Université catholique de Louvain
Louvain School of Management

Louvain Institute for Data Analysis and Modeling in economics and statistics

Center for Operations Research and Econometrics

Models, Methods, and Algorithms in Operational Research: Applications to Inventory Control, Energy Infrastructure, and Computational Biology

Henri Dehaybe

Thesis submitted in partial fulfillment of the requirements for the degree of
Docteur en sciences économiques et de gestion

Supervisors:
Daniele Catanzaro (UCLouvain, Belgium)
Philippe Chevalier (UCLouvain, Belgium)

Jury:
Per J. Agrell (UCLouvain, Belgium)
Quentin Cappart (Polytechnique Montreal, Canada)
Raffaele Pesenti (Università Ca' Foscari Venezia, Italy)

PHD ORGANIZATION

Candidate

Henri Dehaybe
Teaching Assistant, UCLouvain
Louvain School of Management
Center for Operations Research and Econometrics

Thesis Supervisors

Daniele Catanzaro
Associate Professor, UCLouvain
Louvain School of Management
Center for Operations Research and Econometrics
Philippe Chevalier
Associate Professor, UCLouvain
Louvain School of Management
Center for Operations Research and Econometrics

Supervisory Committee

Pierre Schaus
Associate Professor, UCLouvain
Institute of Information and Communication Technologies, Electronics and Applied Mathematics
Mathieu Van Vyve
Associate Professor, UCLouvain
Louvain School of Management
Center for Operations Research and Econometrics

ACKNOWLEDGMENTS

I would like to extend my sincere gratitude to my two supervisors, Daniele and Philippe, for the time and trust you have placed in me. Despite being the two busiest people I know, you always made sure to find time for me whenever I asked. Daniele, thank you for always pushing me to do my best, and for steering me toward this PhD program in the first place. While it comes with a certain amount of pressure, the ambition you have for me has gone a long way in building my self-confidence. Philippe, it's safe to say that I would not be writing these lines if it weren't for you offering me the opportunity to join the PreSupply project. So to both of you, thank you for making this accomplishment possible.

My sincere thanks go to the members of my jury for the time and interest they devoted to my thesis. Your constructive feedback and our discussions were truly appreciated, and I look forward to our future interactions.

I am also grateful to my other co-authors, Per, Manuel, and Raffaele, with whom I had the great pleasure of working. I look forward to our future discussions and to learning from your valuable experience in potential upcoming projects.

I would also like to thank my parents. I've had the privilege of growing up in a stable, supportive, and loving family, something not everyone is fortunate enough to experience. You climbed the social ladder so I wouldn't have to, gave me your time and help whenever I needed it, and always encouraged me to reach my potential. You may not have contributed directly to my research, but you have been absolutely essential to my journey.

Marie, thank you for having accompanied me for the most part of this important chapter of my life. Thank you for your kindness, for your support and encouragements when I needed it, and for making me a better person than I was before I met you. Your presence will always be a part of who I've become.

Finally, I'm grateful to my wonderful colleagues from both CORE and LSM, not only for any help and advice you may have offered, but also for the quality moments we shared along the way.

Everyone you meet is fighting a battle you know nothing about. Be kind. Always.

— Ian Maclaren, *The British Weekly*, Christmas 1897 (paraphrased)

CONTENTS

1 Introduction **1**

1.1 General introduction and context 1

1.2 Background on deep reinforcement learning 3

1.2.1 Tabular reinforcement learning 4

1.2.2 Deep Q-Learning 6

1.2.3 Proximal Policy Optimization 7

1.3 Processing of imperfect model parameter data 10

1.3.1 Fixing graph topologies for pipeline networks 10

1.3.2 Resolving capacity constraints with minimal assumptions 12

1.4 Background on phylogeny inference 13

1.4.1 Phylogenies 13

1.4.2 Phylogeny inference is an optimization problem 14

1.5 Structure of the thesis 17

2 Deep Reinforcement Learning for Inventory Optimization with Non-Stationary Uncertain Demand **19**

2.1 Introduction 19

2.2 Related work 21

2.3 Problem statement 23

2.3.1 An approximate dynamic programming algorithm for the SISLSP 24

2.3.2 The Martingale model of forecast evolution 25

2.4 A DRL approach to non-stationary versions of the SISLSP . . 27

2.4.1 Encoding the SISLSP environment 27
2.4.2 A toy example 30
2.4.3 The Proximal Policy Optimization algorithm 31
2.4.4 Learning under fixed order costs 32
2.5 Computational experiments 33
2.5.1 Implementation details 34
2.5.2 Near-optimality 36
2.5.3 Updating forecasts 40
2.5.4 Lost sales 41
2.5.5 Visualization of the learned policies 42
2.6 Conclusion 44
2.7 Future research 45
2.A Appendix: Figures 46

3 Balancing Supply Security and Decarbonization: Optimizing Germany's LNG port Infrastructure under the European Green Deal 53

3.0.1 Germany in the European Energy Union 55
3.1 Objective 56
3.2 Relevant literature 58
3.3 Model 61
3.4 Application 63
3.4.1 Data collection 63
3.4.2 Demand estimation 65
3.4.3 Import and export 66
3.4.4 Domestic production 67
3.4.5 Market prices 67
3.4.6 Graph construction 68
3.5 Results 69
3.5.1 Greenfield scenarios 69
3.5.2 Brownfield scenario 73
3.5.3 No-FSRU scenario 74
3.5.4 Peace scenario 74
3.5.5 Sensitivity analysis 74
3.6 Comparison with actual planning in Germany 75

3.7 Conclusions and policy implications 76
3.7.1 Input on current policy discourse 78
3.7.2 Lessons for policy makers 78
3.7.3 Limitations and future research 79

4 New Heuristics for Phylogeny Estimation under the Balanced Minimum Evolution Criterion 81
4.1 Introduction 81
4.2 Some background on the TRP and the BMEP 83
4.3 A matheuristic for the BMEP 85
4.3.1 Agglomeration methods 85
4.3.2 The LPNJ algorithm 87
4.3.3 Beam search 89
4.4 Computational experiments 92
4.4.1 Data sets 92
4.4.2 Experimental setup 92
4.4.3 Results 92
4.5 Final remarks and open questions 94
4.6 Future research 94

5 Conclusion 97

LIST OF FIGURES

1.1 One transition of an MDP at time step t. 5

1.2 Left: toy original graph with two disconnected components ($\{A\}$ and $\{B,C\}$) and two unreachable nodes, E and F, in the main component. Right: The same graph repaired with added virtual pipelines (dashed arcs) and virtual bi-directionalities (hollow arrowheads). 11

1.3 Tree representation of a phylogeny rooted in R for a set of six taxa. This tree features a polytomy at the Common Ancestor (CA) $\{C,D\}$. The branches of the tree are weighted, including one null branch indicating no dissimilarity between E and its parent. 14

1.4 Unweighted phylogeny of a set of seven viruses. This tree is the optimal UBT obtained from solving the BMEP. See Catanzaro et al. (2022) for details on the computation of this solution. Reproduced from Catanzaro et al. (2022) with permission. . . . 14

1.5 Example of a most parsimonious phylogeny represented an UBT. The leaves (red) correspond to the six taxa and the internal vertices (blue) to the inferred common ancestors. The edge weights (orange) are the Hamming distances between the vertices. Reproduced from Catanzaro et al. (2022) with permission. 15

1.6 An edge connecting two internal vertices u and v and the four incident subtrees represented as red triangles. Adapted from Catanzaro et al. (2022) with permission. 16

2.1 Architecture of the actor neural network. 35

2.2 Whisker plots of the median agent gaps with respect to the DP baseline for the near-optimality experiments with varying horizons. Each data point represents one agent, each whisker plot thus represents ten agents. 40

2.3 Evolution over time periods of the (s, S) predicted by an agent (blue) and the baseline DP policy (red). 43

2.4 Whisker plots of the median agent gaps with respect to the DP baseline for the near-optimality experiments. Each data point represents one agent, each whisker plot thus represents ten agents. 47

