestion model under demand uncertainty. The framework is applicable to a wide range of congestion measures, travel cost functions, and uncertainty sets, as long as they satisfy the assumptions stated in Section 5.2. In this section, we discuss the specific modeling choices for these components that we adopt in our computational study in Section 5.6. In particular, we consider three different congestion measures and present them in Sections 5.5.1, along with practical considerations for their implementation, we consider three different travel cost functions in Section 5.5.2, and three different uncertainty sets in Section 5.5.3, respectively.
Throughout this section, we use $0 < u_a < \infty$ to denote the practical capacity of an arc $a \in A$, which represents the maximum flow that an arc can accommodate under ideal operating conditions without causing congestion; see, e.g., Section 1.5 in [115] for further details.

5.5.1 Latency Functions

We measure congestion using one of the following three latency functions. We emphasize that all three considered congestion measures are continuous in the flows f, i.e., they satisfy Assumption 5.

Maximum Utilization Function

We measure congestion by focusing on the most heavily used arc in the network. To this end, we consider the function

$$L(f) = \max_{a \in A} \frac{f_a}{u_a}. \tag{5.16}$$

In particular, (5.16) can be used to identify the network's most severe bottleneck.

Total Utilization Function

Whereas the previous function focuses on bottlenecks in the network, we also study the overall traffic load in the network by considering the linear function

$$L(f) = \sum_{a \in A} \frac{f_a}{u_a}. \tag{5.17}$$

Bureau of Public Roads (BPR) Function

Another congestion measure that is widely used in the literature is the BPR function [138], which is given by

$$L(f) = \sum_{a \in A} \mathrm{BPR}(f_a) \tag{5.18}$$

with

$$\mathrm{BPR}(f_a) = c_a^{\mathrm{fix}} \left(1 + \alpha \left(\frac{f_a}{u_a}\right)^{\beta}\right), \quad a \in A. \tag{5.19}$$

Here, $c_a^{\mathrm{fix}} \in \mathbb{R}_{>0}$ is the free-flow time of arc $a \in A$, i.e., the travel time experienced if the arc is used under ideal conditions. Moreover, the parameters $\alpha, \beta \in \mathbb{R}_{>0}$ can be adjusted to account for how quickly travel times increase as flows approach the practical capacity of an arc. Commonly used values are $\alpha = 0.15$ and $\beta = 4$, which reflect typical empirical observations of congestion effects and will also be used in our computational study in Section 5.6. Note that the BPR function (5.19) is convex for $f \geq 0$ in this case.
We point out that the maximum utilization function (5.16) and degree-4 polynomials such as those in the BPR function are not directly supported by general-purpose solvers such as Gurobi. To handle these cases, we thus apply the following reformulation techniques.

Reformulation of the Maximum Utilization Function

When measuring congestion using the maximum utilization function, we consider the problem

$$\max_{d,f} \max_{a \in A} \frac{f_a}{u_a} \quad \text{s.t.} \quad d \in \mathcal{U},\ f \in F(d). \tag{5.20}$$

Because the outer maximization corresponds to an optimization problem and the inner maximization is taken over a discrete set of arcs, the resulting max–max structure cannot be reformulated using a standard epigraph reformulation. Nevertheless, an equivalent MINLP reformulation of Problem (5.20) can be obtained by introducing auxiliary binary variables $w_a \in \{0, 1\}$ for all $a \in A$ as well as a sufficiently large constant $M \in \mathbb{R}_{>0}$. Specifically, Problem (5.20) is equivalent to

$$\begin{aligned}
\max_{d,f,x,\lambda} \quad & \lambda && \text{(5.21a)}\\
\text{s.t.} \quad & \lambda \leq \frac{f_a}{u_a} + M(1 - w_a), \quad a \in A, && \text{(5.21b)}\\
& \sum_{a \in A} w_a = 1, && \text{(5.21c)}\\
& w_a \in \{0, 1\}, \quad a \in A, && \text{(5.21d)}\\
& d \in \mathcal{U},\ (f, x) \in F(d). && \text{(5.21e)}
\end{aligned}$$

In Problem (5.21), exactly one of the w variables takes the value 1. If $w_a = 1$ holds for some arc $a \in A$, the corresponding constraint in (5.21b) enforces $\lambda \leq f_a/u_a$. Because the objective is to maximize λ, this inequality is satisfied with equality for the arc attaining the maximum utilization ratio. All other w variables take the value 0. Nevertheless, to ensure the correctness of the formulation, it is essential to select a sufficiently large constant M. A valid choice is

$$M := \max_{a \in A} \left\{ \frac{\bar{f}_a}{u_a} \right\}, \tag{5.22}$$

where $\bar{f}_a$ denotes an upper bound on the arc flows; see Proposition 6.

Handling Degree-4 Polynomials

Nowadays, general-purpose MILP solvers such as Gurobi can handle convex quadratic programs (MIQPs), but higher-degree polynomials are not directly supported. To address the congestion model with the BPR latency function, we thus introduce auxiliary variables $\phi \in \mathbb{R}^{|A|}_{\geq 0}$ modeling

$$\phi_a = f_a \cdot f_a, \quad a \in A,$$

so that degree-4 polynomials are expressed using only quadratic terms.

5.5.2 Travel Cost Functions

We use the BPR function presented in (5.18) for the travel costs. In this work, we focus on settings with $\alpha = 0.15$ and $\beta \in \{1, 2, 4\}$. For $\beta = 1$, the cost functions are linear; for $\beta = 2$, they are quadratic; and for $\beta = 4$, we obtain the classic BPR function that is commonly studied in the literature and also used as the latency function in this work. In all three cases, the travel cost functions are convex, non-decreasing, and continuously differentiable in $f \in \mathbb{R}^{|A|}_{\geq 0}$, i.e., they satisfy Assumption 6. To handle the degree-4 polynomials in the travel cost functions with $\beta = 4$, we apply the reformulation described in Section 5.5.1.

Remark 2. *In Problem* (5.12)*, the derivatives $c'_a(f_a)$ are polynomials of one degree lower than the original travel cost functions $c_a(f_a)$. Hence, the terms $f_a c'_a(f_a)$, $a \in A$, in* (5.12c) *are polynomials of the same degree as $c_a(f_a)$. As a result, considering the SO instead of the UE does not affect the structural properties of the MINLP reformulation of the congestion model.*

Given the above reformulation techniques and modeling choices, we now summarize the problem variants arising from different selections of the latency function and travel cost functions in

Table 5.1. Properties of the MINLP reformulations of the congestion model depending on the choice of $\beta \in \{1, 2, 4\}$ in the BPR travel cost functions and the choice of the latency function $L(f)$, i.e., maximum utilization (5.16), total utilization (5.17), or BPR (5.18).

β	$L(f)$	convex feasible set	problem type
1	(5.16), (5.17)	✓	MILP
1	(5.18)	✓	MIQP
2	(5.16), (5.17), (5.18)	X	MIQP
4	(5.16), (5.17), (5.18)	X	MIQP

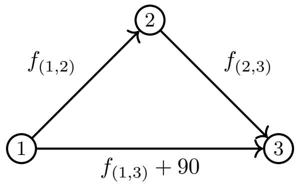


Figure 5.3. The network considered in Example 7. Arc labels represent the travel cost functions associated with each arc.

Table 5.1. Note that the MINLP reformulations of the congestion model under the user equilibrium (5.9) and the system optimum (5.12) have the same structural properties, as discussed in Remark 2.

Finally, let us briefly comment on the monotonicity of the latency function with respect to travel demand. We note that increasing the travel demand of a single commodity does not necessarily lead to higher congestion levels, even if the travel cost functions are linear. The following example illustrates that higher travel demand may in fact reduce congestion levels due to route reallocation effects. For related monotonicity phenomena and paradoxes in congestion games, we also refer to [49].

Example 7. *We consider the network shown in Figure 5.3, which is taken from Example 3 in [49]. We consider three OD pairs given by* $(1, 2)$*,* $(2, 3)$*, and* $(1, 3)$ *with demand* 1*,* 100*, and* 20*, respectively. The resulting equilibrium flows under the UE are* $f_{(1,2)} = 4$*,* $f_{(1,3)} = 17$*, and* $f_{(2,3)} = 103$*. Let us now assume that all arcs have a practical capacity of* 1*, i.e.,* $u_a = 1$ *for all* $a \in A = \{(1, 2), (2, 3), (1, 3)\}$*. Considering the maximum utilization function* (5.16) *as the congestion measure, the worst congestion is attained on arc* $(2, 3)$*, namely* $f_{(2,3)}/u_{(2,3)} = 103$*. As observed in [49], if we now increase the travel demand of commodity* $(1, 2)$ *from* 1 *to* 4*, some of its flow is reallocated. The new equilibrium flows are given by* $f_{(1,2)} = 6$*,* $f_{(1,3)} = 18$*, and* $f_{(2,3)} = 102$*, which leads to the worst-case congestion level of* $102 < 103$*. Hence, increasing the travel demand of a commodity can in fact lead to reduced congestion.*

The previous observation implies that the demand realization causing the worst-case congestion in our framework does not necessarily lie on the boundary of the uncertainty set. Due to route reallocation effects, intermediate demand realizations can produce higher congestion, which contrasts with classic robust optimization settings in which worst cases typically occur at the boundary of the uncertainty set.

5.5.3 Uncertainty Set

We assume that the travel demand $d = (d_k)_{k \in K}$ is uncertain. In this work, we consider three approaches to model this uncertainty: budgeted uncertainty, ellipsoidal uncertainty, and the hose model.

Budgeted Uncertainty

As it is unlikely that all demands simultaneously realize in a worst-case manner, a common approach is to adopt a budgeted (or Γ-robust) uncertainty modeling [25, 26, 127]. This model has been widely used in transportation contexts, including network design [97, 98], traffic assignment [110, 111, 74], and network pricing [16].
In the budgeted uncertainty model, each commodity k has a nominal demand $\bar{d}_k \in \mathbb{R}_{\geq 0}$, which corresponds to the demand value provided in the original instance data, and a maximum deviation $\delta_k \in \mathbb{R}_{\geq 0}$ from the nominal value so that $d_k \in [\bar{d}_k - \delta_k, \bar{d}_k + \delta_k]$ holds for all $k \in K$. Each commodity may deviate by an arbitrary fraction of its maximum deviation δ_k from the nominal value as long as the budget $\Gamma \in \{0, \ldots, |K|\}$ for the aggregated percentage deviation is not exceeded. Overall, this can be modeled using the uncertainty set

$$\mathcal{U}_{\text{budget}} = \left\{ d \in \mathbb{R}^{|K|} \colon d_k = \bar{d}_k + \delta_k z_k,\ z_k \in [-1, 1],\ k \in K,\ \sum_{k \in K} |z_k| \leq \Gamma \right\}. \tag{5.23}$$

Here, the parameter Γ controls the size of the uncertainty set. For $\Gamma = 0$, no uncertainty is taken into account, i.e., we consider the deterministic case, whereas for $\Gamma = |K|$, all commodities may simultaneously experience their maximum deviation. Note that $\mathcal{U}_{\text{budget}}$ is non-empty, convex and compact. By assuming $\delta_k \leq \bar{d}_k$ for all $k \in K$, we ensure $d \in \mathbb{R}^{|K|}_{\geq 0}$, i.e., Assumption 4 is satisfied.

Ellipsoidal Uncertainty

Using an ellipsoidal uncertainty modeling, we replace the cardinality constraint in the budgeted uncertainty set with a quadratic constraint on the vector of deviations, also called perturbation vector in the following. The ellipsoidal uncertainty set is thus given by

$$\mathcal{U}_{\text{ellipsoid}} = \left\{ d \in \mathbb{R}^{|K|} \colon d_k = \bar{d}_k + \delta_k z_k,\ k \in K,\ \|z\|_2 \leq \rho \right\}, \tag{5.24}$$

where $\rho > 0$ is the radius of the ellipsoid and $z = (z_k)_{k \in K}$ is the perturbation vector. To have a fair comparison between the ellipsoidal and the budgeted uncertainty modeling, we select ρ such that the uncertainty sets $\mathcal{U}_{\text{budget}}$ and $\mathcal{U}_{\text{ellipsoid}}$ cover a comparable range of demand realizations. As shown in [87], there exists a geometric relationship between these two sets, and it depends on the choice of parameters ρ and Γ. In their analysis, they describe two cases:

1. If $\rho = \Gamma$, the ellipsoid is the smallest one containing the polyhedron of the budgeted uncertainty model.
2. If $\rho = \Gamma/\sqrt{|K|}$, the ellipsoid is the largest one contained in the polyhedron of the budgeted uncertainty model.

In this work, we introduce a third intermediate case with $\rho = \sqrt{\Gamma}$, which yields uncertainty sets of comparable size. In Figure 5.4, we illustrates all three cases. As before, we assume that $\delta_k \leq \bar{d}_k$ holds for all $k \in K$ so that the set $\mathcal{U}_{\text{ellipsoid}}$ satisfies Assumption 4.

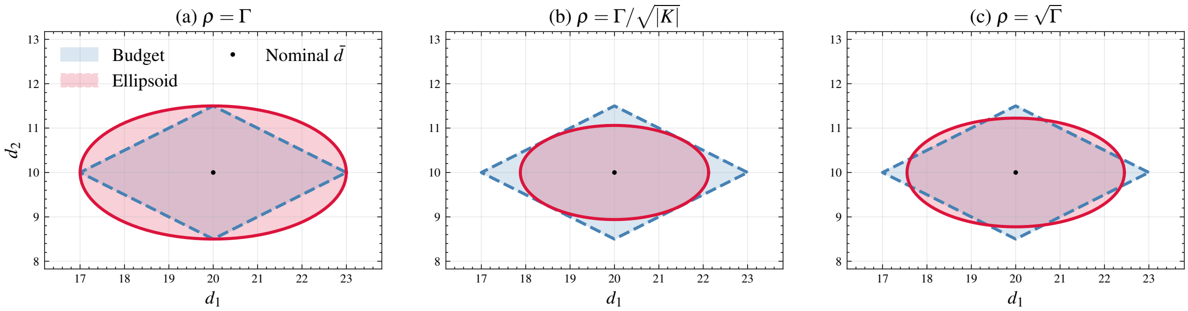


Figure 5.4. Comparison of budgeted and ellipsoidal uncertainty sets.

Hose Model

The hose model, originally introduced in the telecommunications network design literature [59, 57], aggregates demand uncertainty at the node level rather than at the commodity level. This model has been applied in various network design contexts; see, e.g., [5, 42, 95]. For each node $i \in V$, a bound b_i for the total traffic generated by that node is specified. Here, we consider the so-called symmetric hose uncertainty set given by

$$\mathcal{U}_{\text{hose}} = \left\{ d \in \mathbb{R}^{|K|}_{\geq 0} : \sum_{\{k \in K:\, s_k = i \text{ or } t_k = i\}} d_k \leq b_i,\ i \in V \right\}, \tag{5.25}$$

where s_k and t_k denote the source and the destination node of commodity k, respectively. To ensure a fair comparison between the hose model and the budgeted and ellipsoidal uncertainty sets, we determine the node bounds as follows

$$b_i = \sum_{k \in K_i} \bar{d}_k + \sum_{j=1}^{\min\{\Gamma, |K_i|\}} \delta_k^{(j)},$$

where K_i is the set of commodities with source or destination at node $i \in V$ and $\delta_k^{(j)}$ are the deviations of the commodities in K_i sorted in non-increasing order. Hence, we consider the Γ largest deviations among the commodities in K_i. By construction, we have $\mathcal{U}_{\text{budget}} \subseteq \mathcal{U}_{\text{hose}}$ and $\mathcal{U}_{\text{hose}}$ satisfies Assumption 4.

5.6 Computational Setup and Experimental Design

In this section, we discuss the setup and design of our computational study. In Section 5.6.1, we elaborate on the hardware, the software, and the solver used in our computational study. Afterward, in Section 5.6.2, we describe our test instances.

5.6.1 Computational Setup

All experiments were performed on a 64-bit Linux system equipped with a 13th-generation Intel Core i9-13900K processor featuring 24 physical cores and 32 threads, 128 GB of RAM, and a 3 GHz base clock frequency. The models were implemented in Python 3.10.15 and solved using Gurobi 12.0.1. The time limit for each instance was set to 2 h. The parameter MIPGap

was set to 10^{-3} and we disabled Gurobi's presolve, as preliminary experiments showed that these features did not significantly improve performance for our instances and sometimes even led to worse results due to some numerical instability issues. All other parameters were left at their default settings. For each instance, we first solve the deterministic version of the problem, i.e., the problem in which all demands are set to their nominal values, and use the solution to this problem as a warm start for solving the model under demand uncertainty, as this approach has been observed to improve the solver's performance.
Code and instances are publicly available at
https://github.com/fragiancola/congestion-bilevel-under-uncertainty.

5.6.2 Test Instances

In our computational study, we consider instances of the Sioux Falls network [84], whose data is publicly available at https://github.com/bstabler/TransportationNetworks, and instances from the SNDlib [113, 112], which can be accessed at https://sndlib.put.poznan.pl.

Sioux Falls Instances

We consider a set of subnetworks derived from the Sioux Falls network, with varying sizes and levels of complexity, to evaluate the performance and scalability of the proposed optimization approach
We analyze two classes of subnetworks: five small instances with 12 nodes and five medium instances with 18 nodes, which correspond to the subgraphs induced by 50% and 75% of the original nodes, respectively.
All generated subnetworks are strongly connected, while their underlying undirected graphs are not biconnected. This structure allows us to assess the effectiveness of the enhancement techniques introduced in Section 5.4.
Figure 5.5 provides visual representations of the subnetworks including the number of nodes and arcs for each case. For each subnetwork, multiple demand configurations are considered by varying the number of commodities $|K| \in \{20, 30, 50, 75, 100\}$. Commodities are randomly selected from the set of all origin-destination pairs, ensuring heterogeneous demand patterns. Since the original Sioux Falls instances assume deterministic demands, for each commodity k, we define as d_k the demand value in the original instance, but we introduce uncertainty by allowing the demand to vary within a specified range around this original value. In the Γ-robust formulation, this value is used as the nominal demand $\bar{d}_k$, and the maximum deviation is set to $\delta_k = 0.25\,\bar{d}_k$. The parameters of the ellipsoidal and hose uncertainty sets are then derived from $\bar{d}_k$ and δ_k as described in Section 5.5.3.
To mitigate numerical instability in the solver, all demand values and practical arc capacities are scaled by a common factor, preventing the occurrence of very small coefficients in the formulations.

SNDlib Instances

Because the SNDlib instances have originally been created for telecommunication network design, they do not include all parameters required for traffic assignment problems, such as arc capacities and free-flow travel times. Therefore, we extract the network topology and demand data from SNDlib and generate the missing parameters as follows. From each SNDlib

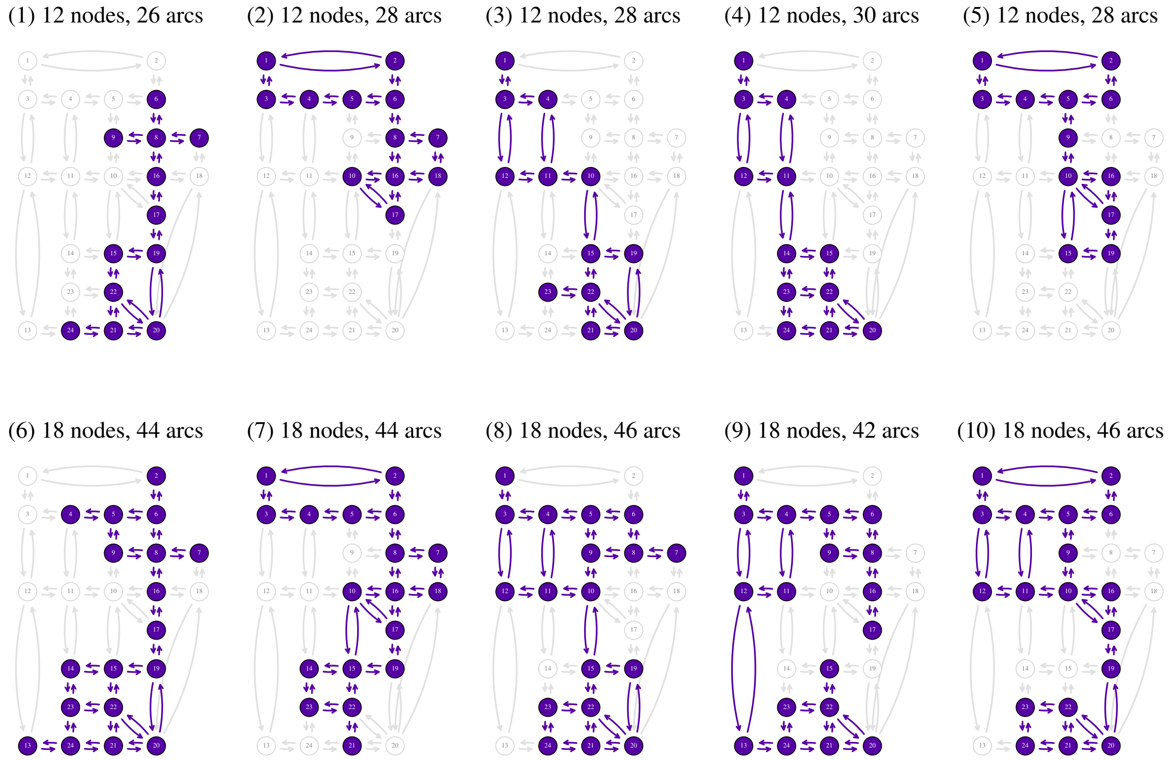


Figure 5.5. In the first row, the five small-sized subnetwork with 12 nodes each; in the second row, the five medium-sized subnetworks with 18 nodes each. In purple are highlighted the nodes and the arcs included in the subnetworks.

Table 5.2. SNDlib network instances characteristics. The number of commodities for each demand level is reported.

Instance	$\lvert V\rvert$	$\lvert A\rvert$	$\lceil 10\,\%\lvert K\rvert\rceil$	$\lceil 25\,\%\lvert K\rvert\rceil$	$\lceil 50\,\%\lvert K\rvert\rceil$	$\lceil 75\,\%\lvert K\rvert\rceil$	$\lvert K\rvert$
abilene	12	30	13	33	66	99	132
atlanta	15	44	21	52	105	158	210
nobel-us	28	42	9	22	45	68	91
pdh	11	68	2	6	12	18	24

(a) abilene (b) atlanta (c) nobel-us (d) pdh

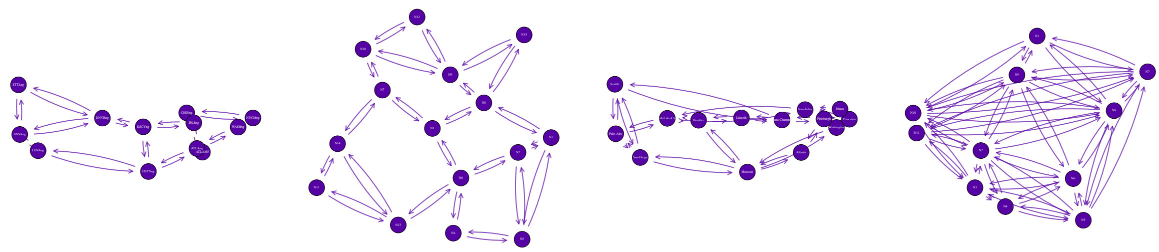


Figure 5.6. Network topologies of the considered SNDlib instances.

instance, we take the set of nodes V with their geographical coordinates, the set of links, and the traffic demands representing origin-destination pairs with associated traffic volumes. The instance generation procedure consists of the following steps. First, we parse the graph topology from the SNDlib native format, duplicating each undirected link to create a directed bidirectional graph. Second, arc capacities are set proportionally to the total demand across all commodities, introducing heterogeneity through a random perturbation: $u_a = \frac{D}{\rho} \cdot \xi_a$. Here $D = \sum_{k \in K} d_k$ denotes the total demand; $\rho = 10$ is a scaling factor that controls the overall capacity level relative to the aggregate demand; and $\xi_a \sim U[0.1, 2.0]$ is a uniformly distributed random factor that introduces variability across arcs. Finally, to set the free-flow travel time c_a^0 of each arc $a = (i, j)$, we consider the Euclidean distance between its endpoints, computed from the geographical coordinates provided in the SNDlib data and normalized with respect to the maximum distance in the network. The free-flow cost c_a^0 is then obtained by linearly scaling the normalized distance to a given cost range $[c^0_{\text{min}}, c^0_{\text{max}}] = [1, 50]$.
Similarly to the Sioux Falls instances, we consider multiple demand configurations by varying the number of commodities for each network: let $|K|$ be the total number of commodities available in the original instance, we consider five demand levels corresponding to 10%, 25%, 50%, 75%, and 100% of the original commodities. Commodities are randomly selected from the set of all origin-destination pairs, and the demand d_k for each commodity k is set equal to the corresponding original demand value. For the Γ-robust formulation, this value serves as the nominal demand $\bar{d}_k$, with the maximum deviation set to $\delta_k = 0.25\,\bar{d}_k$. The parameters of the ellipsoidal and hose uncertainty sets are then derived from $\bar{d}_k$ and δ_k as described in Section 5.5.3.
Table 5.2 summarizes the main characteristics of the considered SNDlib instances, while Figure 5.6 illustrates their network topologies.
As before, to mitigate numerical instability in the solver, all demand values and practical arc capacities are scaled by a common factor, preventing the occurrence of very small coefficients

in the formulations.

5.7 Computational Results and Discussion

In this section, we present and discuss the results of our computational study. When it is not explicitly stated otherwise, results refer to the Sioux Falls subnetworks described in Section 5.6.2 under the budgeted uncertainty set introduced in Section 5.5.3, considering the User Equilibrium formulation (5.9) and the enhancement techniques described in Section 5.4. For each instance, we consider all combinations of the three travel cost functions (BPR with different β values) and the three latency functions (Maximum Utilization, Total Utilization, and BPR) presented in Section 5.5. The section is organized as follows. In Section 5.7.1, we discuss the computational performance and the scalability of our approach. In Section 5.7.2, we analyze the sensitivity of the optimal solutions to the choice of the different travel cost and latency functions. In Section 5.7.3, we compare the User Equilibrium and System Optimum models in terms of congestion levels and computational performance. In Section 5.7.4, we compare the different uncertainty sets introduced in Section 5.5.3. Finally, in Section 5.7.5, we present results on the SNDlib instances described in Section 5.6.2.

For the ease of presentation, we abbreviate the maximum utilization and total utilization latency functions as max_ratio and sum_ratio, respectively. Moreover, we refer to the travel cost functions with $\beta = 1$, $\beta = 2$, and $\beta = 4$ as linear, quadratic, and BPR, respectively.

5.7.1 Performance analysis

In this section, we analyze the performance of the proposed optimization approach considering the enhancement techniques presented in Section 5.4, the scalability of the approach, and the performance across different combinations of cost and congestion functions.

Impact of Enhancement Techniques

By tightened formulation, we refer to the formulation obtained after applying the enhancement techniques described in Section 5.4. These techniques reduce problem size and strengthen the formulation by exploiting network topology and structural properties of optimal flows. With standard formulation, we refer to the original formulation without any of these enhancements.

Table 5.3 reports the impact of the enhancement techniques on solver performance for five demand levels, corresponding to $|K| \in \{20, 30, 50, 75, 100\}$. Results are aggregated over the different latency functions and reported for each travel cost function, since this parameter has the largest impact on execution time, as further discussed in the scalability analysis in Section 5.7.1.

The results show that, for smaller demand levels ($|K| \in \{20, 30\}$), for both formulations all instances are solved to optimality within the time limit, with tightened formulation reducing execution time by more than 93%. As $|K|$ increases, the enhancement techniques start affecting also the solvability. The number of additional instances solved to optimality increases from 1 at $|K| = 50$ to 13 at $|K| = 75$ and 17 at $|K| = 100$. Summarizing, the tightened formulation improves both efficiency and success rate, with larger effects for higher values of $|K|$ and for the BPR travel cost function.

Table 5.3. Performance comparison between tightened and standard formulations for different values of $|K|$ and travel cost functions. Columns report the number of instances solved to optimality, average execution time in seconds (on instances solved to optimality in both configurations), speed-up ratio (time tightened/time standard), percentage of time saved, and difference in the number of optimal solutions.

		Tightened formulation		**Standard formulation**		**Comparison**		
		#opt	avg time (s)	#opt	avg time (s)	speed-up	time saved (%)	Δ #opt
|K|=20		**90**	**0.08**	**90**	**1.77**	**21.34**	**95**	**0**
	linear	30	0.03	30	1.12	32.56	97	0
	quadratic	30	0.06	30	1.95	34.37	97	0
	BPR	30	0.16	30	2.25	14.22	93	0
|K|=30		**90**	**0.29**	**90**	**9.97**	**34.53**	**97**	**0**
	linear	30	0.07	30	2.95	40.55	98	0
	quadratic	30	0.17	30	6.68	38.85	97	0
	BPR	30	0.62	30	20.30	32.63	97	0
|K|=50		**90**	**2.14**	**89**	**59.06**	**27.61**	**96**	**1**
	linear	30	0.23	30	4.68	20.56	95	0
	quadratic	30	1.04	30	17.89	17.16	94	0
	BPR	30	5.25	29	157.90	30.08	97	1
|K|=75		**90**	**16.56**	**77**	**125.97**	**7.61**	**87**	**13**
	linear	30	0.96	30	12.61	13.16	92	0
	quadratic	30	5.14	28	69.60	13.55	93	2
	BPR	30	58.02	19	388.03	6.69	85	11
|K|=100		**81**	**27.44**	**64**	**438.14**	**15.96**	**94**	**17**
	linear	30	4.86	30	407.82	83.84	99	0
	quadratic	30	26.43	26	345.65	13.08	92	4
	BPR	21	127.98	8	911.62	7.12	86	13
Total		**441**	**7.89**	**410**	**106.64**	**13.51**	**93**	**31**

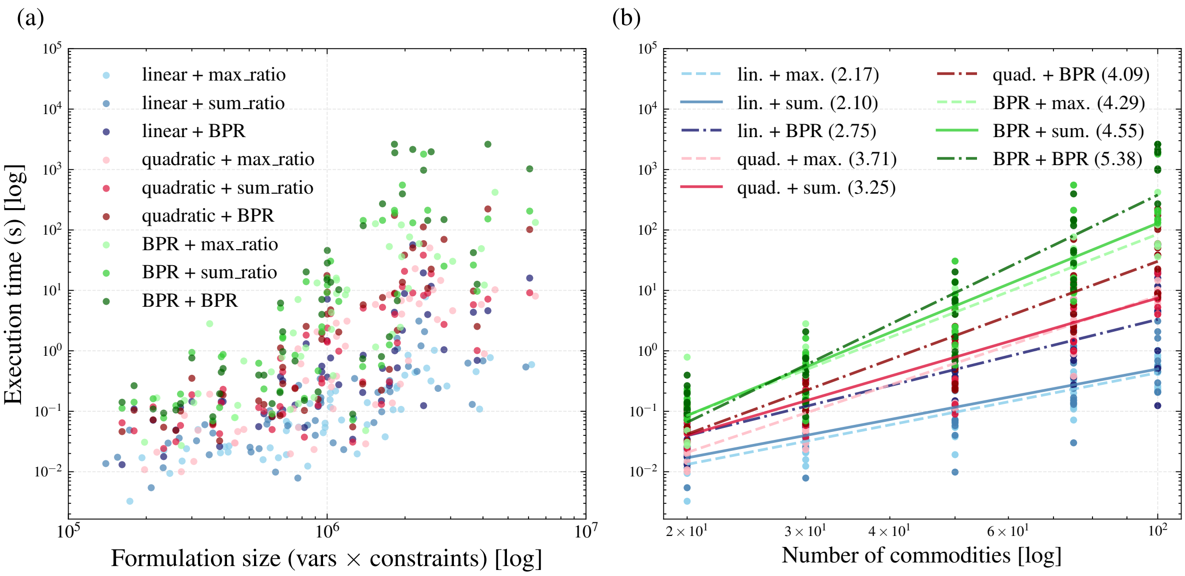


Figure 5.7. Scalability analysis for different configurations of travel cost and latency functions. (a): execution time versus problem size measured as the product of variables and constraints. (b): execution time versus number of commodities with fitted power-law models $t = a \cdot |K|^{\alpha}$, where $|K|$ is the number of commodities and α is the growth exponent (written in the legend).