2.5 Whisker plots of the gap of the agents for each forecast with respect to the DP baseline. Each data point represents one test instance, each whisker plot thus represents the performance one the respective forecast trends of each of the 120 agents (backorders or lost sales). 48

2.6 Whisker plots of the median agent gaps for the Updating Forecasts experiments with varying MMFE parameters. The gaps are computed with respect to the average cost of the dynamic programming (s_t, S_t) policy of each instance, recomputed at each step to account for the updated forecast. Each data point represents one agent, each whisker plot thus represents ten agents. 49

2.7 Whisker plots of the median gap for the Lost Sales experiments for varying environment parameters. The gaps are computed with respect to the average cost of the DP policy of each instance. Each data point represents one instance, each whisker plot thus represents ten instances. 50

2.8 Comparison of the DP (s, S) policies (red) and learned deep RL policies (blue) in a backorders environment (left) and a lost sales environment (right). The dashed and dash-dotted lines are the predicted (s, S) parameters respectively, and the solid lines are the resulting order quantities with respect to the inventory position. 51

2.9 Heatmaps of the gradients of s (top) and S (bottom) with respect to the forecast input for each forecast trend. The agents inspected are those used in Figure 2.8a (left) and Figure 2.8b (right), trained in backorders and lost sales environments respectively. 52

3.1 Global gas and Brent prices in US Dollars per million British thermal units: Jan 2009 – Dec 2023. (Heather, 2024) 55

3.2 Share of Russian pipeline gas in total EU natural gas imports. Source: (European Commission, 2023) 56

3.3 Comparison of the demand scenarios from 2023 to 2050. The Model scenario demand decreases linearly. Forecasts from the DE government (DEG) (BMWK, 2023a) and Net Zero Emission (NZE) (David et al., 2023) are limited to 2030. The dashed lines correspond to the linear projections from 2030 to 2050. 66

3.4 Map of Germany, divided in NUTS2 regions, with the modelled gas transmission network. Arcs with an overcapacity relaxation are emphasized in red. Cropped-out nodes are import nodes. . 70

3.5 FSRU imports over time by port and scenario. 71

3.6 Pipeline import from connected countries over time for each scenario. Countries with a dotted supply line are that of the LNG category in Table 3.4. For clarity, the lines are rendered only where strictly positive. 72

3.7 Sensitivity of the optimal investment policy to four parameters. All horizontal axes are in percentages relative to the default values of the *Greenfield LIN* scenario (vertical dashed lines). . 75

4.1 An example of a phylogeny of five taxa represented by a UBT with 5 leaves and the associated PLM $\boldsymbol{\tau}$. 82

4.2 The three iterations of an agglomeration method required to generate a phylogeny of six taxa: iteration 1) merges singletons $\{1\}$ and $\{2\}$ and introduces internal node A, iteration 2) merges cluster $\{1, 2\}$ with the singleton $\{3\}$ and introduces node B; iteration 3) merges singletons $\{5\}$ and $\{6\}$ and introduces node C, resulting in the final 6-taxon phylogeny. 86

LIST OF TABLES

2.1 Values of the hyperparameters of the PPO algorithm. 34

2.2 Hardware configurations of the different experiments and approximative time (expressed in minutes) to train one agent. Each GPU was dedicated to train one agent at a time. 36

2.3 Parameters of the learning environments. The default values are emphasized in bold. 36

2.4 Time series used to generate the test forecasts for 8 different trends. 38

2.5 Summary of the near-optimality experiments (Section 2.5.2). Each row summarizes the ten agents that were trained with the environment parameters in the four leftmost columns. The four rightmost columns give the average, best, or worst of the ten agents' median gaps (in %) with respect to either the DP method or the Simple heuristic (see the headers). 39

2.6 Summary of the Updating Forecasts experiments (Section 2.5.3). Each row summarizes the ten agents that were trained with the environment parameters in the two leftmost columns. The four rightmost columns give the average, best, or worst of the ten agents' median gaps (in %) with respect to either the DP method or the Simple heuristic (see the headers). 41

2.7 Summary of the Lost Sales experiments (Section 2.5.4). Each row summarizes the ten agents that were trained with the environment parameters in the three leftmost columns. The four rightmost following columns give the average, best, or worst of the ten agents' median gaps (in %) with respect to either the DP method or the Simple heuristic (see the headers). The three rightmost columns give the count of agents with a median gap within selected intervals. 42

3.1 Description of sets and subsets. 63

3.2 Description of parameters. 64
3.3 Description of variables 65
3.4 Classification of countries exporting to Germany. The LNG category is that of countries that sell regasified LNG; Producers that sell domestically produced natural gas; and Transit that only act as transit countries. 68
3.5 Summary of scenario-specific changes to the model and parameters. 69
3.6 Comparison between the optimal solutions of the scenarios. Capex includes pipeline construction costs. Pipeline gas import costs are divided between the source categories of Table 3.4 in addition to FSRU imports. All costs are nominal (undiscounted) over the 2023-2050 horizon. 73

4.1 Comparison of the tested methods for each data set. The *Best* columns indicate the percentage of instances in the respective data sets where the best tree across all methods was identified by the method. The *Gap* columns report the average optimality gap (in percents) over the data set. Results are presented separately for standalone agglomeration algorithms and after applying Subtree Pruning and Regrafting (SPR) optimization. Bold values highlight the best results in each column. 93
4.2 Average run time of each method in seconds per instance size (number of taxa). 94

Chapter

ONE

INTRODUCTION

1.1 General introduction and context

Operational Research (OR) has evolved far beyond its origins in military logistics to become a broad and dynamic field that addresses complex decision-making problems across a wide range of domains. This evolution is marked not only by methodological advances—from mathematical programming to machine learning—but also by an expanding and increasingly urgent scope of applications. OR today plays a central role in tackling contemporary global challenges, offering tools to support decisions in areas as diverse as supply chain management, public health, energy planning, and beyond.

This thesis is structured around three studies, each inspired by a major global event that occurred during my years as a PhD student: the rise of artificial intelligence, the COVID-19 pandemic, and the geopolitical energy crisis following Russia's invasion of Ukraine. Each study addresses a distinct problem and employs a different set of OR tools, thereby showcasing the field's versatility and relevance. Collectively, these chapters exemplify the modern OR paradigm—one that blends classical rigor with innovation to produce solutions grounded in both theory and practice. In doing so, they also highlight OR's capacity to adapt and contribute meaningfully to fast-evolving, real-world problems——including a perhaps unexpected application to computational biology.

The following chapters explore each study in turn, beginning with a problem rooted in the transformative impact of artificial intelligence. In 2016, DeepMind achieved a milestone by defeating the world champion of Go, Lee Sedol——an accomplishment many experts had predicted was still decades away. Their approach, AlphaGo (later generalized to AlphaZero), combined Deep Reinforcement Learning (DRL) with classical tree search algorithms (Silver et al., 2016, 2017). At the same time, a now-famous paper introduced the revolutionary Transformer neural network architecture (Vaswani et al., 2017), which catalyzed the rapid development of Large Language Models (LLMs). At the time of the writing of this thesis, the plethora of available AI models have capabilities that few were able to even fathom when I started this journey. As a young researcher looking for a topic, major breakthroughs like this come as a blessing as they open up unexplored research questions. Reinforcement Learning (RL) has been used as an optimization framework in operational research well before the aforementioned events. However, its limited ability to scale to problems of realistic size never allowed the methods to find many applications outside of research. The successful integration of neural networks into the framework allowed the discovery of algorithms capable of solving highly complex and dynamic tasks, such as playing Atari video games or robot locomotion (Mnih et al., 2015; Haarnoja et al., 2018). These advances raised the question: do these results transfer to operations management problems? The first chapter of this thesis attempts to partially answer that question. There, we focus on the single-level stochastic inventory problem with non-stationary demand. This problem has been studied for decades and has well-known properties and a simple and interpretable optimal decision policy (Scarf, 1960). The goal of our research was to evaluate the capacity of DRL algorithms to autonomously discover the results obtained by past researchers. Notably, could a DRL algorithm discover the characteristic (s, S) optimal policy of the problem. We then exploited our findings to design a learning environment that allows the algorithm to reliably converge to near-optimal policies for more complex problems.

The second major event was the COVID-19 pandemic and its associated lockdowns during 2020-2021. As the SARS-CoV-2 virus evolved into increasingly contagious variants, tracking its evolution became essential for identifying emerging health threats (Nextstrain, 2025). The challenge of inferring the phylogeny of a set of individuals (taxa) is not new to the pandemic and has been actively researched for decades (Hadfield et al., 2018). It turns out that deriving the tree that most likely retraces the evolution of a set of taxa is an optimization problem that belongs to the broad class of network design problems. This is typically addressed by minimizing a measure of the tree's length. In this thesis, we focus on the Balanced Minimum Evolution Problem (BMEP). Traditionally, bioinformatics approaches have relied on efficient yet straightforward heuristic methods to tackle this problem, whereas operational research (OR) has employed exact solution methods—often via mathematical programming formulations or implicit enumeration algorithms. Although using mathematical programming to solve this phylogeny inference problem is an active area of research, its application is limited by the numerical properties of the BMEP. In Chapter 4, we demonstrate how insights from exact solution approaches can be harnessed to enhance widely used heuristic methods. Specifically, we propose the Linear Programming Neighbor Joining (LPNJ) matheuristic, which integrates mathematical programming into the well-known Neighbor Joining (NJ) heuristic algorithm. We further embed both NJ and LPNJ into a beam search framework to generate a larger pool of candidate solutions. Numerical experiments show that our proposed algorithms yield improved phylogenies compared to current state-of-the-art approaches.

The third significant event was Russia's invasion of Ukraine, which led to Western economic sanctions. In response, Moscow curtailed gas flows to Europe, leveraging energy exports as a geopolitical tool (European Commission, 2023). The subsequent sabotage of the Nord Stream pipelines further constrained Europe's import capacity. Germany was particularly affected, as it is one of Europe's largest consumers of natural gas for both heating and electricity generation. In response, the German government swiftly implemented a plan to replace Russian gas with Liquefied Natural Gas (LNG) by installing regasification infrastructures. This plan involves the short-term deployment of Floating Storage and Regasification Units (FSRUs) and the long-term construction of onshore facilities. This crisis coincides with the growing urgency around climate change, creating tension between immediate energy security needs and long-term environmental commitments. In the third chapter of this thesis, we assess whether Germany's decision to develop LNG regasification infrastructure aligns with its 2050 net-zero carbon emissions target. We analyze multiple scenarios for the evolution of Germany's natural gas demand and determine the optimal infrastructure investment strategy using a facility location and network flow model. Our objective is to evaluate whether Germany's energy diversification plan is consistent with the European Green Deal's decarbonization objectives.

Before delving into the core chapters of the thesis, the following sections present the background necessary to support the more technical content. Specifically, Section 1.2 provides the fundamentals of Deep Reinforcement Learning relevant to Chapter 2, Section 1.3 describes in deeper details the pre-processing treatment that was necessary to obtain a feasible gas-flow model in Chapter 3, while Section 1.4 offers a brief introduction to the problem of phylogeny inference.

1.2 Background on deep reinforcement learning

In this section, we begin by reviewing classical tabular Reinforcement Learning (RL) methods for solving Markov Decision Processes (MDPs). Here, we purposefully stick to the essential concepts and redirect the interested reader to the classical book of Sutton and Barto (2018) for an indepth introduction. We then discuss how these methods extend to Deep Reinforcement Learning. Finally, we describe in detail the Proximal Policy Optimization (PPO) algorithm, which we employ in Chapter 2.

1.2.1 Tabular reinforcement learning

Reinforcement Learning is an algorithmic paradigm where an *agent* learns to maximize its accumulation of rewards that are obtained during sequential interaction with an *environment*. It uses the well-known discrete-time MDP framework to model these interactions by defining the state of the environment, the actions that the agent can take, and how the environment will react to the actions by sending rewards and transitioning to new states (see Figure 1.1) (Sutton and Barto, 2018). Specifically, at each time step t, the environment is in state x_t, and the agent selects an action a_t according to its *policy* π. The policy may be deterministic, in which case $a_t = \pi(x_t)$, or stochastic, in which case $\pi(x_t)$ defines a parameterized probability distribution over the action-space—e.g. the mean and standard deviation of a Gaussian, or the logits of a categorical distribution—from which a_t is sampled. The environment then (possibly stochastically) transitions to the next state x_{t+1} and returns a *reward* r_t depending on the state-action pair (x_t, a_t). This cycle repeats until termination (if the MDP has terminal states). The aim of RL is to let the agent learn a policy π that maximizes the expected return

$$J(\pi) \;=\; \mathbb{E}\Big[\sum_{t=1}^{T} \gamma^{t-1}\, r_t\Big],$$

that is, the discounted sum of rewards where $\gamma \in (0, 1]$. The central idea is that because the agent interacts sequentially with the environment, it must find a policy that yields good immediate rewards while keeping the environment in a state likely to generate high rewards in future interactions. Applying RL to a specific problem requires two steps: modeling the problem as an environment, and executing a learning algorithm to optimize the agent's policy.

Let us illustrate this by describing the basic reasoning necessary to formulate an MDP to model a simple inventory control problem. For the state to be Markovian, it must contain all the information necessary to model the environment. That is, the state must be a sufficient input for the policy to take a fully informed decision, without the need to remember previous states and actions. The most obvious state variables that the vector x_t must contain should encode the inventory levels of the different items in consideration. Additionally, to preserve the Markovian property in the presence of lead times, x_t must also encode past orders that are currently awaiting delivery. The typical decision (the action) taken by the agent at each time step is the order quantity for each item, i.e. a_t can be encode as a vector of positive real values representing individual order quantities. We will see in Chapter 2 that depending on the characteristics of the problem, the choice of how to represent an action may be of crucial importance for the ability of the algorithm to correctly learn a good policy. The rewards are the periodic elements that would otherwise constitute the objective function of the problem in a traditional optimization framework. Here, the negative of the costs of inventory, stockouts, and ordering

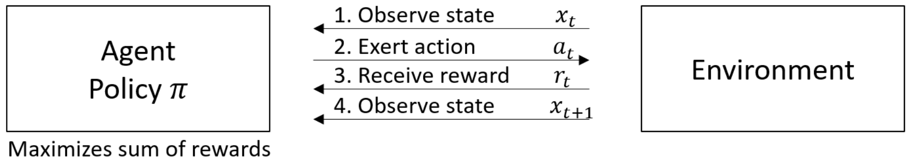


Figure 1.1: One transition of an MDP at time step t.

are the usual components of the reward. Finally, the transition to a new state is computed by taking the previous levels of inventory, to which we subtract the consumption due to, say, a stochastic exogenous demand, and add deliveries according to past orders.

When the environment's transition probabilities are known, we can apply Dynamic Programming (DP) methods—such as policy-iteration or value-iteration. However, as the state space grows, computation time and memory requirements scale exponentially, rendering DP intractable for high-dimensional problems. Moreover, fully modeling the environment dynamics can be challenging in and of itself. Reinforcement Learning addresses these issues by letting the agent learn from experience: it collects data over *episodes* of interaction and uses that data to update its estimate of the *action-value* function. The main advantage of this approach is that the agent-environment interactions can be programmed in the form of simulations, which does not require to explicitly compute the transition probabilities.