Scalability Analysis

We evaluate the scalability of the optimization approach across different configurations of travel cost functions $c(f)$ and congestion measures $L(f)$. Figure 5.7 presents some results according to two complementary measures of problem size. In Figure 5.7 (a), the problem size is measured as the number of variables and constraints, #var and #constr in the figure, while in Figure 5.7(b), it is measured by the number of commodities.

Both plots in Figure 5.7 reveal a certain dispersion in execution times: for comparable formulation sizes or identical values of $|K|$, execution times span several orders of magnitude. This variability indicates that problem size alone is not sufficient to explain the computational behavior and that structural properties of the formulation play a major role. In both cases, the dominant factor affecting scalability is the choice of the travel cost function $c(f)$. Instances with BPR costs show the highest execution times and the steepest growth trends, whereas linear costs lead to the lowest execution times across all configurations. This behavior is visible in Figure 5.7 (a) through the vertical separation of color groups and is confirmed in Figure 5.7 (b) by the larger scaling exponents associated with BPR costs. Quadratic costs show intermediate behavior between linear and BPR costs. In contrast, the impact of the latency function $L(f)$ is less evident. For a fixed cost function, differences among sum_ratio, max_ratio and BPR congestion models result in smaller variations in execution time, as indicated by the overlap of points of the same color but different shades in Figure 5.7 (a) as well as in the proximity of curves with the same color but different line styles in Figure 5.7 (b). Nevertheless, it is possible to observe that, for all three cost functions, the BPR congestion measure yields the longest execution times, followed by max_ratio, while sum_ratio provides the fastest solution times. Overall, the figure indicates that algorithmic scalability is primarily determined by the choice of travel cost function $c(f)$, with BPR costs exhibiting the least favorable scaling behavior,

Table 5.4. Performance metrics by function combination. The execution time is averaged over instances solved to optimality.

	Execution Time (s)			Success Rate (%)		
$c(f)$	max_ratio	sum_ratio	BPR	max_ratio	sum_ratio	BPR
linear	0.33	0.32	3.05	100.0	100.0	100.0
quadratic	3.72	3.65	19.23	100.0	100.0	100.0
BPR	27.52	83.42	303.24	92.0	94.0	96.0

while the choice of latency function $L(f)$ plays a secondary role, but still influences execution times within each cost function category in a non-negligible manner.

Performance Across Function Combinations

As observed in the scalability analysis, the choice of travel cost function $c(f)$ has a stronger impact on computational performance than the choice of latency function $L(f)$. To quantify these differences, we aggregate performance metrics across all instances solved to optimality for each combination of travel cost and latency functions.

Table 5.4 summarizes average execution times and success rates for each configuration. The results confirm that the travel cost function $c(f)$ is the main factor affecting the computational performance. Congestion measures also have their influence, but in a different manner: there is not an absolute ranking that holds across all cost functions. For example, when using linear costs or quadratic costs, all instances are solved to optimality and the max_ratio congestion measure shows execution times comparable to sum_ratio, and faster than BPR. Nevertheless, when considering BPR tarvel costs, max_ratio leads to more challenging optimization problems, increasing the fraction of instances that fail to reach optimality within the time limit.

5.7.2 Sensitivity of Network Congestion to Travel Cost and Latency Functions

We examine now the impact of travel cost function $c(f)$ and latency function $L(f)$ on congestion estimates. To keep the analysis consistent, we consider only instances that were solved to optimality for all nine combinations of travel cost and latency functions, for each value of $|K|$ (number of commodities).

Impact of the Travel Cost Function on Congestion Estimates

Figure 5.8 presents the evolution of congestion levels as a function of the number of commodities $|K|$ for different travel cost functions $c(f)$, when optimizing for each latency function $L(f)$.

The analysis reveals the impact of travel cost functions on congestion estimates. For a fixed latency function $L(f)$ (within each panel), the choice of cost function $c(f)$ significantly affects the magnitude of estimated congestion. BPR cost functions yield lower congestion estimates than linear or quadratic costs across all three measures.

Figure 5.8a shows this effect for the max_ratio latency function: BPR costs produce estimates approximately 15%–20% lower than linear costs for large instances, indicating that nonlinear cost functions with steeper penalties near capacity are more effective in limiting extreme arc saturation. Figure 5.8c confirms this pattern for the BPR latency function, where BPR costs

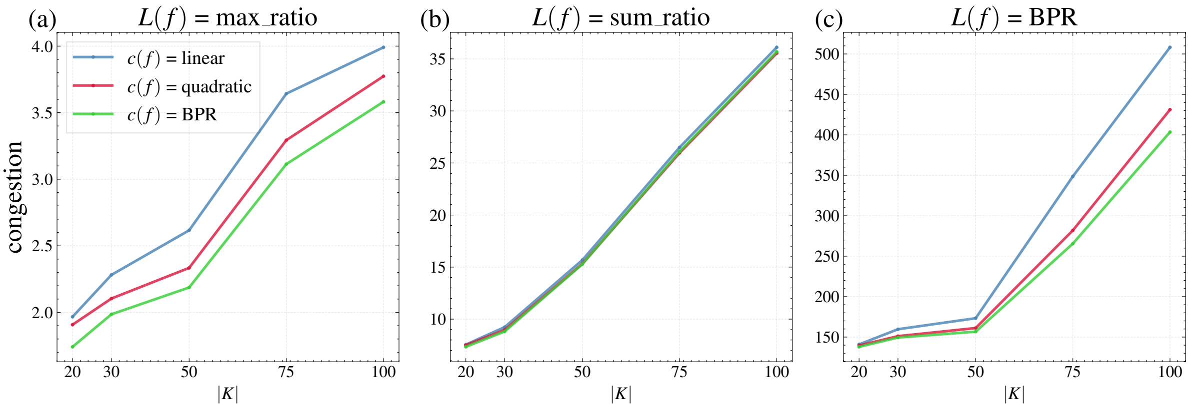


Figure 5.8. Congestion estimates when optimizing for each latency function. (a): optimization for max_ratio. (b): optimization for sum_ratio. (c): optimization for BPR congestion measure. Each panel shows how the average objective value varies with the number of commodities for different cost functions.

achieve the lowest congestion values overall, with a more pronounced effect for instances with larger $|K|$.

In contrast, Figure 5.8b shows that the sum_ratio metric is less sensitive to the choice of travel cost function, with all three functions yielding similar values across the entire range of $|K|$. This behavior is explained by the aggregate nature of the metric: while different cost functions lead to different flow distributions at the arc level, these local variations tend to compensate when summed over the network, as higher utilization on some arcs is balanced by lower utilization on others. The small size of the network further amplifies this effect, since the limited number of alternative paths constrains the degree to which flows can be redistributed in response to changes in the cost function.

Impact of the Latency function on Congestion Estimates

To assess the impact of the congestion measure used in the objective function, we analyze how solutions obtained by optimizing one latency function perform, when evaluated under alternative measures. Specifically, for a given latency function L_{opt}, we compute the optimal flow $f^*_{L_{\text{opt}}}$ and evaluate its congestion level using a different measure L_{eval}. We remark that in this analysis we do not consider the issue of multiple equilibria: there may exist an equilibrium solution that is optimal (or equivalent) with respect to L_{opt} but performs better when evaluated under L_{eval}. Here, we limit ourselves to evaluating the solution returned by the solver.

For each ordered pair $(L_{\text{opt}}, L_{\text{eval}})$, we define the relative gap

$$\text{gap}(L_{\text{opt}}, L_{\text{eval}}) = \frac{L_{\text{eval}}(f^*_{L_{\text{opt}}}) - L_{\text{eval}}(f^*_{L_{\text{eval}}})}{L_{\text{eval}}(f^*_{L_{\text{eval}}})} \times 100\%, \tag{5.26}$$

where f^*_L denotes the flow that optimizes congestion measure L.

By construction, $f^*_{L_{\text{eval}}}$ maximizes L_{eval}, and therefore the gap is non-positive. Values close to zero indicate that optimizing for L_{opt} yields solutions that are nearly optimal when evaluated under L_{eval}, while more negative values indicate a larger discrepancy between the two formulations.

Table 5.5. Relative absolute gap (%) on L_{eval} when optimizing for L_{opt}, by number of commodities $|K|$. Values are averaged over all cost functions. Negative values indicate underestimation of worst-case congestion relative to direct optimization for L_{eval}.

L_{eval}	L_{opt}	$\|K\| = 20$	$\|K\| = 30$	$\|K\| = 50$	$\|K\| = 75$	$\|K\| = 100$
max_ratio	sum_ratio	3.1	2.3	3.8	5.0	7.8
	BPR	1.6	0.5	0.7	1.1	2.9
sum_ratio	max_ratio	2.3	3.3	3.2	3.1	6.0
	BPR	1.3	1.3	1.2	1.6	1.4
BPR	max_ratio	0.1	0.3	0.9	2.7	10.2
	sum_ratio	0.2	0.7	2.1	5.6	7.8

Table 5.5 reports the gaps for all off-diagonal combinations of $(L_{\text{eval}}, L_{\text{opt}})$ and for each value of $|K|$. For small instances ($|K| \in 20, 30$), the discrepancies between latency functions are limited, with gap values remaining within 3.3% for all metric combinations. As $|K|$ increases, larger differences emerge. For example, for $|K| = 100$, optimizing with respect to sum_ratio instead of max_ratio results in a gap of approximately 8%. Similarly, optimizing for max_ratio or sum_ratio instead of BPR leads to gaps of up to 10%. Conversely, flows obtained by optimizing for BPR exhibit smaller discrepancies when evaluated under the other latency functions, with gap values below 3% in most cases. These results indicate that BPR-optimized flows lead to Maximum Utilization and Total Utilization levels close to their respective worst-case. Overall, the magnitude of the discrepancy between latency function increases with the number of commodities.

5.7.3 User Equilibrium vs. System Optimum

We now compare the User Equilibrium (UE) and System Optimum (SO) formulations of the congestion model. We analyze both computational performance and congestion levels achieved by the two formulations. We focus on instances that were solved to optimality by both formulations for a fair comparison.

Computational Time Comparison

Figure 5.9 presents the computational time comparison between UE and SO formulations. Figure 5.9a shows a scatter plot of execution times on a logarithmic scale, where each point represents a single instance. Points below the diagonal line indicate instances where UE requires less time than SO. The plot reveals that UE formulations consistently require shorter execution times across all problem sizes. Moreover, as the number of commodities increases (lighter colored points), the relative advantage of UE becomes more pronounced.

Figure 5.9 (b) aggregates execution time ratios by function combination. The mean time ratio ranges from approximately 0.4 to 0.6 across all configurations, indicating that UE requires only 40–60% of the computational time needed by SO. Linear cost functions exhibit the highest ratios (approximately 0.6), while quadratic and BPR cost functions yield progressively lower ratios, indicating that the computational advantage of UE over SO increases for more complex travel cost structures. No particular pattern is observed with respect to the latency function choice.

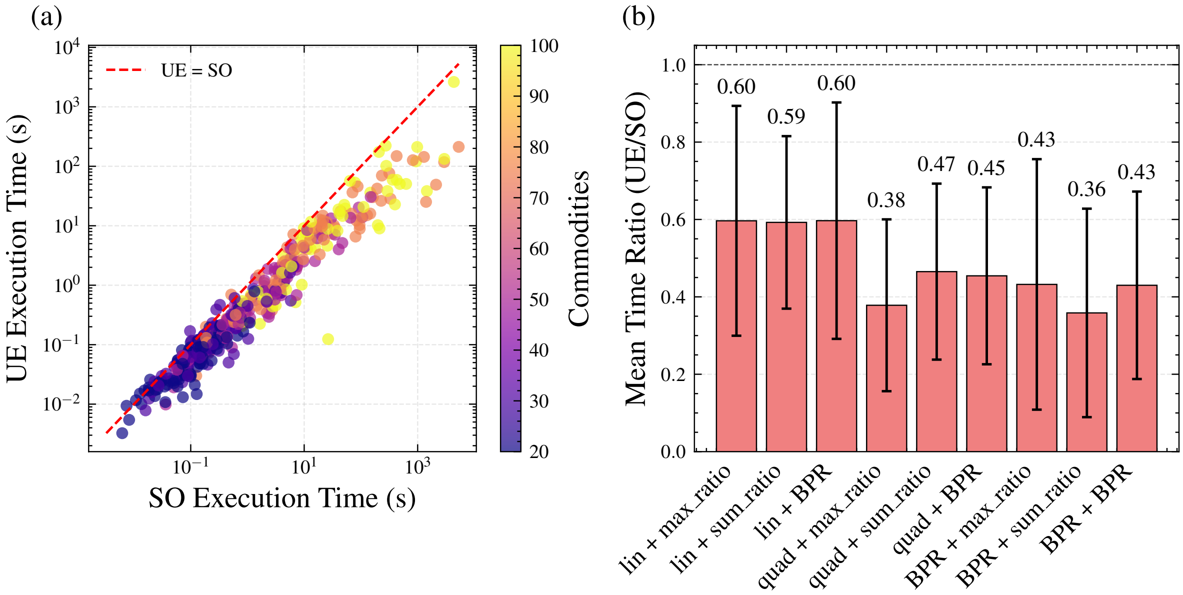


Figure 5.9. Computational time comparison between UE and SO formulations. (a): scatter plot of execution times (log scale) for UE versus SO, with color indicating the number of commodities. (b): mean time ratio (UE/SO) with standard deviation (error bars) for each combination of travel cost and latency functions. The mean values are annotated above each bar.

Congestion Levels under UE and SO

We now compare congestion levels achieved by the UE and SO formulations by analyzing the distribution of the Congestion Ratio, introduced in Section 5.2.3, across all instances and function combinations.

Figure 5.10a shows a histogram of CR values, revealing that most instances achieve CR values close to 1, indicating that UE solutions typically yield congestion levels comparable to SO solutions. However, there are notable outliers with CR values up to approximately 1.45, indicating that in some cases, UE solutions can lead to significantly higher congestion than SO solutions. At the same time, there are instances with CR values below 1, consistent with Example 5. Figure 5.10 (b) shows the distribution of CR values across different combinations of travel cost and latency functions using violin plots. Coherently with what shown in Figure 5.10 (a), the distributions are generally concentrated near 1. The combinations including the max_ratio congestion measure tends to exhibit higher CR values, indicating that UE solutions under this measure can lead to more pronounced congestion levels compared to SO solutions. In contrast, the sum_ratio function yield the smallest CR values, suggesting that UE solutions under this measure can also lead to more favorable solutions. Finally, combinations with BPR costs show a wider spread of CR values, indicating greater variability in congestion outcomes between UE and SO formulations.