Formally, under policy π, the action-value function is defined by the recursive identity

$$Q^\pi(x_t, a_t) \;=\; \mathbb{E}\big[r_t + \gamma\, Q^\pi(x_{t+1}, a_{t+1}) \;\big|\; x_t, a_t\big], \tag{1.1}$$

with the boundary condition $Q^\pi(x_T, a) = 0$, for all a, at terminal states. $Q(x, a)$ can be interpreted as the expected value of taking action a when in state x. As Equation (1.1) emphasizes, the expected value is recursive in time: the action-value at time t is equal to the expected immediate reward plus the expected action-value at time $t+1$. The expectation accounts for stochasticity in transitions, policy, and rewards. If Q is known for all state–action pairs, the optimal policy is

$$\pi(x_t) \;=\; \operatorname*{argmax}_a Q(x_t, a). \tag{1.2}$$

The *learning* in Reinforcement Learning stems from the the central task of estimating this Q-function, which the agent then uses to interact with the environment via the simple policy given in (1.2). Multiple algorithms exist for this; the class of *temporal difference (TD) learning* methods exploits the recursive structure of Equation (1.1) to update Q as follows. Given an episode $(x_1, a_1, r_1, x_2, a_2, r_2, \ldots, x_T)$ collected from agent–environment interactions, the *Q-learning* algorithm applies the following update for each *transition* represented by the quadruplet (x_t, a_t, r_t, x_{t+1}):

$$Q(x_t, a_t) \leftarrow Q(x_t, a_t) + \alpha\Big(r_t + \gamma \max_{a'} Q(x_{t+1}, a') - Q(x_t, a_t)\Big), \tag{1.3}$$

where $\alpha \in (0, 1]$ is the *learning rate*. Intuitively, this update forces the Q-function to be "temporally consistent" with itself: by Identity (1.1), the interior of the parentheses in (1.3) should be 0 on average. In tabular RL, the Q-function is stored as a table indexed by each (x, a) pair. While Q-learning theoreticaly converges to the optimal Q^*, under certain conditions, visiting the entire state–action space can be prohibitively slow in large problems.

A case of particular interest to us in the context of Inventory Control is that of continuous state spaces, where the number of possible states becomes infinite. In such settings, tabular RL is unusable and one must resort to *approximate RL*, wherein the value function is represented by a parametrized function approximator of the state, such as a linear function. The underlying assumption is that two distinct states with numerically similar state variables typically have similar action-values. This observation remains valuable in Inventory Control applications, despite inventory levels often being integer rather than real-valued. Consequently, a well-tuned function approximator with sufficiently high representation power might be able to interpolate the value of a previously unseen state. In contrast, tabular value representations need to learn the value of each distinct state-action pair independently, a process which necessitates a large number of visitations per key in the table. A similar argument can be made about continuous action-spaces, in which case it becomes necessary to also approximate the policy function.

One special class of function approximators is that of Deep Neural Networks (DNNs). Their successful use in approximating the action-value function led to a series of breakthroughs, enabling RL to tackle far more complex environments than were previously possible. The resulting field of *Deep* RL is the subject of the next sections.

1.2.2 Deep Q-Learning

DNNs are universal function approximators parameterized by thousands to billions of parameters, able to represent any continuous function (Hornik et al., 1989). Deep learning is the process of optimizing the values of these parameters by minimizing a loss function so that the network approximates a desired function. In the remainder of this chapter, because the parameters are usually not considered as inputs of the function they approximate, we will separate their associated symbol from the other arguments with a semi-colon to emphasize the distinction. A DNN is typically trained (or fitted, or tuned) by taking the gradient of a loss function with respect to its weights parameters, and applying a gradient descent-like algorithm. In modern deep learning frameworks, the gradient computation is automatically carried out by means of autograd algorithms, such as those used in the well-known PyTorch and Tensorflow libraries (Abadi et al., 2015; Paszke, 2019). In classical supervised deep learning, given an input to the DNN, the loss function gives a measure of the error between its output and the known true label. The training algorithm iteratively updates the neural network weights so that it fits the function underlying the distribution that generated the inputs-labels inh the dataset, in the hope that it may then predict the correct label given a new input (Goodfellow et al., 2016).

However, this approach is not possible in RL as the target output is not known, this would require the RL problem to be already solved. Fortunately, the Q-learning algorithm already provides a straightforward solution by adapting Equation (1.3) to the case where Q is a neural network. In that case, the gradient of the squared temporal difference error is used to update the parameters. That is, given a transition observed during agent-environment interactions, the training procedure computes the gradient with respect to θ (the weights of the DNN) of the loss

$$L(\theta) = \left(r_t + \max_a Q(x_t, a; \theta') - Q(x_t, a_t; \theta)\right)^2 \tag{1.4}$$

deep Q-Learning alternates between collecting transitions data of several episodes, and updating the neural network by taking the average gradients of experience batches for several epochs. Exploration of the action-space is ensured by letting the agent randomly deviate from the approximate policy (1.2). Once the training phase is over, the agent strictly follows (1.2). In practice, several additional considerations need to be made in order for this approach to be effective. For example, the Loss function (1.4) uses a *target* neural network θ', which is a moving average of θ, necessary to mitigate an oscillation effect that occurs when the Q-network defines its own learning target. As in classical deep learning, it is also necessary to take the average gradient of (1.4) for a batch of transitions sampled from a *replay buffer*—a database of past agent-environment interactions. This approach was proposed by Mnih et al. (2015) who managed to teach DNNs how to play several atari games at a human-like level. This was the first major breakthrough which propelled DRL to one of the most active machine learning research domains at the time. Nevertheless, deep Q-Learning is characterized by a low stability in learning and is inherently unable to tackle continuous action-spaces.

1.2.3 Proximal Policy Optimization

Several algorithms have been proposed to address the limitations of Deep Q-Learning. Among these, Proximal Policy Optimization (PPO), introduced by Schulman et al. (2017), stands out for its balance between performance and simplicity. PPO enhances scalability, sample efficiency (the ability to reuse experience multiple times during learning), robustness to hyperparameter choices, and applicability to a broader range of environments, including those with continuous action spaces—which are of particular interest to us. By employing a clipped surrogate objective, PPO stabilizes policy updates without resorting to complex second-order methods like Trust Region Policy Optimization (Schulman et al., 2015). Here, we provide an introduction to the underlying principles of the algorithm.

PPO belongs to the class of *actor-critic* algorithms, which are composed of two neural networks. The first, the actor (with parameters ϕ), learns the policy π of the agent, while the second, the critic (with parameters θ), approximates the value function. Unlike in Deep Q-Learning, the value function learned by the critic is not used to select actions directly; instead, it provides a baseline that guides the actor's updates, hence the name *critic*. In PPO, the critic approximates the state-value function

$$V(x_t;\theta) = \mathbb{E}_{a\sim\pi(x_t;\phi)} Q(x_t, a), \tag{1.5}$$

that is, the expected return from state x_t under the policy π. The actor approximates a stochastic policy conditioned on the state by returning the parameters of a probability distribution. It is common to denote the policy as $\pi(a_t|x_t;\phi)$ to refer to the probability, or probability density, of taking action a_t when in state x_t. For instance, if the actor network $\pi(x_t;\phi)$ returns a mean vector μ and a standard deviation vector σ to represent a k-dimensional multivariate Gaussian distribution with diagonal covariance, then the associated log-probability distribution function is

$$\ln \pi(a|x_t;\phi) = -\frac{1}{2}\sum_{i=1}^{k} \frac{(a_i-\mu_i)^2}{\sigma_i^2} \; - \; \sum_{i=1}^{k} \ln(\sigma_i) \; - \; \frac{k}{2}\ln(2\pi). \tag{1.6}$$

The training objective adjusts action probabilities based on their observed performances: if the observed return from action a_t in state x_t exceeds the expected value of x_t, the selection probability of a_t is increased; if it falls short, the probability is decreased. The actor is trained according to this principle by computing the *advantage* defined by

$$\hat{A}_t = \hat{Q}(x_t, a_t) - V(x_t;\theta), \tag{1.7}$$

which is positive when a_t outperforms the state-value baseline, and negative otherwise. As the true Q is unknown, it is replaced by a sample estimate obtained from observed interaction between the agent and the environment. The simplest estimate use the temporal-difference error arising from Equations (1.1) and (1.5):

$$\hat{A}_t = \delta_t = r_t + \gamma V(x_{t+1};\theta) - V(x_t;\theta). \tag{1.8}$$

This estimate of A_t is severely biased by the approximation error of V. Schulman et al. (2016) proposed a Generalized Advantage Estimation (GAE) to alleviate this issue by also incorporating the rewards obtained in steps $t+1$ to $T-1$ with exponentially decreasing factors:

$$\hat{A}_t = \sum_{k=t}^{T-1} (\gamma\lambda)^{k-t} \delta_k, \tag{1.9}$$

where δ_k is defined by Equation (1.8) and $\lambda \in (0, 1]$ is a parameter, usually between 0.95 and 0.98.