5.7.4 Uncertainty Set Comparison

We now present a comprehensive comparison of five uncertainty sets introduced in Section 5.5.3:

1. Budget (Γ-robust): the polyhedral budgeted uncertainty set (5.23);

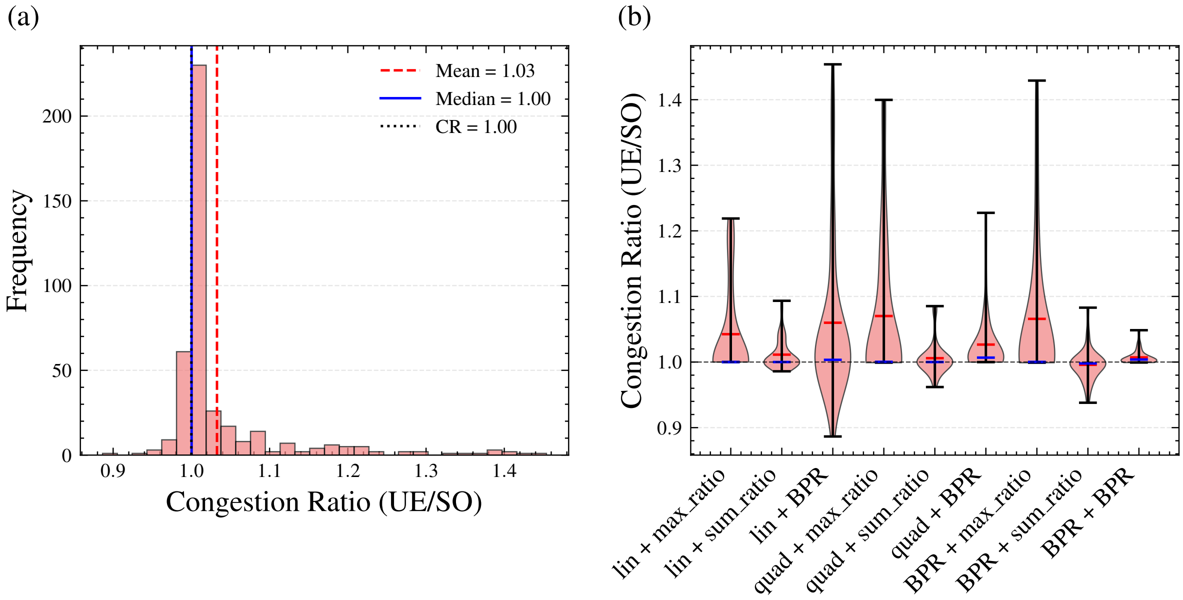


Figure 5.10. Distribution of Congestion Ratio (CR = UE/SO) across instances. (a): histogram showing the overall distribution with mean and median values indicated. (b): violin plots showing CR distribution for each combination of travel cost and latency functions, with mean (red) and median (blue) markers.

2. Ellipsoidal with $\rho = \sqrt{\Gamma}$: ellipsoidal set (5.24) with comparable size to budget;

3. Ellipsoidal with $\rho = \Gamma$: ellipsoidal set (5.24) with budget polyhedron contained in the ellipse;

4. Ellipsoidal with $\rho = \Gamma/\sqrt{|K|}$: ellipsoidal set (5.24) with ellipse contained in the budget polyhedron;

5. Hose: the symmetric hose uncertainty set (5.25).

We restrict the analysis to instances derived from the first Sioux Falls subnetwork with 12 nodes and 26 arcs (Figure 5.5), considering $|K| \in \{20, 30, 50, 75, 100\}$ commodities and all nine combinations of travel cost and latency functions, for a total of 45 instances per uncertainty set.

Computational Tractability

Figure 5.11 visualizes the number of instances solved to optimality within the time limit for each uncertainty set and function combination.
The results indicate a clear hierarchy in computational complexity: budget uncertainty is the most tractable, followed by the smaller ellipsoidal sets ($\rho = \Gamma/\sqrt{|K|}$ and $\rho = \sqrt{\Gamma}$), the hose model, and finally the larger ellipsoidal formulation ($\rho = \Gamma$). This ordering reflects both the geometric properties of the sets (contained vs. containing) and the nature of the resulting optimization problems (polyhedral vs. second-order cone).

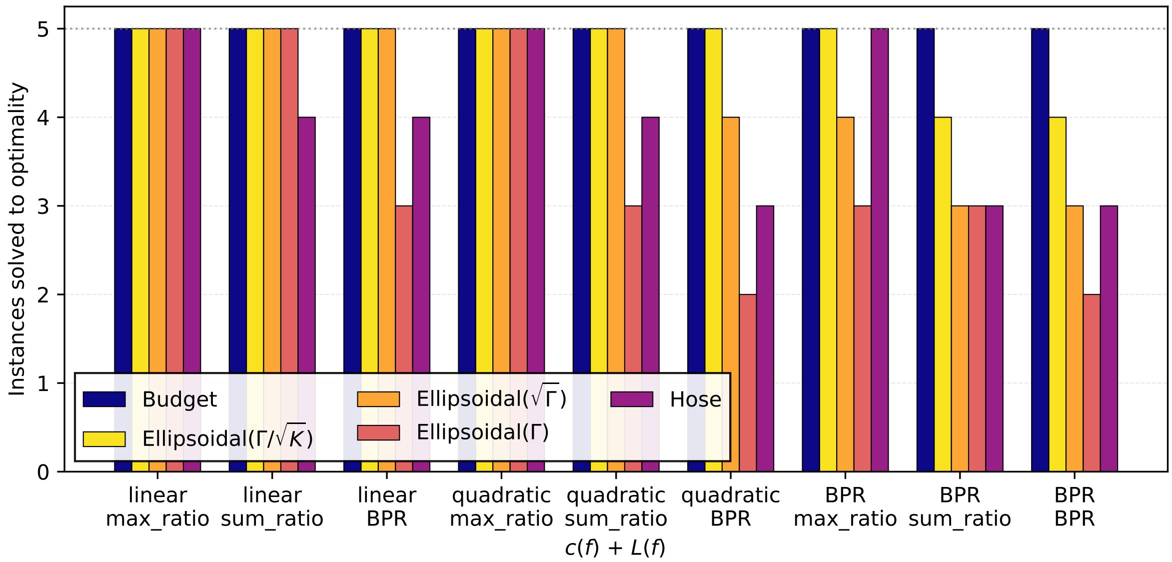


Figure 5.11. Number of instances solved to optimality by uncertainty set and function combination.

To ensure a fair comparison of computational times, we restrict the analysis to the 30 instances (out of 45) that were solved to optimality by all five uncertainty sets. Table 5.6 reports the average execution times for each configuration.
The budget uncertainty set achieves the fastest solution times across all function combinations. The ellipsoidal set with $\rho = \Gamma$ is the set requiring longest computation times, often exceeding other methods by two to three orders of magnitude. Notably, the combination of travel cost and latency functions has a significant impact on solution times: the BPR cost functions tend to yield longer execution times and lower success rates.

Impact on Worst-Case Congestion

We now assess the impact of each uncertainty set on worst-case congestion by comparing with the deterministic case (where demand equals the nominal value $\bar{d}$).
Figure 5.12 presents heatmaps of the percentage increase over the deterministic case for each uncertainty set, disaggregated by travel cost and latency function. First of all, we observe that all uncertainty sets lead to increased worst-case congestion levels compared to the deterministic scenario, with a clear hierarchy across sets. The lowest worst-case congestion is achieved by the ellipsoidal set with $\rho = \Gamma/\sqrt{|K|}$ (contained in the budget polyhedron), followed by the budget set, the ellipsoidal set with $\rho = \sqrt{\Gamma}$, the ellipsoidal set with $\rho = \Gamma$ (containing the budget polyhedron), and finally the hose model, which yields the highest worst-case congestion. This ordering is consistent with the geometric relationships established in Section 5.5.3. When comparing the impact of travel cost and latency functions, we observe distinct patterns. The heatmaps reveal that the `max_ratio` congestion measure leads to larger increases across all uncertainty sets, while `sum_ratio` yields the smallest deviations. The hose model and the largest ellipsoidal set exhibit the darkest colors, confirming that they yield higher worst-case congestion. It is interesting to observe that for the budgeted and the the budgeted and the smaller ellipsoidal uncertainty sets, the choice of the travel cost function has a limited effect on congestion increases. In contrast, a clearer dependence on the travel cost function emerges

Table 5.6. Average computational time (seconds) by uncertainty set and function combination. Times are computed only for the 30 instances solved to optimality by all five methods.

$c(f)$	$L(f)$	#solved	Budget	$\mathrm{E}(\frac{\Gamma}{\sqrt{\lvert K\rvert}})$	$\mathrm{E}(\sqrt{\Gamma})$	$\mathrm{E}(\Gamma)$	Hose
linear		**12**	**0.05**	**0.60**	**1.55**	**150.72**	**1.14**
	sum_ratio	4	0.01	0.11	1.29	3.90	1.62
	max_ratio	5	0.07	0.83	1.97	24.34	0.87
	BPR	3	0.06	0.87	1.18	557.09	0.94
quadratic		**10**	**1.20**	**9.93**	**33.30**	**446.52**	**5.54**
	sum_ratio	3	0.05	1.75	1.99	4.36	1.16
	max_ratio	5	2.34	18.65	65.19	884.85	9.99
	BPR	2	0.08	0.38	0.53	13.95	0.52
BPR		**8**	**0.05**	**1.76**	**1.68**	**42.07**	**21.34**
	sum_ratio	3	0.65	3.17	2.85	98.01	55.90
	max_ratio	3	0.72	1.12	1.11	12.92	0.47
	BPR	2	0.12	0.59	0.78	1.86	0.81

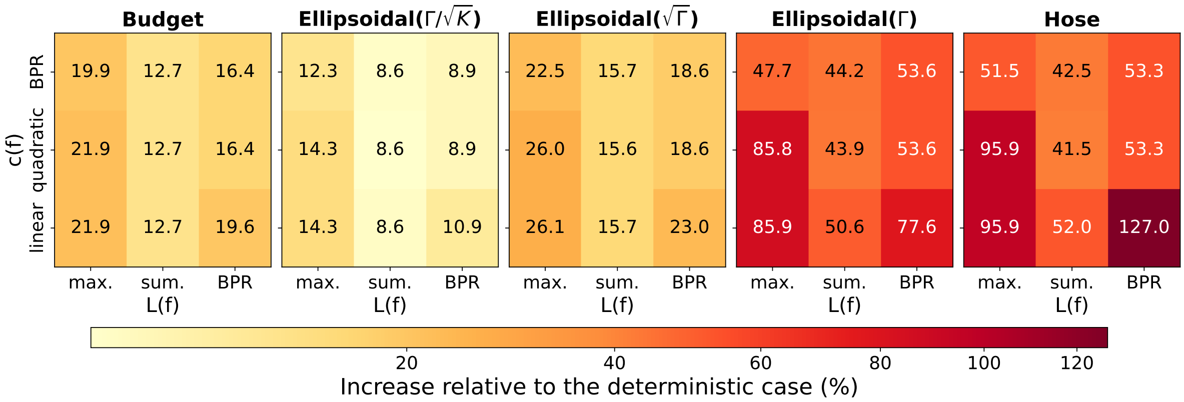


Figure 5.12. Percentage increase in worst-case congestion relative to the deterministic case, by uncertainty set and function combination. All panels share a common color scale for comparability. Darker colors indicate larger deviations from the deterministic solution.

Table 5.7. SNDlib instances: number of instances solved to optimality (out of 9) and average execution time (in seconds) by graph and demand level.

	10%		25%		50%		75%		100%	
Graph	#Opt	Time	#Opt	Time	#Opt	Time	#Opt	Time	#Opt	Time
abilene	9	2.2	9	7.0	9	11.9	6	24.9	6	28.0
atlanta	7	73.5	9	637.4	5	1993.4	3	29.9	4	1320.1
nobel-us	9	1.5	9	67.1	4	1094.1	3	88.3	3	67.7
pdh	9	0.6	9	7.2	6	48.5	3	17.0	3	43.8

for the larger ellipsoidal set and for the hose model. In particular, the hose model shows over 50% increase in worst-case congestion for most function combinations, with peaks exceeding 90% for the cases with linear and quadratic cost functions.

5.7.5 SNDlib Instances Results

To evaluate the behavior of our approach on heterogeneous network instances, we consider the SNDlib instances described in Section 5.6.2. These instances feature different network topologies and demand patterns compared to the Sioux Falls subnetworks, providing an independent validation of the observed trends.

Table 5.7 reports the number of solved instances and average execution time for each network and demand level, i.e., with different numbers of commodities. At low demand levels (10% and 25% of the total number of commodities), almost all instances are solved to optimality across all networks. As the number of commodities increases, the computational difficulty grows considerably. The atlanta network is the most challenging, with only 5 out of 9 instances solved at the 50% level and higher execution times even for the easiest cases. The nobel-us and pdh networks show a similar pattern: both are fully solved at low demand levels, but the success rate drops to 3 out of 9 at the 75% and 100% levels. In contrast, abilene remains the most tractable network, solving all instances up to the 50% level and maintaining a success rate of 6 out of 9 even at the highest demand levels.

Table 5.8 presents the computational performance by travel cost and latency function combination. Overall, 125 out of 180 instances (69.4%) are solved to optimality. The results confirm the trends observed on the Sioux Falls subnetworks: all 20 instances with linear travel costs are solved to optimality (100%). The success rate decreases to 65.0% for quadratic and 43.3% for BPR costs. Among the latency functions, sum_ratio tends to require longer execution times when combined with quadratic costs (916.9 seconds on average), while max_ratio remains faster across all cost functions.

Overall, these SNDlib instances are more challenging than the Sioux Falls subnetworks, especially when considering non-linear travel cost functions. This is likely due to the more complex topology of the SNDlib networks, specifically the absence of articulation points, which limits the effectiveness of some of the enhancement techniques described in Section 5.4.

5.7.6 Managerial Insights

The computational study yields several insights relevant for practitioners and network operators.

Table 5.8. Computational performance by travel cost and latency function combination. The table reports the number of instances solved to optimality and the average execution time (in seconds).

$c(f)$	$L(f)$	#Optimal	Time
linear	sum_ratio	20/20	39.3
	max_ratio	20/20	24.8
	BPR	20/20	22.2
quadratic	sum_ratio	14/20	916.9
	max_ratio	12/20	29.0
	BPR	13/20	538.7
BPR	sum_ratio	8/20	70.0
	max_ratio	9/20	577.7
	BPR	9/20	47.5

When selecting travel cost functions, congestion measures, routing policies and uncertainty sets for robust congestion modeling, practitioners must balance model realism, computational tractability, and solution quality.

The choice of travel cost function has a stronger impact on computational tractability than the choice of congestion measure. Linear cost functions provide the fastest solution times and highest success rates. BPR cost functions, while more realistic in capturing congestion effects, require longer solution times and may fail to reach optimality for large instances. For operational settings requiring frequent re-optimization, linear or quadratic costs offer a practical compromise between model realism and computational efficiency. Nevertheless, BPR-optimized flows produce congestion levels that are close to the worst-case across all latency measures, suggesting that BPR may serve as a robust default choice when the appropriate congestion metric is uncertain or when one aims to consider simultaneously the worst-case scenario across multiple congestion measures.

Among congestion measures, sum_ratio provides the best balance between computational efficiency and solution quality. The max_ratio measure, while effective at limiting extreme arc utilization, leads to more challenging optimization problems when combined with nonlinear cost functions.

The UE formulation requires 40–60% of the computational time needed by SO. This computational advantage makes UE preferable for real-time or repeated-solution applications. The Congestion Ratio between UE and SO typically remains close to 1, indicating that selfish routing does not lead to substantial efficiency losses/gains for most network configurations. However, configurations with max_ratio congestion measures show higher variability, suggesting that network operators concerned with peak congestion should consider coordinated routing strategies.

Finally, the analysis highlights the importance of considering uncertainty in congestion modelling. Accounting for demand uncertainty is essential: even the smallest uncertainty set (ellipsoidal with $\rho = \Gamma/\sqrt{|K|}$) increases worst-case congestion by at least 4.7% compared to the deterministic case. On average, solutions under uncertainty exhibit 11–72% higher congestion than their deterministic counterparts, depending on the choice of uncertainty set. This confirms that ignoring uncertainty can lead to significant underestimation of network congestion.

Table 5.9. Summary of uncertainty set performances: percentage of instances solved to optimality, average computational time (on commonly solved instances), and average percentage increase in worst-case congestion relative to the deterministic case.

Uncertainty Set	Solvability	Avg. Time (s)	Avg. Increase (%)
Budget (Γ-robust)	100%	0.56	17.7
Ellipsoidal ($\rho = \Gamma/\sqrt{\lvert K\rvert}$)	95.6%	4.02	11.1
Ellipsoidal ($\rho = \sqrt{\Gamma}$)	86.7%	12.17	20.9
Ellipsoidal ($\rho = \Gamma$)	68.9%	220.35	63.8
Hose	80.0%	7.96	72.3

The choice of uncertainty set involves a trade-off between robustness and computational tractability, as summarized in Table 5.9.

Budgeted uncertainty offers the most favorable balance: it provides protection against demand fluctuations with moderate increases in worst-case congestion, while achieving good solvability and the fastest computation times. Among ellipsoidal formulations, the smallest variant ($\rho = \Gamma/\sqrt{|K|}$) provides a favorable trade-off between tractability and worst-case congestion. The ellipsoidal set with $\rho = \sqrt{\Gamma}$ yield similar worst-case congestion levels to the budget uncertainty set, but the second achieves better computational tractability. Therefore, the budget set may be preferred when computational efficiency is a priority, accepting a moderate increase in worst-case congestion estimation. The hose model and the ellipsoidal set with $\rho = \Gamma$ yield the highest worst-case congestion, with average increases of 72% and 64%, respectively. These approaches should be reserved for applications where strong protection against correlated demand deviations is critical. However, practitioners should be aware of the reduced computational tractability of these formulations.

For typical operational planning, it may be preferable the combination of budgeted uncertainty with the `sum_ratio` congestion measure and linear or quadratic cost functions. This configuration provides robust solutions in reasonable computation times for networks with up to 100 origin-destination pairs. When peak congestion is the primary concern, the `max_ratio` measure should be used, with awareness that some solvability challenges may arise. For long-term planning where solution time is less constrained, BPR cost functions provide more realistic congestion modeling.

5.8 Conclusion

We study the problem of determining the worst-case congestion in a multi-commodity traffic network under demand uncertainty. Our aim is to stress-test a given network by identifying demand realizations and corresponding travelers' route choices that maximize congestion. To this end, we propose a novel bilevel formulation in which a traffic planner acts as the leader and the users of the traffic network act as the followers. In the leader's problem, we account for demand uncertainty using ideas from robust optimization, whereas the follower's problem models a traffic equilibrium in which travelers follow one of the two Wardrop principles—the user equilibrium (UE) or the system optimum (SO). We develop single-level mixed-integer nonlinear reformulations for the resulting congestion models that exploit binary variables and big-M constants, prove the existence of optimal solutions, derive valid big-Ms, and propose several enhancement techniques to strengthen the formulations. Moreover, we perform an extensive computational study on instances of the Sioux Falls network and the SNDlib, which provides several key insights.

First, we observe that the proposed enhancement techniques significantly improve computational performance, reducing solution times by over 90 % and enabling the solution of larger network instances that would otherwise remain unsolved within the time limit. Second, the choice of travel cost and latency functions has a substantial impact on both the computational tractability of the congestion models and the estimated worst-case congestion. In particular, we observe that the choice of the cost function has a stronger impact on the tractability of the model than the congestion measure. Whereas considering linear cost functions leads to the fastest solution times and the largest number of solved instances, the BPR cost function generally requires longer computation times but provides a conservative choice when the most suitable congestion measure is uncertain. Third, solving the UE formulation can be done considerably faster than solving the SO formulation, whereas congestion levels under the UE are typically very close to those under the SO. Indeed, by studying the congestion ratio, we show that the two equilibrium concepts yield similar worst-case congestion levels, suggesting that, under demand uncertainty, centralizing routing decisions, as in the SO, does not provide a significant advantage over the decentralized UE in terms of congestion reduction. We further note that, although the SO aims at minimizing the total travel time in the system, it does not always lead to less congestion compared to the UE in our framework. Fourth and finally, our analysis of different uncertainty sets—budgeted, ellipsoidal, and hose uncertainty—reveals a clear trade-off between conservatism and computational cost: the budgeted uncertainty model achieves the fastest solution times, whereas ellipsoidal and hose models provide more conservative estimates at the expense of increased computation time. Based on these findings, we derive managerial guidelines for choosing the most appropriate modeling of congestion, travel costs, and uncertainty.

This work provides a first step towards designing traffic networks that are more resilient to congestion under demand uncertainty. Building on our stress-testing framework, a natural next step is to embed our framework into a network design model, resulting in a robust bilevel optimization problem in which infrastructure decisions are made while anticipating traffic equilibrium responses under demand uncertainty.

Chapter 6

A Demand Modelling Pipeline for an Agent-Based Traffic Simulation of the City of Barcelona [1]

6.1 Introduction

In recent years, urban mobility has become increasingly complex due to several interrelated factors: *(i)* population growth and its concentration in cities, *(ii)* the proliferation of new mobility options and modes of transportation, and *(iii)* the evolution of citizens' habits and behaviors, including the shift toward e-commerce, car-sharing, and new work-from-home practices following the Covid-19 pandemic. Due to these changes, understanding how people move through cities — and how efficiently — becomes increasingly complex. Traffic patterns fluctuate throughout the day in ways that directly affect citizens' quality of life, contributing to greenhouse gas emissions, noise pollution, and disruptions to urban supply chains. For legislators, public administrations, network and logistics operators, and other stakeholders, gaining a clear picture of these dynamics is no longer optional but essential. It is the foundation for informed decision-making on interventions such as road network redesigns, restrictions on vehicle types, charging station placements, or public transport fleet dimensioning — all of which have direct implications for both urban mobility and supply chain performance and sustainability. The main challenge is that urban travel demand is *latent*: it cannot be directly observed at the individual level, yet it is precisely this individual-level demand—who travels, when, where, and by which mode—that determines the emergent congestion patterns observed on a network. Traditional analytical or aggregate models (e.g., four-step models) provide macro-level demand estimates but are unable to capture the rich behavioral heterogeneity and the feedback dynamics between individual choices and network conditions.

To address these problems, this work combines two elements: open-access data from the local government of Barcelona (the "Open Data BCN" repository and periodic statistical surveys on mobility and sociodemographics) with the open-source simulation tool MATSim. In particular,

[1] Based on the work presented in this chapter, a paper has been published in the proceedings of the 2023 Winter Simulation Conference (WSC 2023), the premier international conference on simulation. The paper is co-authored with Jonas F. Leon (Spindox S.A. / Universitat Oberta de Catalunya), Andrea Boccolucci (Spindox S.p.A.), and Mattia Neroni (Spindox S.p.A.) [85].

it involves constructing a complete, reproducible data pipeline to generate city-level, multi-modal travel demand and feeding it into an agent-based traffic simulation. Concretely, given a large metropolitan area (the city of Barcelona), the problem consists in:

1. synthesizing a *population* of agents endowed with realistic sociodemographic attributes (age, gender, employment status, home location), derived entirely from open-access census and survey data;

2. building a multi-modal *transportation network* (road, bus, metro, tram, walking, cycling) from open geographic sources;

3. generating individual *activity plans* for each agent—sequences of activities (home, work, education, leisure, health) with associated locations, timing, and travel modes—so that the aggregate demand matches observed mobility statistics;

4. calibrating the resulting simulation so that emergent traffic patterns replicate the real mobility behavior of the city.

The simulation engine used is MATSim [72], an open-source activity-based micro-simulator in which agents iteratively re-optimize their plans in response to experienced travel times, converging to a stochastic user equilibrium.
In terms of academic contributions, the generation of reliable transport demand profiles based on purely open-access data through a reproducible pipeline is lacking in the literature [73]. Furthermore, although partial models of the city of Barcelona exist, none covers multi-modal transport for the whole urban area. To the best of the authors' knowledge, this is the first work proposing a pipeline to synthesize transport demand in the city of Barcelona considering a multi-modal network and using purely open-access data and open-source tools.

The rest of the chapter is structured as follows. Section 6.2 presents a literature review on the key topics covered in this chapter. Section 6.3 describes the simulation tool and datasets used. Section 6.4 details the proposed modelling approach. Section 6.5 presents the calibration and results. Section 6.6 draws conclusions and outlines future research directions.

6.2 Related Work

The simulation of the traffic at a city level is a complex challenge that spans across multiple dimensions, such as the spatial coverage of the model, the level of detail to be considered or the interaction between the modeled components [144]. Historically, there have been different approaches to study urban mobility, ranging from macroscopic wave-like modelling, to detailed agent-based simulations [103]. Regardless of the simulation paradigm employed (micro-, meso- or macro-simulation, discrete-events or discrete-time, trip- or activity-based, etc.), researchers identify the same types of problems, namely: *(i)* data acquisition [15], *(ii)* model calibration and validation [32] and *(iii)* model scaling [90]. In recent years, activity-based micro-simulation seems to be gaining ground, since some of the aforementioned problems are diminishing due to the improved access to data at agent level or the increase in computational processing power [105]. The data acquisition has been simplified by the effort of state agencies, private companies and volunteers from the open community, that provide tools and data to the general public [109]. On the other hand, the increase in computational power has allowed the scientific community to exploit the advantages of agent-based traffic micro-simulations, such as their interpretability and adaptability by controlling individual interaction between agents [69].

In order to implement this type of traffic micro-simulation, there are numerous possible alternatives available to researchers and practitioners. There are some software dedicated to traffic micro-simulation like SUMO, MATSim or libraries like Python CityFlow. These software packages provide ready-to-use environments for simulating traffic and to handle road networks, vehicles, etc. Equally, powerful commercial options like Vissim, Aimsun or Paramics exist, but in spite of offering free-trial versions, their source code is proprietary, limiting reproducibility and community-based extensibility. Finally, generic agent-based simulation software can also be used to build traffic models, such as NetLogo, Python libraries like Mesa or Pandora, or commercial software like AnyLogic [106]. These non-specific agent-based software has the advantage of being more flexible, but they require much more effort in order to be able to describe the road network and the dynamic and interaction of the agents. In [93], a comprehensive review of different traffic simulation software is provided. For the particular case of MATSim, apart from enjoying the already mentioned micro-simulation advantages, its open modularity contributes also to expand its simulation capabilities. For instance, it allows incorporating the study of multi-modal public transportation [119], or the investigation of new mobility trends in urban areas, such as car-sharing [46].

For the city of Barcelona, there are some publications that have covered the study of its mobility dynamics using a micro-simulation description. In [6], researchers approached the modelling of Barcelona similarly as in the present work, but using SUMO as their simulation package and being limited to private cars as the only mode of transport. In [15], researchers employed MATSim for simulating the city of Barcelona, but the main data source for modeling the demand were mobile phone records. The results helped to study different policies in the Metropolitan Area of Barcelona. Finally, [107] considers a traffic flow simulation, modeled using AIMSUM, but it is limited to a single Barcelona district. Our simulation model, covering the multi-modal mobility of the entire city of Barcelona, can help public and private organizations to make better and more informed decisions, such as, for instance: where to place electric charging stations, which roads to open or close to traffic, or how to dimension the public bus fleet.

6.3 Data Sources and Simulation Tool

MATSim [72] is an open-source simulation framework that allows mobility analysis for large-scale scenarios, in the present case, at city level. In MATSim, individual agents (people) are modeled by assigning them a spatial location, some interaction rules and an activity plan (i.e., a series of time-scheduled activities) that they try to undertake. In aggregate, these individual agents constitute the population to be simulated (or a representative percentage of it), and their activity plans constitute the aggregated transport demand. In that sense, it could be considered an agent-based, and also an activity-based, simulation model. It can also be classified within the traffic micro-simulation paradigm, since the interaction of individual vehicles-like agents can be studied directly, following a queue-based approach as opposed to the more computationally expensive car-following behavior. The different agent plans compete with one another following a stochastic co-evolutionary algorithm that reaches an equilibrium after a certain number of iterations. In the equilibrium, every agent is carrying out its preferred plan to the best of its possibilities (e.g., without delays), and cannot further improve it without

deteriorating another agent's plan.
This study has been produced using both open-access data and open-source software, allowing the wider scientific community to consult and modify the code, and potentially to reproduce the pipeline presented in the following sections. Table 6.1 presents the main data sources that were employed to feed the demand pipeline.

Table 6.1. Open data sources.

Source	Description
A. OpenStreetMap	Open geographic database used to extract the nodes and links that compose the network of Barcelona used for the simulation.
B. Open Data BCN	Public repository of data for the city of Barcelona. It contains hundreds of datasets regarding population, transportation and services.
C. ESDB	Sociodemographic survey of Barcelona for the year 2020. It contains comprehensive social and economic data for the city of Barcelona.
D. EMEF	Working day mobility survey for the city of Barcelona (2021). The survey summarises the daily transportation habits of Barcelona's citizens.

6.4 Modelling Approach

In this section, the modelling approach followed is described in detail. In particular, the generation process for the three main ingredients of the simulation model is described, namely the network, the population and the transportation demand. The demand generation pipeline is graphically described in Figure 6.1. It is important to stress that the proposed pipeline is merging together very different and huge datasets, that are mostly related to spatial and sociodemographic distributions of people, and, after synthesizing a population of individual agents, trying to infer the mobility dynamic of an entire city.

6.4.1 Network Generation

In this section, the geographical scope of the simulation is defined, and the first two steps of the demand pipeline are explained. The steps consist in the generation of the network composed of nodes and links, and the localization of facilities within that network. The modelled area covers the approximately 101 km^2 of the urban area of Barcelona, and includes the 10 districts in which it is divided, with its corresponding 73 neighborhoods. Figure 6.2 shows a map view of the target extension of the simulation.

Nodes and links (Step 0)

The network is modeled as a direct graph, where each arc (i.e., link) represents a road or street, and each node represents a road junction. While a node is described solely by its

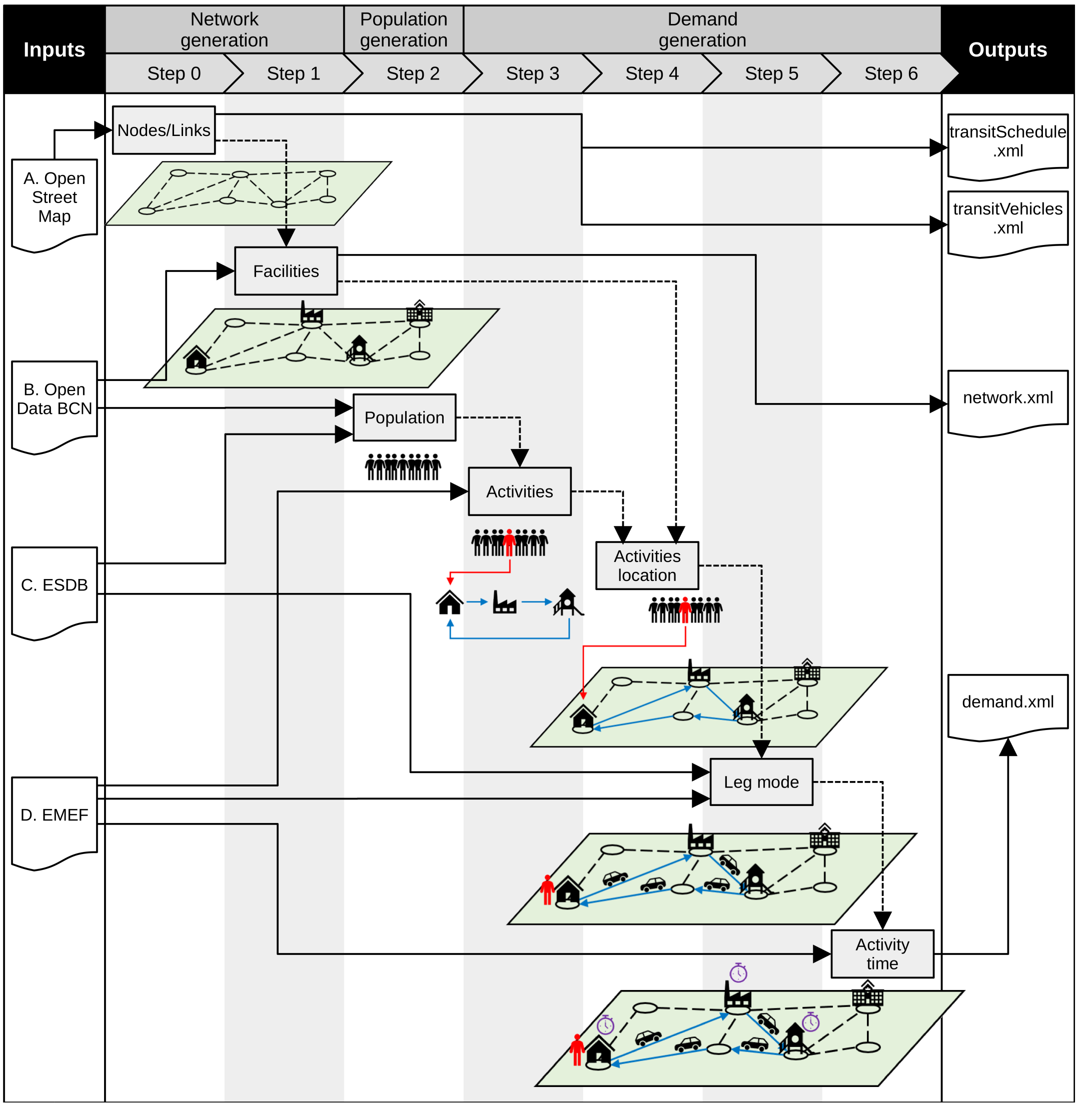


Figure 6.1. Demand generation pipeline. The steps followed are described in detail in Section 6.4 and the input data sources are contained in Table 6.1. Infographics are inspired and adapted from Simunto MATSim tutorials.