PPO optimizes the actor parameters ϕ by minimizing the loss function

$$L(\phi) = -\hat{\mathbb{E}}_t \left[\min\left(\rho_t(\phi)\, \hat{A}_t,\ \text{clip}\big(\rho_t(\phi), 1-\epsilon, 1+\epsilon\big)\, \hat{A}_t \right) \right], \tag{1.10}$$

where

$$\rho_t(\phi) \;=\; \frac{\pi(a_t \,|\, x_t; \phi)}{\pi(a_t \,|\, x_t; \phi_{\text{old}})},$$

and $\hat{\mathbb{E}}_t[\cdot]$ denotes the empirical average over a batch of transitions. The clip function constrains the ratio $\rho(\phi)$ between $1-\epsilon$ and $1+\epsilon$. ϕ_{old} denotes the neural network weights that were used by the policy during the agent-environment interaction phase for data collection. This loss function ensures that the action probabilities change by at most ϵ before new data is collected for further training, hence the name *proximal*. The clipped objective allows both the actor and critic networks to be trained for multiple epochs of stochastic gradient descent on the same set of collected data before new interactions are played with the updated policy. To encourage exploration of the action-space by ensuring a sufficiently random policy, PPO also subtracts an entropy bonus regularization term

$$\beta\pi(a_t|x_t;\phi)\ln(\pi(a_t|x_t;\phi))$$

from the loss (1.10), where β is a positive parameter. This also ensures that the policy does not collapse to a deterministic distribution. As in deep Q-Learning, the network is trained by taking the average gradients over batches of data for several epochs.

The critic parameters θ are learned by minimizing the mean squared temporal-difference error of the value-function with a loss analogous to (1.4):

$$L(\theta) = \hat{\mathbb{E}}_t \Big[\big(r_t + \gamma\, V(x_{t+1};\theta) - V(x_t;\theta) \big)^2 \Big].$$

Here, we only covered the core principles of PPO; some further implementation details required to improve stability and efficiency during training are discussed later in Section 2.5.1.

Since its introduction, PPO has become one of the most widely adopted algorithms in deep reinforcement learning, demonstrating strong performance across diverse tasks. Notable landmark applications of PPO include achieving human-level performance when playing Atari games, virtual robotic locomotion tasks, and dexterous in-hand robotic manipulation and solving of a Rubiks cube (Schulman et al., 2017; Akkaya et al., 2019). More recently, PPO has become the de facto algorithm of choice in one of the post-training phases used to align the outputs of large language model-based chatbots such as ChatGPT with human preferences (OpenAI, 2022). These qualities make PPO a good default algorithm of choice to use in Chapter 2, where we will employ it to learn inventory control policies.

1.3 Processing of imperfect model parameter data

In Chapter 3, we employ a relatively straightforward network-flow model to obtain insights into energy-infrastructure policies. In this instance, a significant share of the research effort consisted in the processing of the imperfect available data. In Section 3.4.6, we describe how we used the public IGGIELGN database (Diettrich et al., 2021; Pluta et al., 2022) to construct the graph object that serves as the basis of our network-flow model. We omitted these details in the Energy Policy journal due to its less technical focus. Here, we fill that gap by describing in more detail the issues we faced during development, the methods we used to address them, and the scientific principles that guided our methodological choices. The objective is to provide the reader with a deeper description of this important phase of the research. In Section 1.3.1, we focus on the methods we employed to fix the topology of the graph we obtained from the available data, while in Section 1.3.2, we describe our approach to deal with numerical adjustments to the parameters of the model.

1.3.1 Fixing graph topologies for pipeline networks

The IGGIELGN dataset consists of several collections, from which we use a general collection of nodes, a collection of "consumer" nodes, and a collection of pipeline segments. Nodes are characterized by their geographical coordinates, country and NUTS identifiers, and—if they also appear in the consumer dataset—by a production capacity in MW. Pipelines are characterized by their endpoints' coordinates, capacity (in million cubic meters per day), length, and whether they are bidirectional.

The goal of the optimization model in Chapter 3 is to determine the gas flow from a set of source nodes to the remaining nodes of the graph—hereafter referred to as demand nodes. By mapping each database entry to its geographical coordinates, we identified which nodes correspond to pipeline endpoints and thus established the graph's vertices and arcs. Although this construction is theoretically sufficient, the model proved infeasible, a problem we traced to

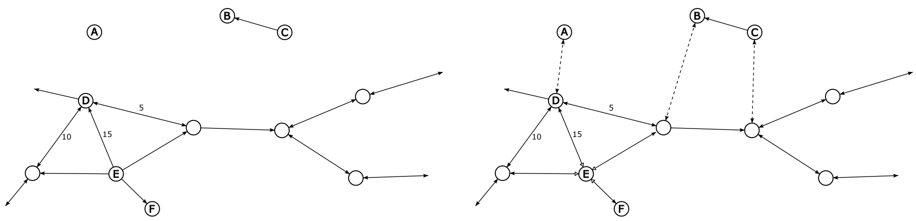


Figure 1.2: Left: toy original graph with two disconnected components ({A} and {B,C}) and two unreachable nodes, E and F, in the main component. Right: The same graph repaired with added virtual pipelines (dashed arcs) and virtual bi-directionalities (hollow arrowheads).

the presence of disconnected components in the graph whose demand cannot be satisfied (see nodes A, B, and C in Figure 1.2, left). Resolving this issue is not trivial: it requires careful adjustments to the data to maintain the scientific validity of the analysis.

Identifying disconnected components is straightforward. One approach is to iteratively execute undirected graph traversals from each node not yet visited during a previous traversal. However, since computing time is not critical in our case, we opted instead for a standard implementation of the Floyd–Warshall algorithm (Floyd, 1962) on an undirected version of the graph to identify groups of nodes connected by at least one path.

Fortunately, for this specific pipeline network, the structure of the disconnected components permitted a straightforward resolution. Each component contained only one or two nodes, leaving little ambiguity about the missing pipelines' endpoints. We can recover a feasible graph in two ways: reconnect the components or remove them entirely. As a general scientific principle, one should make as few assumptions as possible (this is the *lex parsimoniae* principle, which reappears in Section 1.4 in a different context). Here, reconnecting is preferable: it is more plausible that pipelines are missing from the database than that fictitious nodes were added. Moreover, removing nodes could substantially shift local demand and alter nearby flows. Regarding the characteristics of the added pipelines, we again minimize assumptions: we connect each disconnected node to its nearest neighbor, make the pipelines bidirectional, and assign them infinite capacity. Although "infinite" may seem unrealistic, it simply assumes the pipeline's capacity suffices to meet demand——its effective capacity is still bounded by the known capacities elsewhere in the graph. For instance, in Figure 1.2 (right), the virtual pipeline DA is effectively limited to a flow of 30 by node D's total inflow capacity.

Finally, as illustrated by nodes E and F in Figure 1.2, some nodes may be topologically connected yet remain unsupplied due to pipeline orientations. Node E is unreachable because it has no incoming arcs, and node F is unreachable because all its neighbors are themselves unreachable. Here, we consider reachability specifically from source nodes (otherwise, F would be deemed reachable from E). To identify these unreachable nodes, we again employ a shortest-path algorithm to detect nodes with no path from any source. We then convert all outgoing arcs on those nodes to bidirectional to restore reachability.

1.3.2 Resolving capacity constraints with minimal assumptions

We now consider feasibility in terms of numerical data. In our case, the pipelines' capacities proved insufficient to deliver gas to all demand nodes. Two likely causes arise: first, the capacities in the dataset may be incorrect; second, our estimated nodal demand may exceed what the surrounding pipelines can carry. Again, following the principle of lex parsimoniae, we seek to make as few changes as possible to restore feasibility. Indeed, we originally estimated the nodal demand by assuming it is proportional to each node's total incoming pipeline capacity, so imposing further assumptions about demand distribution would be undesirable.

Instead, we opted for an approach that permits to violate the pipeline capacity constraints of the model. Let X_e be the total flow in a pipeline e and k_e its capacity. We replace the original capacity constraints ((3.4) in the complete model in Chapter 3, simplified here)

$$X_e \leq k_e \quad \forall e \in E \tag{1.11}$$

with the relaxed form

$$X_e \leq k_e + v_e \quad \forall e \in E \tag{1.12}$$

where E is the set of pipelines and v_e is a non-negative artificial variable measuring any capacity violation. Following the principle of lex parsimoniae would entail minimizing

$$v = \sum_{e \in E} v_e.$$

This can be achieved by a classical multi-objective lexicographic approach where we first minimize v, then minimize the original objective of the model by ensuring v remains minimal.

However, the lexicographic scheme as described is ultimately invalid for our purposes. By insisting on the absolutely minimal violation v in the first stage—without regard for cost—the model may select a solution that, although optimal in terms of v, is arbitrarily expensive. Once v is fixed at its minimum, the second stage cannot revisit trade-offs between violation and cost, so it is powerless to improve the original objective beyond that initial, cost-agnostic choice. In other words, minimizing v in isolation can "lock in" a suboptimal cost configuration that no longer admits any cost-driven adjustments. Fortunately, allowing an increase of 20% for the value of v compared to its minimal value is sufficient to allow the model to find cost-effective solutions while keeping capacity adjustments within an acceptable range.

1.4 Background on phylogeny inference

Phylogeny inference is the problem of reconstructing the evolutionary history of a set of taxa given their associated molecular sequence data (e.g. RNA sequences). This methodology underpins applications including, but not limited to, cancer phylogenetics for mapping clonal dynamics in tumors (Beerenwinkel et al., 2015), epidemiological tracing of pathogen spread (Hadfield et al., 2018), with a notable application to the tracing of the SARS-Cov-2 variant (Nextstrain, 2025), and biodiversity assessment through phylogenetic diversity metrics (Duellman et al., 2016). The objective is to reconstruct the phylogeny (i.e., the evolutionary tree) that best fits the available molecular sequence evidence. Several criterion were proposed to assess the quality of a tree. One successful choice is that of minimizing the Balanced Minimum Evolution (BME) criterion, which was shown to be statistically consistent (Desper and Gascuel, 2004) and has proven to be empirically well-performing (Lefort et al., 2015b). In Chapter 4, we propose several heuristic solution methods for the Balanced Minimum Evolution Problem (BMEP), treating it as an optimization problem. Here, we lay out the fundamentals of Phylogenetics and the motivations behind studying the problem under an OR lens.

1.4.1 Phylogenies

A phylogeny is a representation of the evolutionary relationship of a set of taxa. This relationship is, in most cases, represented as a tree whose leaves are each associated to a taxon. The biological meaning of a phylogeny depends on the application. For example, in the case of systematics, the taxa are the species and the internal vertices of the tree correspond to speciation events; whereas in the study of tumor evolution, the taxa are cells sampled from a tumoral mass and the internal vertices correspond to genetic mutations. Finally, the edges of a phylogeny have weights whose values quantify the evolutionary change between two vertices in the tree. The sum of the weights along the path that connects two leaves of the tree thus gives a measure of the dissimilarity between the two corresponding taxa. A simple illustrative example is given in Figure 1.3

In general, the tree structure of the phylogeny is an Unrooted Binary Tree (UBT), that is, a tree where all internal vertices have a degree of three (see Figure 1.4). Rooted and non-binary trees (such as those arising when polytomies are considered) can always be converted to an equivalent UBT by adding dummy edges.

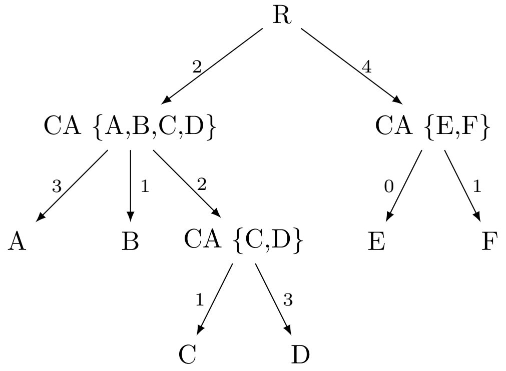


Figure 1.3: Tree representation of a phylogeny rooted in R for a set of six taxa. This tree features a polytomy at the Common Ancestor (CA) {C,D}. The branches of the tree are weighted, including one null branch indicating no dissimilarity between E and its parent.

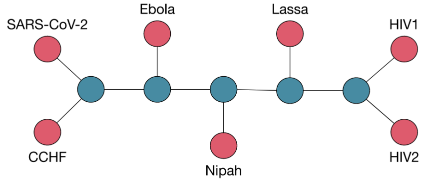


Figure 1.4: Unweighted phylogeny of a set of seven viruses. This tree is the optimal UBT obtained from solving the BMEP. See Catanzaro et al. (2022) for details on the computation of this solution. Reproduced from Catanzaro et al. (2022) with permission.

1.4.2 Phylogeny inference is an optimization problem

The construction of a phylogeny is achieved by solving an optimization problem for both the topology of the tree, and the weights of the edges. Several alternative quality criterion exist, each based on a particular set of assumptions. Formally, finding the best possible phylogeny topology T with edge weights w for a set Γ of n taxa according to a criterion Λ is equivalent to solving the generic problem

$$\begin{aligned} \text{optimize} \quad & \Lambda_\Gamma(T, w) \\ s.t. \quad & T \in \mathcal{U}_n \\ & (T, w) \in \Omega_\Gamma \end{aligned} \tag{1.13}$$

where $\mathcal{U}_n$ is the set of $(2n-5)!! = 3 \times 5 \times \cdots \times (2n-5)$ possible UBTs with n leaves, and Ω_Γ is a set of constraints to ensure the tree satisfies a certain set

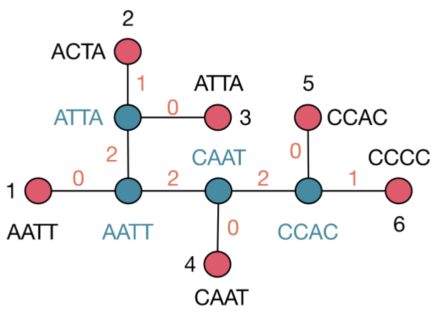


Figure 1.5: Example of a most parsimonious phylogeny represented an UBT. The leaves (red) correspond to the six taxa and the internal vertices (blue) to the inferred common ancestors. The edge weights (orange) are the Hamming distances between the vertices. Reproduced from Catanzaro et al. (2022) with permission.

of assumptions.

One common assumption in the general scientific methodology is that of *maximum parsimony.* This principle, also known as Occam's razor, states that among all theories that could explain an observation, one should favor that which makes the least number of assumptions (Popper, 2005). The underlying idea is that the simplest explanation is also the most likely to be true, as long as there is no evidence to contradict it. In the context of molecular evolution, this is accomplished by minimizing the number of mutations necessary to explain the lineage that separates two taxa. Solving the maximum parsimony problem by means of Problem (1.13) can be achieved by assigning weights to the edges on the path that connects two taxa such that the sum of those weights is equal to the number of point-mutations that separate the two associated molecular sequences (Fitch, 1971). For the phylogeny to make sense under the maximum parsimony criterion, each internal vertex of the tree must be associated to a specific intermediate sequence of taxa and the weight of an edge must be equal to the Hamming distance between the sequences associated to the endpoints of the edge. This necessity is enforced by a set of constraints in Ω_Γ. Minimizing the sum of all edge-weights on the tree is an instance of being *parsimonious*, in that it assumes the least number of mutations occurred. As an example, Figure 1.5 shows the most parsimonious phylogeny for a set of six taxa whose molecular sequences are composed of four nucleotides. Notice that the sum of the edges on a path connecting two taxa may be greater than the Hamming distance between their respective sequences (e.g. taxa 1 and 6).