coordinates (using EPSG:2062 as spatial reference system), the links contain more information. Firstly, the link needs to indicate the two nodes that are connected, indicating the direction. The other characteristics relate to the length of the link, the maximum allowed speed, the maximum number of agents that can leave the link per unit time (called capacity), the number of lanes, the direction of the link, and the allowed means of transport in the link. Regarding the means of transport, an agent can move through a link on foot, by bicycle, by car or by public transport. In order to extract the spatial coordinates of nodes and links, the open-data community-maintained OpenStreetMap tool was employed. Some level of data processing was necessary for properly defining intersections and for classifying the type of road for the purpose of importing the data into MATSim. At the end, more than almost 100, 000 nodes and approximately 260, 000 links were included in the simulation model. The public transport

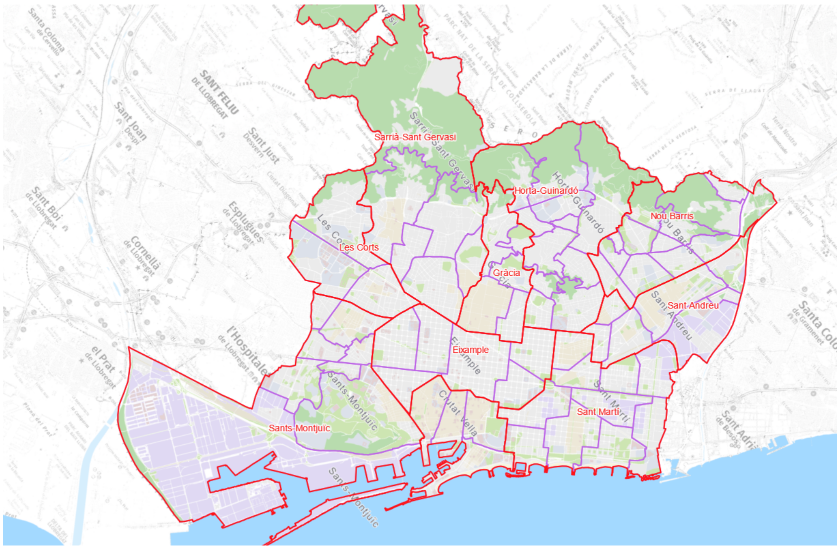


Figure 6.2. Map of the city of Barcelona, showing the districts and neighborhoods considered in the simulation. *Source*: Interactive map available at Barcelona City Council webpage: https://ajuntament.barcelona.cat/es/

considered, which include buses, underground (metro) and trams, were merged into the model network after obtaining the data from GTFS and employing the MATSim module pt2Matsim to properly format the data.

Facilities (Step 1)

Given Barcelona's network of nodes and arcs representing the city's streets, the first step is to enrich this network by adding "facilities", which represents the places where agents perform activities. The different types of facilities can be divided into six categories: (1) *Home*: Represents the places where the agents live, from where they start their daily activities and where they return at the end of the day; (2) *Work*: Represents the places where the agents work (office buildings, shops, factories...), normally places visited once a day, during a certain number of hours, by those agents that represent adult citizens that are employed; (3) *Education*: Represents the places (schools, universities, etc.) where younger agents, and those whose main occupation is studying, spend a number of hours every day, normally travelling there only once; (4) *Leisure*: Represents shopping centers, sport centers, parks or other facilities where all type of agents go from time to time to spend a variable amount of time for amusement; (5) *Pharma*: Represents hospitals, pharmacies, and other facilities where agents go to take care of their physical or mental health; (6) *Other*: This category was introduced to handle the parking procedure that is explained in Step 5 (Subsection 6.4.3). For defining the agent homes, random coordinates were taken within each residential area, according to the polygonal representation of the city of Barcelona. To define the facilities in which work, education, leisure and health-related activities take place, the Open Data Barcelona dataset reporting the economic activities census of the city of Barcelona was considered (see Table 6.1). At the end of this step, the list of facilities for each neighborhood, classified by the type of activity that can be carried out in it, is available.

6.4.2 Population Generation

In practical terms, generating the population means setting up a .xml file containing all the information regarding the agents to be considered in the simulation. The characteristics of each agent, such as their home location, their age and gender, have to be provided to the simulator. The process of generating the "synthetic" population to be used in the MATSim simulator is explained in this section, making also reference to the data sources employed. The city of Barcelona had approximately 1.6 million inhabitants in 2021. Simulating this amount of agents is not feasible if reasonable computational resources are to be employed, and for that reason, only the 1% of the total population was used. Using a percentage of the total population is a common approach since the aim is only to simulate the mobility flow and to study the congestion patterns in the city of Barcelona. In other words, since the model is interested in the global behavior of the agents, rather than in the interaction details in a specific area (e.g., a roundabout or a neighborhood), a subset of the population can be used as proxy of the total mobility configuration.

Sociodemographic profile (Step 2)

The second step in the pipeline is the creation of groups of agents per neighborhood, with basic sociodemographic attributes. These agents were generated by combining information contained in different datasets from Open Data Barcelona and from the "ESDB" Barcelona sociodemographic survey (see Table 6.1), specifically, the datasets related to: *(i)* Number and composition of households per neighborhood; *(ii)* Number of people by address; *(iii)* Population per neighborhood (according to gender and age); *(iv)* Level of employment and education. Having defined the agents and the neighborhood in which they live, it is possible to assign a home location (generated in Step 1 – see Section 6.4.1) to each agent, considering the household composition. At the end of this step, a list of agents with sociodemographic characteristics (gender, age, home location, occupation and education), is available.

6.4.3 Demand Generation

The last piece of information required to run the simulation is the travel demand. This demand represents the sequence of activities (and their associated movements through the network) that each agent wants to perform during the day. Therefore, the location, the start and end-time, and the mean of transport employed must be specified for each activity. Obviously, the agents' demand should reflect the mobility habits of the inhabitants of the city of Barcelona. It should be noted that the agents' plans can be subject to variations according to the traffic state of the network, which evolves following the stochastic co-evolutionary algorithm of MATSim, until an equilibrium state is reached. In fact, the activities' start times are set as the ideal start time according to agents' wishes, but the duration of the travel between two consecutive activities' locations may vary over time.

Activity chain (Step 3)

The third step consists in assigning an activity chain (i.e., a sequence of activities to be performed during the day) to each agent. The activity chains are defined using the "EMEF" Working Day mobility survey (see Table 6.1) as a baseline, on top of which more elaborated activity chains were added. All activity chains are a sequential combination of the six types of activities defined by their associated facilities, as described in Step 1 (Subsection 6.4.1). In general, an activity can be repeated more than once throughout the day. Some examples of activity chains are: *"Home-Work-Home"*; *"Home-Education-Home"*; *"Home-Work-Leisure-Home"*; *"Home-Education-Home-Leisure-Home"*; *"Home-Pharma-Home"*. It is important to note that the activity-chain-to-agent assignment considers the sociodemographic characteristics of the agent (e.g., an activity chain containing the activity "Work" will not be assigned to an agent without employment), and also respects the percentage of population carrying out each type of activity as included within the "EMEF" Working Day mobility survey. At the end of this step, the activity-chain has defined for the agents, which represents the skeleton of their daily plan. In the following steps, this initial plan will be enriched with detailed information on each activity and with the movements between the different activities in the plan.

Activity location (Step 4)

The fourth step determines where the agents will perform each activity of their plans. The activity-to-location assignment was done following the procedure detailed in Algorithm 2, where different sources of statistical data were combined. This intricate data processing was required because no direct source of information is publicly available for linking particular agents to locations. In fact, it would be difficult to obtain this type of information without compromising the privacy of agents, since the residence, workplace, or regularly visited venues will be linked to every agent, together with their typical schedule. The information is extracted mainly from the "ESDB" Sociodemographic survey of Barcelona (see Table 6.1), taking into account: *(i)* the probability of changing area/district within a trip depending on the origin, *(ii)* the average distances traveled per trip, and *(iii)* the average distances traveled by activity type. At the end of this step, the agents have each activity of their plan associated to a location.

Leg mode (Step 5)

The fifth step consists in choosing the mode of transport for each trip (or leg) that the agents must take to perform the activities of their plan. The modes considered were introduced in Subsection 6.4.1. For assigning the mode of transport to each leg, information extracted from "ESDB" Sociodemographic survey of Barcelona (see Table 6.1) was used. In particular, the statistical data referring to: *(i)* the average traveled distances by mode of transportation, *(ii)* the probability of using a mode as a function of the agent sociodemographic characteristics (mainly age and gender), *(iii)* the probability of using a mode as a function of the type of activity. To continue with the details regarding the choice of transportation mode, it is necessary to introduce the concept of "round-trip". A round-trip is defined as a subset of activities in a plan where the first and last activity is a *"Home"* type activity. For instance, the plan *"Home-Work-Home-Leisure-Home"*, contains two round-trips, namely: *"Home-Work-Home"* and *"Home-Leisure-Home"*. This is an important concept, since it was assumed that the agents can change their transportation mode only at the end of a round-trip. This means, for instance, that if the agents leave home by car, they cannot return home by public transport; but if they return home and leave again, the second trip can be made by public transport. This implies that the mode of transportation will be defined at the round-trip level. The procedure is detailed in Algorithm 3.

A computational problem arises when the selected mode of transportation is "car" but the

Algorithm 2 Activity-to-location assignment

Input: population of agents ▷ Each agent has its Home location and set of activities
Output: location for every agent activity

for every agent in population **do**
    for every agent activity **do**
        origin ← Home(agent)
        P ← probability of changing district(origin)
        d ← distance travelled(activity)
        if random(0,1) > P **then** ▷ Decide whether the agent changes district or not
            neighborhoods ← filter districts ≠ origin
        for every remaining neighborhood **do**
            if distance(origin, neighborhood) > d **then**
                neighborhoods ← filter out of reach
        for every remaining neighborhood **do**
            {neighborhood$_i$, p_i} ← assign probability based on number of facilities dedicated to (activity)
        destination neighborhood ← select random considering probabilities ({neighborhood$_i$, p_i})
        location ← select random in (destination neighborhood)
end

Algorithm 3 Mode-to-leg assignment

Input: population of agents ▷ Each agent has a plan (set of activities)
Output: mode for every round-trip

for every agent in population **do**
    for every agent activity plan **do**
        round-trip ← split(plan)
        for every round-trip **do**
            d_{max} ← calculate maximum distance(round-trip)
            modes ← remove modes based on(d_{max}) ▷ E.g., not reasonable to walk 25km daily
            for every remaining mode **do**
                {mode$_i$, p_i} ← assign probability based on(agent, activity)
                mode ← select random considering probabilities ({mode$_i$, p_i})
end

facility to be reached is on a link of the network that cannot be traveled by car. In order to overcome this problem, a "parking" procedure is invoked, introducing additional activities of type "Other" which represent parking locations and allow the route to be broken up into several parts. With this logic, the extreme links of the route can be covered on foot while the central links can be covered by car. An example of this is shown in Figure 6.3. At the end of this step, the agents' plans have a location associated to every activity and also the travel mode between each pair of activities.

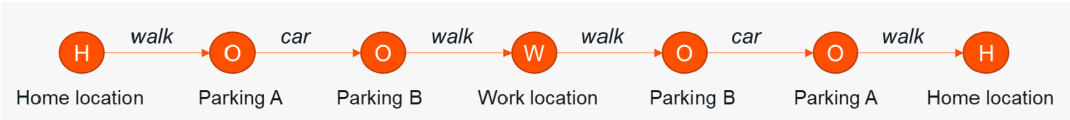


Figure 6.3. Result of applying the "parking" procedure to a *Home-Work-Home* activity plan, where the Work facility is not reachable by car.

Activity time (Step 6)

In this final step of the pipeline, the timing is added to each activity in an agent's plan. Technically, for each activity it is possible *(i)* to indicate its duration or *(ii)* to indicate an activity end time. This means that, after the agents arrive at a node and start an activity, they will remain at the node for the timespan indicated (the activity duration), or until the activity end time is reached, whichever comes first. Moreover, activities cannot overlap in time. The "EMEF" Working Day mobility survey (see Table 6.1) was used to assign an end time for every activity, considering the survey data and the total number of activities to be performed within the plan, i.e., time windows were defined to make sure that each activity's end-time leaves time to perform subsequent activities. For the "parking" activities explained in Step 5 (Subsection 6.4.3), a fixed activity duration was assumed. At the end of this step, the complete information for every agent's plan is available, namely the activities, with their associated location, mode of transport and timing.

6.5 Results: Calibration and Validation

The simulation pipeline described in previous sections contains a number of parameters that could be modified in order to calibrate the simulation. This calibration is an important activity that aims to adjust the simulation in order to reproduce the actual mobility behavior observed in the city as accurately as possible. The first pipeline calibration parameter considered was the possible set of activity chains, as indicated in Step 3 (Section 6.4.3). This means that, in order to better approximate the right number of movements per hour, some activity chains were modified, always maintaining the statistics at a macro-level. The second pipeline parameter tuned was the maximum distance d_{max}, used for defining the mode of transport in Step 5 (Section 6.4.3). This parameter was explicitly introduced in line 6 of Algorithm 3. On the other hand, some of the critical MATSim parameters to be tuned are the parameters related to the flow through the links and their storage capacity. These parameters, were adjusted to compensate for the reduction in the percentage of the simulated population, as detailed in Section 6.4.2. Also, the number of iterations considered for the co-evolutionary algorithm has an effect in the quality of the results. The results of the calibrated model can be seen in Figure 6.4.

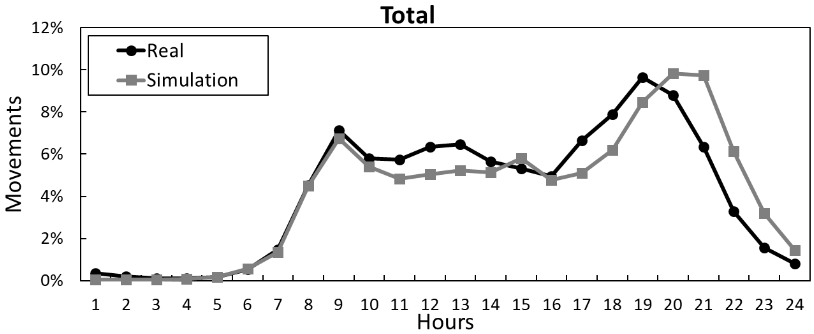


Figure 6.4. Comparison between the results of the Barcelona MATSim simulation against the real data provided in "EMEF" (see Table 6.1). The percentage of population that is travelling over the total is shown for each time bucket. Time bucket "1" covers from 00:00 until 00:59, bucket "2" covers from 01:00 until 01:59, and so on.