Unfortunately, the parsimony criterion was proven to be *statistically inconsistent* (Felsenstein, 1978). A statistically consistent method is one whose probability of inferring the correct evolutionary tree tends to one as the amount

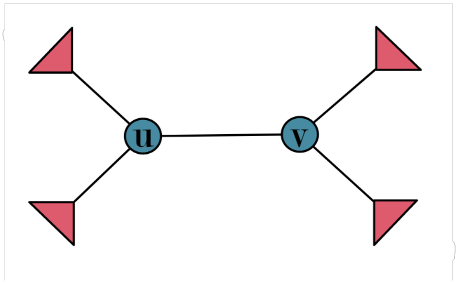


Figure 1.6: An edge connecting two internal vertices u and v and the four incident subtrees represented as red triangles. Adapted from Catanzaro et al. (2022) with permission.

of available sequence data grows. Because evolution is not an observable process, it is of crucial importance to work with methods capable of recovering the correct phylogeny based solely on the current state of the evolutionary process.

Instead of directly working with sequence data, the *Miniminum Evolution* (ME) criterion proposed by Kidd and Sgaramella-Zonta (1971) uses an n by n dissimilarity matrix $\mathbf{D}$ estimated via probabilistic models of sequence evolution (see Felsenstein (2004a)), whose generic entry d_{ij} encodes an estimate of the evolutionary distance that separates taxa i and j. The Minimum Evolution Problem is thus formulated as Problem (1.13) where one minimizes

$$\Lambda_\Gamma(T, w) = \sum_{e \in E(T)} w_e$$

such that

$$\sum_{e \in P^T_{ij}} w_e = d_{ij} \quad i, j \in \Gamma \tag{1.14}$$

where $E(T)$ is the set of edges of T, and P^T_{ij} is the set of edges on the path in T that connects i to j. Rzhetsky and Nei (1993a) proved that if $\mathbf{D}$ encodes an unbiased estimate of the true evolutionary distance, then the expected length of the true phylogeny—the sum of its edge weights—is shorter than any phylogeny where (1.14) is satisfied; thereby proving statistical consistency of the ME criterion. However, because $\mathbf{D}$ is only an estimation, (1.14) often does not admit a solution. Thus, an Ordinary Least Squares (OLS) estimate of the weights is used instead (Cavalli-Sforza and Edwards, 1967), fortunately without losing the statistical consistency property. However, solutions of the ME problem with OLS have the undesirable property of allowing negative edge weights, devoid of any biological meaning, and likely the consequence of wrong assumptions in the construction of $\mathbf{D}$.

A solution to this issue was eventually found by Pauplin (2000). Consider two adjacent internal vertices u and v and the four disjoint subtrees rooted in vertices incident to u and v (see Figure 1.6). The author observed that, under the OLS model, the subtrees are implicitly not equally weighted in the calculation of the tree length due to the computation of w being correlated to that of topology of the phylogeny. This fact not being supported by any biological evidence, he proposed a *balanced* version of the ME criterion where the distance between two subtrees rooted in i and j is merely the average of the distances between i and the two subtrees rooted in j's children. This leads to a scheme where each subtree is equally weighted. The calculation of w is no longer achieved by explicitly treating the weights as variables, but instead becomes a simple recursive average function of $\boldsymbol{D}$ and the topology T. The length of a tree under the *Balanced Minimum Evolution* (BME) criterion is given by

$$\ell(T) = \sum_{i,j \in \Gamma} \frac{d_{ij}}{2^{\tau_{ij}}}, \tag{1.15}$$

where τ_{ij} is the topological distance, i.e. the number of edges, between i and j, features non-negative edges weights provided that $\mathbf{D}$ is a distance matrix. The problem of minimizing (1.15) such that $\boldsymbol{\tau}$, the topological distance matrix, or Path-Length Matrix (PLM), encodes an UBT is called the Balanced Minimum Evolution Problem (BMEP).

The BME has proven to be a significantly superior choice of criterion than other distance-based methods. This fact motivated significant research effort over the past two decades, leading to both improved solution methods and a deeper understanding of the mathematical properties of the BMEP. It is only recently that the first Integer Programming (IP) formulation for the problem was proposed by Catanzaro et al. (2024). Computing high-quality solutions to the BMEP by exploiting this breakthrough will be the central subject of Chapter 4. There, we will propose several heuristic methods that employ well-known approaches in OR, in combination to the recent IP formulation, to compute improved UBT trees under the BME criterion.

1.5 Structure of the thesis

The next chapters of this thesis are structured to follow the order in which the related articles have been published or submitted for publication.

Chapter 2 presents the results published in:

> Dehaybe, H., Catanzaro, D., & Chevalier, P. (2024). Deep Reinforcement Learning for inventory optimization with non-stationary uncertain demand. *European Journal of Operational Research*, 314(2), 433–445, `https://doi.org/10.1016/j.ejor.2023.10.007`.
>
> My personal contributions include the original idea, the design and execution of the experiments, and co-authorship of the manuscript.

Chapter 3 presents the results published in:

> Agrell, P. J., Dehaybe, H., & Rodriguez, M. H. (2025). Balancing supply security and decarbonization: Optimizing Germany's LNG port infrastructure under the European Green Deal. *Energy Policy*, 198, `https://doi.org/10.1016/j.enpol.2024.114484`.
>
> My personal contributions include the design of the optimization model, data collection, execution of the experiments, and authorship of the experimental methods and results.

Chapter 4 presents the results published in:

> Catanzaro, D., Dehaybe, H., & Pesenti, R. (2025). New heuristics for phylogeny estimation under the balanced minimum evolution criterion. *Bioinformatics*, 41(7), `https://doi.org/10.1093/bioinformatics/btaf361`
>
> My personal contributions include improvements on the original idea, the design and execution of the experiments, and co-authorship of the methodological and results sections.

Compared to the journal versions of each chapter, the studies presented here are supplemented with brief discussions of further research ideas—either too speculative for publication or conceived after the original submissions—that relate directly to the work carried out in each chapter.

Chapter

TWO

DEEP REINFORCEMENT LEARNING FOR INVENTORY OPTIMIZATION WITH NON-STATIONARY UNCERTAIN DEMAND

Abstract

We consider here a single-item lot sizing problem with fixed costs, lead time, and both backorders and lost sales, and we show that, after an appropriate training in randomly generated environments, *Deep Reinforcement Learning* (DRL) agents can interpolate in real-time near-optimal dynamic policies on instances with a rolling-horizon, provided a previously unseen demand forecast and without the need to periodically resolve the problem. Extensive computational experiments show that the policies provided by these agents compete, and in some circumstances even outperform by several percentage points of gap, those provided by heuristics based on dynamic programming. These results confirm the importance of DRL in the context of inventory control problems and support its use in solving practical instances featuring realistic assumptions.

2.1 Introduction

Recent trends in the literature on Operations Management remarked a growing interest of the scientific community towards *Deep Reinforcement Learning* (DRL) approaches to *Inventory Control* (IC) problems characterized by features and dynamics possibly too hard to be modeled and solved by means of classical dynamic or mathematical programming (Vanvuchelen et al., 2020;

Oroojlooyjadid et al., 2021; Gijsbrechts et al., 2022; De Moor et al., 2022; van Hezewijk et al., 2022). The idea at the core of these approaches consists of representing the value function of a traditional *Reinforcement Learning* (RL) agent by means of a deep neural network. The replenishment policy is approximated by selecting the order quantity with the highest expected value. More advanced methods directly learn the replenishment policy function of a given IC system by means of a second deep neural network, whose weights encode the parameters of the policy itself. The network is then trained to autonomously learn the replenishment policy by interacting with a black-box environment that models the system and other arbitrarily complex assumptions. The earliest works pioneering the use of DRL solution approaches to IC problems considered classical systems characterized by a nontrivial optimal replenishment policy and a stationary demand distribution (Vanvuchelen et al., 2020; Oroojlooyjadid et al., 2021; Gijsbrechts et al., 2022; De Moor et al., 2022; van Hezewijk et al., 2022). This last assumption, however, rarely holds in practice. Here, we introduce a new methodology that aims to extend the use of DRL approaches to IC problems that feature a non-stationary demand and a rolling forecast horizon. This methodology is based on training the policy network on random infinite-horizon problems, provided a forecast for H future periods embedded in the state input. The methodology does not require to train the policy network on forecast data and allows to reuse the learned replenishment policy for many periods despite the non-stationarity of demand. We show here that this methodology can be readily applied to advanced forecast dynamics evolving according to the *Martingale Model of Forecast Evolution* (MMFE) first introduced by Iida and Zipkin (2006) (see Section 2.3.2). Extensive computational experiments show that in the case of the well-solved *Single-Item Stochastic Lot-Sizing Problem* (SISLSP) with backorders and in presence of a simple prior on the trend of the demand (simulated e.g., by means of an erratic uniform pattern without autocorrelation), the proposed methodology provides replenishment policies that, post-training, interpolates close-to-optimal order quantities given a previously unseen realistically trending forecast. These replenishment policies can be also a preferable alternative to a *Dynamic Programming* (DP) baseline policy even in the case of a highly variable MMFE. On harder lost sales problems, instead, the replenishment policies provided by the deep neural network significantly outperform the approximate DP baseline policy when lead times are sufficiently long.