In terms of the mobility patterns, it can be observed that around 07:00 the transportation activity begins in the city of Barcelona, with a first traffic peak around 09:00, coinciding with the start of the activities in schools and work places. Throughout the central hours of the day, people move about in a somehow constant manner, probably reflecting the dynamism of a metropolis of the size of Barcelona, where different activities contribute to maintain a stable condition. Although the simulation captures this behavior, it underestimates the amount of movements in the city during this period. Around 16:00 there is a change in trend and the amount of journeys increases steadily up until around 20:00, when it starts to decrease at approximately the same rate. Once again, the simulation captures perfectly the dynamic of the mobility demand, but a slight delay can be observed in the second peak, which occurs later. In terms of intensity of the plateau area and the peaks and valleys, it can be seen that the simulation also matches the actual intensity levels.

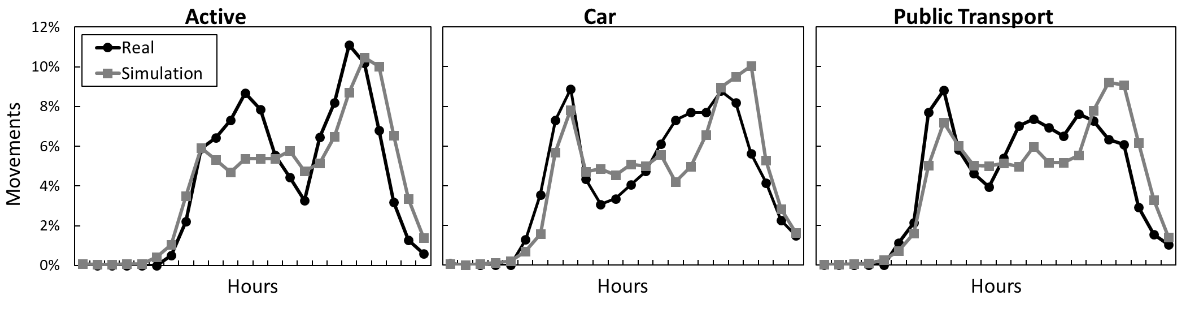


Figure 6.5. Comparison between the results of the Barcelona simulation against the real data provided in "EMEF" (see Table 6.1). The percentage of population that is travelling over the total is shown for each time bucket, and split by mode of transport. The time buckets are the same as in Figure 6.4.

Since one of the contributions of this study is the multi-modal capability of the MATSim implementation, the same comparison for population movements can be made segregating by mode of transport, as presented in Figure 6.5. Since these graphs are the decomposition of

Figure 6.4, the conclusions that can be extracted are very similar. If we analyze them one by one, it is interesting to observe that the simulation underestimates the use of active mobility (walking and biking) in the morning period. This underestimation is compensated with the use of cars and public transport, and for all of them it is clear the delay in the last part of the graph. That fact, namely that in the simulation there is a concentration of movements at the end of the day, could be explained by the way the activities are slotted in the pipeline. As explained in Step 6 of the pipeline (Section 6.4.3), activities can be of fixed duration or have a fixed start/end-time. Even considering well-defined times for the activities does not ensure that these take place as scheduled. In fact, the duration of travel between two consecutive activities is affected by at least two factors: *(i)* the traffic conditions on the network, that vary as a result of the co-evolutionary algorithm of MATSim; and *(ii)* the adjustment to the plan introduced by the parking procedure, which was necessary to ensure consistency in the data. Delays accumulate and this is reflected as a consequence on the latest activities of agents' plans, leading to more trips to be performed in the last hours of the day. This fact could be improved in a subsequent version of the model in order to distribute the activities end-times in such a way that reflects with more realism the actual mobility demand dynamic. Finally, in order to quantitatively evaluate the overall adjustment of the MATSim model to the real data, in Table 6.2, a summary of the gap between the obtained results and the real mobility data is provided for groups of time buckets and per mode of transport. Additionally, the Mean Average Percentage Error (MAPE) is calculated for each mode and also in total.

Table 6.2. Quantitative numerical results of the simulation model.

Mode of transport	Gap in time bucket group				MAPE
	1–6	7–12	13–18	19–24	
Active	1%	5%	3%	7%	37%
Car	0%	1%	6%	8%	24%
Public Transport	0%	3%	9%	12%	27%
Total	0%	3%	5%	8%	31%

As explained above, the gap increases towards the end of the day bucket groups, with values between 7% to 12%. The MAPE was used to quantify the difference between the actual and simulated values, obtaining figures close to 30%, which could be interpreted as a model accuracy level of roughly around 70%. This accuracy varies depending on the transport mode, being greater for the car than for the public transport and active mobility, probably reflecting the fact that the software logic was conceived initially for road traffic simulation. The presented model could greatly benefit from further efforts in order to increase its validity, specially when looking at street level traffic comparison. The match between simulation and reality at this maximum granularity level were difficult to obtain given the current state of the model.

6.6 Conclusions and Future Work

In this work, a detailed step-by-step pipeline for generating the demand for mobility for the city of Barcelona has been defined. The pipeline was entirely developed considering open-access data (mostly from the Barcelona City Council open-access initiatives) and open-source tools (MATSim simulator). The generated demand was represented by a population of agents with sociodemographic characteristics and a plan of activities to be carried out during the day. This

population of agents was feed into MATSim, which simulates the interaction between them, as they to move throughout the network of the city, using multiple modes of transportation, i.e. not only cars, but also bikes and public transport. From this agent-based micro-level interaction emerges the mobility patterns that characterize the city of Barcelona. The simulation model has been calibrated in order to reproduce the real behavior as faithfully as possible, with the final results showing a remarkable correspondence between the model and the reality.
A calibrated simulation model that replicates the dynamics of citizen mobility at the level of the city of Barcelona is an extremely useful tool on its own for researchers and decision-makers, and opens many future lines of research and application. One direct use of the simulator could consist in studying the impact of different public policies on the traffic profile of the city, such as the changes in people mobility as a consequence of restricting certain types of vehicles in some areas, changing the availability or the routes of some public transport lines, or moving the car-sharing service stations. Furthermore, health and sustainability aspects can be considered and the effects of different policies on CO2 emissions reductions evaluated. One very interesting line of research is considering the use of this tool in combination with other developed scientific fields, like optimization and operations research, where the simulator could be an essential part in advanced simulation-optimization algorithms. As concluding remarks, it is worth mentioning that the details provided for the demand pipeline construction, and the use of publicly available tools and dataset, should encourage other researchers to try and replicate this modelling methodology in other cities, where similar data is available. Also, the study's result has demonstrated that a synergetic alliance between public administrations and the research community can help towards creating powerful tools that empower citizens and companies towards smart city with the final goal of increase efficiency and reduce carbon emissions.

Acknowledgments

This work has been partially funded by Spindox and the Industrial Doctorate Program of the Catalan Government (2020 DI 116), the Spanish Ministry of Science (PID2019-111100RB-C21/AEI/ 10.13039/501100011033), and the Barcelona City Council – "la Caixa" Foundation under the framework of the Barcelona Science Plan 2020-2023 (grant 21S09355-001).

Bibliography

[1] B. Addis, G. Carello, A. Grosso, E. Lanzarone, S. Mattia, and E. Tanfani. Handling uncertainty in health care management using the cardinality-constrained approach: advantages and remarks. *Operations Research for Health Care*, 4:1–4, 2014.

[2] A. Agra and M. Santos. A robust uncapacitated lot-sizing problem under interval demand uncertainty with row-and-column generation. *European Journal of Operational Research*, 306(2):523–536, 2023.

[3] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.

[4] N. Aissaoui, M. Haouari, and E. Hassini. Supplier selection and order lot sizing modeling: A review. *Computers & Operations Research*, 34(12):3516–3540, 2007.

[5] A. Altın, E. Amaldi, P. Belotti, and M. Pınar. Provisioning virtual private networks under traffic uncertainty. *Networks*, 49(1):100–115, 2007.

[6] J. Argota Sánchez-Vaquerizo. Getting real: The challenge of building and validating a large-scale digital twin of barcelona's traffic with empirical data. *ISPRS International Journal of Geo-Information*, 11(1):24, 2022.

[7] P. Avella, S. Mattia, and A. Sassano. Metric inequalities and the network loading problem. *Discrete Optimization*, 4:103–114, 2007.

[8] A. Babaei, M. Khedmati, and M. R. Akbari Jokar. A branch and efficiency algorithm to design a sustainable two-echelon supply chain network considering traffic congestion and uncertainty. *Environmental Science and Pollution Research*, 30:28274–28304, 2023.

[9] A. Babaei, M. Khedmati, and M. R. Akbari Jokar. A new branch and efficiency algorithm for an optimal design of the supply chain network in view of resilience, inequity and traffic congestion. *Annals of Operations Research*, 321:49–78, 2023.

[10] T. Bacci, S. Mattia, and P. Ventura. The realization-independent reallocation heuristic for the stochastic container relocation problem. *Soft Computing*, 27:11529–11550, 2023.

[11] E. Balas, S. Ceria, and G. Cornuéjols. A lift-and-project cutting plane algorithm for mixed 0–1 programs. *Mathematical Programming*, 58(1–3):295–324, 1993.

[12] J. Banks. *Handbook on Simulation. Principles, Methodology, Advances, Applications and Practice.* Wiley, New York, 1998.

[13] J. F. Bard. *Practical Bilevel Optimization: Algorithms and Applications*. Springer, 1998.

[14] C. Basnet and J. M. Y. Leung. Inventory lot-sizing with supplier selection. *Computers & Operations Research*, 32(1):1–14, 2005.

[15] A. Bassolas, J. J. Ramasco, R. Herranz, and O. G. Cantú-Ros. Mobile phone records to feed activity-based travel demand models: Matsim for studying a cordon toll policy in barcelona. *Transportation Research Part A: Policy and Practice*, 121:56–74, 2019.

[16] Y. Beck, M. Labbé, and M. Schmidt. Toll setting with robust wardrop equilibrium conditions under budgeted uncertainty. Technical report, 2024.

[17] O. Ben-Ammar, B. Bettayeb, and A. Dolgui. Optimization of multi-period supply planning under stochastic lead times and a dynamic demand. *International Journal of Production Economics*, 218:106–117, 2019.

[18] O. Ben-Ammar, B. Bettayeb, I. Slama, A. Dolgui, and D. Ivanov. Multi-period replenishment planning with supplier assignment under a dynamic demand and stochastic lead-times. *International Journal of Production Research*, 0(0):1–25, 2025.

[19] O. Ben-Ayed, D. E. Boyce, and C. E. Blair. A general bilevel linear programming formulation of the network design problem. *Transportation Research Part B: Methodological*, 22(4):311–318, 1988.

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

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

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

[23] J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252, 1962.

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

[25] D. Bertsimas and M. Sim. Robust discrete optimization and network flows. *Mathematical Programming*, 98:49–71, 2003.

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

[27] D. Bertsimas and A. Thiele. A robust optimization approach to inventory theory. *Operations Research*, 54(1):150–168, 2006.

[28] D. Bienstock, S. Chopra, O. Günlük, and C.-Y. Tsai. Minimum cost capacity installation for multicommodity network flows. *Mathematical Programming*, 81(2):177–199, 1998.

[29] D. Bienstock and S. Mattia. Using mixed-integer programming to solve power grid blackout problems. *Discrete Optimization*, 4:115–141, 2007.

[30] J. R. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer, 2nd edition, 2011.

[31] G. R. Bitran and H. H. Yanasse. Computational complexity of the capacitated lot size problem. *Management Science*, 28(10):1174–1186, 1982.

[32] C. Bowman, J. A. Miller, and Y. Wang. Microscopic vehicular traffic simulation: Comparison of calibration techniques. In *Proceedings of the 2022 Winter Simulation Conference*, pages 2234–2245, Singapore, 2022. Institute of Electrical and Electronics Engineers, Inc.

[33] S. D. Boyles, N. E. Lownes, and A. Unnikrishnan. Transportation network analysis, volume i: Static and dynamic traffic assignment, 2025.

[34] N. Brahimi, N. Absi, S. Dauzère-Pérès, and A. Nordli. Single-item dynamic lot-sizing problems: An updated survey. *European Journal of Operational Research*, 263(3):838–863, 2017.

[35] N. Brahimi, S. Dauzère-Pérès, N. M. Najid, and A. Nordli. Single item lot sizing problems. *European Journal of Operational Research*, 168(1):1–16, 2006.

[36] L. Brotcorne, M. Labbé, P. Marcotte, and G. Savard. A bilevel model for toll optimization on a multicommodity transportation network. *Transportation Science*, 35(4), 2001.

[37] C. Buchheim. Bilevel linear optimization belongs to NP and admits polynomial-size KKT-based reformulations. *Operations Research Letters*, 51(6):618–622, 2023.

[38] G. P. Cachon and S. Netessine. Game theory in supply chain analysis. In *Handbook of Quantitative Supply Chain Analysis*, pages 13–65. Springer, 2004.

[39] L. E. Cárdenas-Barrón, J. O. Cunha, and R. A. Melo. Extended formulation and valid inequalities for the multi-item inventory lot-sizing problem with supplier selection. *Computers & Operations Research*, 130:105210, 2021.

[40] M. Cerulli, C. Archetti, E. Fernández, and I. Ljubić. A bilevel approach for compensation and routing decisions in last-mile delivery. *Transportation Science*, 58(5):1076–1100, 2024.

[41] D. Charypar and K. Nagel. Generating complete all-day activity plans with genetic algorithms. *Transportation*, 32(4):369–397, 2005.

[42] C. Chekuri, G. Oriolo, M. Scutellà, and B. Shepherd. Hardness of robust network design. *Networks*, 50(1):50–154, 2007.

[43] M. Chouman, T. G. Crainic, and B. Gendron. Commodity representations and cut-set-based inequalities for multicommodity capacitated fixed-charge network design. *Transportation Science*, 51(2):650–667, 2017.

[44] M. Christopher. *Logistics and Supply Chain Management*. Financial Times Prentice Hall, 3rd edition, 2005.

[45] J. Chu, K. Huang, and A. Thiele. A robust optimization approach to model supply and demand uncertainties in inventory systems. *Journal of the Operational Research Society*, 70(11):1885–1899, 2019.

[46] F. Ciari, N. Schuessler, and K. W. Axhausen. Estimation of carsharing demand using an activity-based microsimulation approach: Model discussion and some results. *International Journal of Sustainable Transportation*, 7(1):70–84, 2013.

[47] B. Colson, P. Marcotte, and G. Savard. Bilevel programming: A survey. pages 87–107.

[48] B. Colson, P. Marcotte, and G. Savard. An overview of bilevel optimization. *Annals of Operations Research*, 153(1):235–256, 2007.

[49] R. Cominetti, V. Dose, and M. Scarsini. Monotonicity of equilibria in nonatomic congestion games. *European Journal of Operational Research*, 316(2):754–766, 2024.

[50] J. R. Correa and N. E. Stier-Moses. Wardrop equilibria. In *Wiley Encyclopedia of Operations Research and Management Science*. Wiley, 2011.

[51] M. S. Daskin. *Network and Discrete Location: Models, Algorithms, and Applications*. John Wiley & Sons, 2nd edition, 2013.

[52] S. Dempe. *Foundations of Bilevel Programming*. Springer, 2002.

[53] S. Dempe, V. Kalashnikov, G. A. Pérez-Valdés, and N. Kalashnykova. *Bilevel Programming Problems: Theory, Algorithms and Applications to Energy Networks*. Springer, 2015.

[54] S. Dempe and A. Zemkoho. Bilevel road pricing: theoretical analysis and optimality conditions. *Annals of Operations Research*, 196:223–240, 2012.

[55] S. Dempe and A. B. Zemkoho. Bilevel optimization: theory, algorithms, applications and a bibliography. In *Bilevel Optimization: Advances and Next Challenges*, pages 581–672. Springer, 2020.

[56] S. Dewez, M. Labbé, P. Marcotte, and G. Savard. New formulations and valid inequalities for a bilevel pricing problem. *Operations Research Letters*, 36(2):141–149, 2008.

[57] N. G. Duffield, P. Goyal, A. Greenberg, P. Mishra, K. K. Ramakrishnan, and J. E. van der Merive. A flexible model for resource management in virtual private networks. In *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, SIGCOMM '99, page 95–108, New York, NY, USA, 1999. Association for Computing Machinery.

[58] M. C. Ferris and J. S. Pang. Engineering and economic applications of complementarity problems. *SIAM Review*, 39(4):669–713, 1997.

[59] J. Fingerhut, S. Suri, and J. S. Turner. Designing least-cost nonblocking broadband networks. *Journal of Algorithms*, 24(2):287–309, 1997.

[60] P. Fontaine and S. Minner. Benders decomposition for discrete–continuous linear bilevel problems with application to traffic network design. *Transportation Research Part B: Methodological*, 70:163–172, 2014.

[61] J. Fortuny-Amat and B. McCarl. A representation and economic interpretation of a two-level programming problem. *Journal of the Operational Research Society*, 32(9):783–792, 1981.

[62] V. Gabrel, C. Murat, and A. Thiele. Recent advances in robust optimization: An overview. *European Journal of Operational Research*, 235(3):471–483, 2014.

[63] D. J. Garcia and F. You. Supply chain design and optimization: Challenges and opportunities. *Computers & Chemical Engineering*, 81:153–170, 2015.

[64] A. M. Geoffrion. Generalized Benders decomposition. *Journal of Optimization Theory and Applications*, 10(4):237–260, 1972.

[65] A. Golmohammadi and E. Hassini. Review of supplier diversification and pricing strategies under random supply and demand. *International Journal of Production Research*, 58(11):3455–3487, 2020.

[66] R. E. Gomory. Outline of an algorithm for integer solutions to linear programs. *Bulletin of the American Mathematical Society*, 64(5):275–278, 1958.

[67] R. E. Gomory. An algorithm for the mixed integer problem. *Technical Report RM-2597, The RAND Corporation*, 1960.

[68] K. Govindan, M. Fattahi, and E. Keyvanshokooh. Supply chain network design under uncertainty: A comprehensive review and future research directions. *European Journal of Operational Research*, 263(1):108–141, 2017.

[69] R. J. Gräbe and J. W. Joubert. Are we getting vehicle emissions estimation right? *Transportation Research Part D: Transport and Environment*, 112:103477, 2022.

[70] P. Hansen, B. Jaumard, and G. Savard. New branch-and-bound rules for linear bilevel programming. *SIAM Journal on Scientific and Statistical Computing*, 13(5):1194–1217, 1992.

[71] F. Hnaien and H. M. Afsar. Robust single-item lot-sizing problems with discrete-scenario lead time. *International Journal of Production Economics*, 185:223–229, 2017.

[72] A. Horni, K. Nagel, and K. W. Axhausen. *The Multi-Agent Transport Simulation MATSim.* Ubiquity Press, 2016.

[73] S. Hörl and M. Balac. Synthetic population and travel demand for paris and Île-de-france based on open and publicly available data. *Transportation Research Part C: Emerging Technologies*, 130:103291, 2021.

[74] Y. Ito. Robust wardrop equilibria in the traffic assignment problem with uncertain data, 2011.

[75] V. Kalashnikov, J. G. Flores Muñiz, and N. Kalashnykova. *Bilevel Optimal Tolls Problems with Nonlinear Costs: A Heuristic Solution Method*, pages 481–516. Springer International Publishing, Cham, 2020.

[76] P. Kall and S. Wallace. *Stochastic Programming.* Wiley-Interscience, New York,, 1994.

[77] T. Kleinert, M. Labbé, I. Ljubić, and M. Schmidt. A survey on mixed-integer programming techniques in bilevel optimization. *EURO Journal on Computational Optimization*, 9:100007, 2021.

[78] T. Kleinert, M. Labbé, F. Plein, and M. Schmidt. There's no free lunch: On the hardness of choosing a correct big-M in bilevel optimization. *Operations Research*, 68(6):1716–1733, 2020.

[79] E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. In *Proceedings of the 16th Annual Conference on Theoretical Aspects of Computer Science*, STACS'99, pages 404–413, Berlin, Heidelberg, 1999. Springer-Verlag.

[80] P. Kouvelis and G. Yu. *Robust Discrete Optimization and its Applications.* Kluwer Academic Publisher, 1997.

[81] M. Labbé, P. Marcotte, and G. Savard. A bilevel model of taxation and its application to optimal highway pricing. *Management Science*, 44(12.1), 1998.

[82] A. H. Land and A. G. Doig. An automatic method of solving discrete programming problems. *Econometrica*, 28(3):497–520, 1960.

[83] L. J. LeBlanc and D. E. Boyce. A bilevel programming algorithm for exact solution of the network design problem with user-optimal flows. *Transportation Research Part B: Methodological*, 20(3):259–265, 1986.

[84] L. J. LeBlanc, E. K. Morlok, and W. P. Pierskalla. An efficient approach to solving the road network equilibrium traffic assignment problem. *Transportation Research*, 9(5):309–318, 1975.

[85] J. Leon, F. Giancola, A. Boccolucci, and M. Neroni. A demand modelling pipeline for an agent-based traffic simulation of the city of barcelona. pages 1771–1782, 12 2023.

[86] C. Li and I. E. Grossmann. A review of stochastic programming methods for optimization of process systems under uncertainty. *Frontiers in Chemical Engineering*, 2:622241, 2021.

[87] Z. Li, R. Ding, and C. A. Floudas. A comparative theoretical and computational study on robust counterpart optimization: I. robust linear optimization and robust mixed integer linear optimization. *Industrial & Engineering Chemistry Research*, 50(18):10567–10603, 2011. PMID: 21935263.

[88] M. Lu and Z. Shen. A review of robust operations management under model uncertainty. *Production and operations management*, 2020.

[89] Z.-Q. Luo, J.-S. Pang, and D. Ralph. *Mathematical Programs with Equilibrium Constraints.* Cambridge University Press, 1996.

[90] Z. Ma and M. Fukuda. A multi-agent spatial simulation library for parallelizing transport simulations. In *Proceedings of the 2015 Winter Simulation Conference*, pages 115–126, Huntington Beach, CA, 2015. Institute of Electrical and Electronics Engineers, Inc.

[91] T. L. Magnanti and R. T. Wong. Accelerating benders1962partitioning decomposition: Algorithmic enhancement and model selection criteria. *Operations Research*, 29(3):464–484, 1981.

[92] T. L. Magnanti and R. T. Wong. Network design and transportation planning: Models and algorithms. *Transportation Science*, 18(1):1–55, 1984.

[93] K. Mahmud and G. E. Town. A review of computer tools for modeling electric vehicle energy requirements and their impact on power distribution networks. *Applied Energy*, 172:337–359, 2016.

[94] P. Marcotte. Network design problem with congestion effects: A case of bilevel programming. *Mathematical Programming*, 34(2):142–162, 1986.

[95] S. Mattia. The robust network loading problem with dynamic routing. *Computational Optimization and Applications*, 54(3):619–643, 2013.

[96] S. Mattia. The cut property under demand uncertainty. *Networks*, 66(2):159–168, 2015.

[97] S. Mattia. A polyhedral analysis of the capacitated edge activation problem with uncertain demands. *Networks*, 74(2):190–204, 2019.

[98] S. Mattia and M. Poss. A comparison of different routing schemes for the robust network loading problem: polyhedral results and computation. *Computational Optimization and Applications*, 69(3):753–800, 2018.

[99] S. Mattia, F. Rossi, M. Servilio, and S. Smriglio. Staffing and scheduling flexible call centers by two-stage robust optimization. *Omega*, 72:25–37, 2017.

[100] J. T. Mentzer, W. DeWitt, J. S. Keebler, S. Min, N. W. Nix, C. D. Smith, and Z. G. Zacharia. Defining supply chain management. *Journal of Business Logistics*, 22(2):1–25, 2001.

[101] P. Metzker Soares, S. Thevenin, Y. Adulyasak, and A. Dolgui. Adaptive robust optimization for lot-sizing under yield uncertainty. *European Journal of Operational Research*, 313(2):513–526, 2024.

[102] A. Migdalas. Bilevel programming in traffic planning: Models, methods and challenge. volume 7, pages 381–405. 1995.

[103] J. A. Miller, H. Peng, and C. N. Bowman. Advanced tutorial on microscopic discrete-event traffic simulation. In *Proceedings of the 2017 Winter Simulation Conference*, pages 705–719, Las Vegas, NV, 2017. Institute of Electrical and Electronics Engineers, Inc.

[104] S. Minner. Multiple-supplier inventory models in supply chain management: A review. *International Journal of Production Economics*, 81–82:265–279, 2003.

[105] R. Moeckel, N. Kuehnel, C. Llorca, A. T. Moreno, and H. Rayaprolu. Agent-based simulation to improve policy sensitivity of trip-based models. *Journal of Advanced Transportation*, 2020:1902162, 2020.

[106] S. Mohebbi and P. S. Murali. Spatial agent-based simulation of connected and autonomous vehicles to assess impacts on traffic conditions. In *Proceedings of the 2022 Winter Simulation Conference*, pages 2487–2498, Singapore, 2022. Institute of Electrical and Electronics Engineers, Inc.

[107] L. Montero, M. P. Linares, O. Serch, and J. Casanovas-Garcia. A visualization tool based on traffic simulation for the analysis and evaluation of smart city policies, innovative vehicles and mobility concepts. In *Proceedings of the 2017 Winter Simulation Conference*, pages 3196–3207, Las Vegas, NV, 2017. Institute of Electrical and Electronics Engineers, Inc.

[108] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*. John Wiley & Sons, 1988.

[109] F. T. Neves, M. de Castro Neto, and M. Aparicio. The impacts of open data initiatives on smart cities: A framework for evaluation and monitoring. *Cities*, 106:102860, 2020.

[110] F. Ordóñez and N. E. Stier-Moses. Robust wardrop equilibrium. In T. Chahed and B. Tuffin, editors, *Network Control and Optimization*, pages 247–256, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[111] F. Ordóñez and N. E. Stier-Moses. Wardrop equilibria with risk-averse users. *Transportation Science*, 44(1):63–86, 2010.

[112] S. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessäly. SNDlib 1.0–Survivable Network Design Library. In *Proceedings of the 3rd International Network Optimization Conference (INOC 2007), Spa, Belgium*, 2007.

[113] S. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessäly. SNDlib 1.0–Survivable Network Design Library. *Networks*, 55(3):276–286, 2010.

[114] M. Padberg and G. Rinaldi. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM Review*, 33(1):60–100, 1991.

[115] M. Patriksson. *The Traffic Assignment Problem: Models and Methods*. Dover Publications, 2015.

[116] D. Peidro, J. Mula, R. Poler, and F.-C. Lario. Quantitative models for supply chain planning under uncertainty: a review. *The International Journal of Advanced Manufacturing Technology*, 43(3–4):400–420, 2009.

[117] S. Pineda and J. M. Morales. Solving linear bilevel problems using big-Ms: Not all that glitters is gold. *IEEE Transactions on Power Systems*, 34(4):2469–2481, 2019.

[118] Y. Pochet and L. A. Wolsey. *Production Planning by Mixed Integer Programming*. Springer Series in Operations Research and Financial Engineering. Springer, New York, 2006.

[119] F. Poletti, P. M. Bösch, F. Ciari, and K. W. Axhausen. Public transit route mapping for large-scale multimodal networks. *ISPRS International Journal of Geo-Information*, 6(9):268, 2017.

[120] R. Rahmaniani, T. G. Crainic, M. Gendreau, and W. Rei. The Benders decomposition algorithm: A literature review. *European Journal of Operational Research*, 259(3):801–817, 2017.

[121] D. Rey and M. W. Levin. A branch-and-price-and-cut algorithm for discrete network design problems under traffic equilibrium. Technical report, 2024.

[122] S. Ross. *Simulation*. Academic Press, San Diego, 2002.

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

[124] J. K. Sankaran, A. Gore, and B. Coldwell. The impact of road traffic congestion on supply chains: Insights from Auckland, New Zealand. *International Journal of Logistics: Research and Applications*, 8(2):159–180, 2005.

[125] A. Schrijver. *Combinatorial Optimization*. Springer-Verlag Berlin Heidelberg, 2003.

[126] A. Shapiro, D. Dentcheva, and A. Ruszczyński. *Lectures on Stochastic Programming: Modeling and Theory*. SIAM, 3rd edition, 2021.

[127] M. Sim. Robust optimization.

[128] D. Simchi-Levi, X. Chen, and J. Bramel. *The Logic of Logistics: Theory, Algorithms, and Applications for Logistics Management*. Springer, 3rd edition, 2014.

[129] A. L. Soyster. Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research*, 21(5):1154–1157, 1973.

[130] H. Stadtler, C. Kilger, and H. Meyr. *Supply Chain Management and Advanced Planning: Concepts, Models, Software, and Case Studies*. Springer, 5th edition, 2015.

[131] S. Terzi and S. Cavalieri. Simulation in the supply chain context: a survey. *Computers in Industry*, 53(1):3–16, 2004.

[132] S. Thevenin, Y. Adulyasak, and A. Dolgui. Material requirements planning under stochastic lead times using stochastic programming. *Production and Operations Management*, 31(6):2556–2575, 2022.

[133] S. Thevenin, O. Ben-Ammar, and N. Brahimi. Robust optimization approaches for purchase planning with supplier selection under lead time uncertainty. *European Journal of Operational Research*, 303:1199–1215, 2022.

[134] A. Thorsen and T. Yao. Robust inventory control under demand and lead time uncertainty. *Annals of Operations Research*, 257(1–2):207–236, 2017.

[135] L. Tian, A. Bashan, D.-N. Shi, and Y.-Y. Liu. Articulation points in complex networks. *Nature Communications*, 8(14223), 2017.

[136] P. Toth and D. Vigo. *Vehicle Routing: Problems, Methods, and Applications*. SIAM, 2nd edition, 2014.

[137] W. Tutte. Lectures on matroids. *Journal of Research National Bureau of Standards Section B*, 69:1–47, 1695. reprinted in: "Selected Papers of W.T. Tutte", Vol. II (D. McCarthy, R.G. Stanton, eds.), Charles Babbage Research Centre, St. Pierre, Manitoba, 1979, p. 439-496.

[138] U.S. Bureau of Public Roads. *Traffic assignment manual.* Washington, D.C., 1964.

[139] H. M. Wagner and T. M. Whitin. Dynamic version of the economic lot size model. *Management Science*, 5(1):89–96, 1958.

[140] J. G. Wardrop. Some theoretical aspects of road traffic research. *Proceedings of the Institution of Civil Engineers*, 1(3):325—-362, 1952.

[141] J. G. Wardrop and J. I. Whitehead. Correspondence. some theoretical aspects of road traffic research. *ICE Proceedings: Engineering Divisions*, 1(5):767, 1952.

[142] G. Weisbrod and S. Fitzroy. Traffic congestion effects on supply chains: Accounting for behavioral elements in planning and economic impact models. In S. Renko, editor, *Supply Chain Management – New Perspectives.* IntechOpen, 2011.

[143] D. B. West. *Introduction to Graph Theory.* Pearson Education, Inc., 2nd edition edition, 2001.

[144] J. Weyl, U. A. Lenfers, T. Clemen, D. Glake, F. Panse, and N. Ritter. Large-scale traffic simulation for smart city planning with mars. In *Proceedings of the 2019 Summer Simulation Conference*, July 22$^{\text{nd}}$-24$^{\text{th}}$, Berlin, Germany, pages 1–12, San Diego, CA, USA, 2019. Society for Computer Simulation International.

[145] L. Wolsey. Progress with single-item lot-sizing. *European Journal of Operational Research*, 86:395–401, 1995.

[146] L. A. Wolsey. *Integer Programming.* John Wiley & Sons, 1998.

[147] İ. Yanıkouglu, B. L. Gorissen, and D. den Hertog. A survey of adjustable robust optimization. *European Journal of Operational Research*, 277(3):799–813, 2019.

[148] W. I. Zangwill. A backlogging model and a multi-echelon model of a dynamic economic lot size production system—a network approach. *Management Science*, 15(9):506–527, 1969.

[149] Y. Zhao and D. Klabjan. A polyhedral study of lot-sizing with supplier selection. *Discrete Optimization*, 9(2):65–76, 2012.