The chapter is organized as follows. In Section 2.2, we review the literature on IC and on previous DRL solution approaches to IC problems. In Section 2.3, we state the non-stationary version of the SISLSP; we describe an approximate baseline solution method based on DP and present a generalization of the problem based on the multiplicative MMFE. In Section 2.4, we present the methodology to solve the non-stationary SISLSP via a DRL approach. In Section 2.5, we describe the computational experiments obtained when testing this methodology on the backorders, lost sales and updating forecasts versions of the problem. Finally, in Section 2.6 we draw some preliminary conclusions and we provide perspectives on future research efforts in Section 2.7.

2.2 Related work

Characterizing the analytic solutions to IC problems engaged the community of Operations Management for decades, especially under the assumption of fixed order costs (Porteus, 2002; Axsäter, 2015). One of the earliest, and historically most important attempts was proposed by Scarf (1960), who first proved the optimality of the (s_t, S_t) policy of a periodic-review single-item lot sizing problem with uncertain non-stationary demand and fixed order cost. Three years later, Iglehart (1963) extended these results to the case of stationary infinite-horizon problems. Subsequent research efforts focused on generalizing these results, by relaxing some assumptions or by introducing other features. Examples include the introduction of unimodal inventory costs (Veinott, 1966) or concave order costs (Porteus, 1971). These results only apply under the assumption of a stationary demand, which unfortunately rarely holds in practice (Graves, 1999). Further notable attempts to characterize the solutions to inventory management problems concerned the lost sales version of the single-item problem, introduced by Karlin and Scarf (1958). In this context, the (s, S) policy is only proven to be optimal in the zero lead time case with stationary demand distributions (Veinott, 1966). Under more general assumptions, however, the optimal policy is unknown. This fact has justified the frequent use of heuristics and DP-based solution approaches in the literature, such as those described in Nahmias (1979) and Hill and Johansen (2006).

Several heuristic approaches have been proposed in the literature on IC to compute the policy parameters of the SISLSP under non-stationary uncertain demand. Askin (1981) proposed a myopic heuristic procedure to compute the (s_t, S_t) parameters, by selecting the number of periods to cover with the next order quantity. Bookbinder and Tan (1988) discussed the use of alternative policies to the optimal (s_t, S_t) to reduce both computational complexity and planning instabilities. The authors showed that suboptimal policies offer near-optimal performances under a rolling-horizon setting where the parameters are recomputed at every period. Bollapragada and Morton (1999) proposed to speedup the computation of the parameters of the (s_t, S_t) policy by recursively approximating the problem with a stationary demand. Rossi et al. (2015) proposed a mixed integer linear programming formulation for the SISLSP under the *static-dynamic uncertainty strategy* (i.e. the order periods are decided in advance) with both lost sales and backorders. Xiang et al. (2018) used a similar approach for the *dynamic-dynamic uncertainty strategy* (i.e. the order decision is taken for the current period only). Finally, Dural-Selcuk et al. (2020) studied the problem under a receding-horizon setting by using the static-dynamic and the *static uncertainty strategy* (i.e. the order quantities are also decided in advance). That is, the strategy parameters are recomputed at every period but the forecast window shortens to be 1 at the last period. The authors showed that even the static uncertainty strategy becomes competitive in this setting.

In this chapter we also consider the case of updating forecasts under an MMFE framework. This version of the IC problem was introduced in the literature by Graves et al. (1986), who first described a model that enables the update of the forecast in a multi-stage production process. The acronym MMFE was subsequently proposed by Heath and Jackson (1994) to refer to a general probabilistic model of the evolution of forecasts through time. The authors showed the relevance of MMFE as a way of improving forecasting techniques in a simulation study based on a firm's historical data. Dong and Lee (2003) showed that the base-stock policy remains optimal in the serial system of Clark and Scarf (1960) when the demand is modeled by a MMFE. Iida and Zipkin (2006) developed a computational approach to approximate optimal base-stock policies for a single-item problem. Finally, Norouzi and Uzsoy (2014) derived a better characterization of a MMFE demand process by modeling the evolution of the conditional covariance.

The difficulty posed by the introduction of additional dynamics to classical inventory models motivated several studies to consider them under the new DRL lens. Vanvuchelen et al. (2020) investigated the use of the ubiquitous *Proximal Policy Optimization* (PPO) DRL algorithm to address the joint replenishment problem with a two-dimensional action-space. The approximated policy is a parametric multinomial logistic distribution of the probability of each possible pair of quantities. Oroojlooyjadid et al. (2021) used the *Deep Q-Network* (DQN) algorithm to address the beer game problem, a serial multi-echelon problem commonly used to demonstrate the bullwhip effect in a supply chain. The DQN algorithm approximates only an action-value function, mapping state-action pairs to an expected return. The policy is to select the action with the highest value estimation in the current state by complete enumeration of the action-space. Gijsbrechts et al. (2022) proposed the use of the Asynchronous Advantage Actor-Critic algorithm to address a single-item lost sales model, a dual-sourcing model and a multi-echelon model. The authors proposed to approximate the stochastic policy with a neural network that takes the state of the inventory system as input and predicts the parameters of a categorical distribution over each possible order quantity as an output. De Moor et al. (2022) used the DQN algorithm and a transfer learning method called reward-shaping to stabilize the learning of perishable-goods inventory policies by incorporating knowledge from well-performing heuristics. Finally, van Hezewijk et al. (2022) addressed a stationary problem with capacity constraints and setup costs by adding feasibility masks to the action space. Boute et al. (2021) provided a survey of existing approaches in the literature and discussed other future research directions and challenges for the use of DRL in IC.

2.3 Problem statement

We consider here a rolling-horizon version of the SISLSP under a non-stationary uncertain demand, introduced by Scarf (1960). We study the problem when a finite-horizon forecast is given (an hypothesis necessary to accommodate a non-stationary demand), but assuming at the same time that the process will continue beyond the horizon. This is usually addressed by repeatedly solving a finite problem at every time period (Axsäter, 2015). We start by stating the SISLSP and then we present a DP method that is used to approximate the solution to the SISLSP when considering backorders.

At period t, the forecast for the H upcoming periods is given by a sequence of demand distributions

$$F_{t:t+H-1} := \{D_k : k \in t, \ldots, t+H-1\}$$

of the exponential family (Schäl, 1976), with respective probability density functions ϕ_k. This model assumes that

1. the demand distributions are *non-stationary* (i.e. they are not identical across periods);
2. a fixed order cost is incurred to place orders of non-zero quantities;
3. the lead time L between order and delivery is a fixed integral number of periods;
4. unsatisfied demand is back-ordered;
5. there is a per period discount factor $0 < \gamma < 1$.

The process starts with an initial inventory level z_1. At the beginning of each period, one must decide to raise the inventory position by ordering a quantity q_t. The inventory reaches a level $z_t + q_{t-L}$, available to satisfy the demand of the period that realizes according to its distribution, $d_t \sim D_t$. The inventory level thus reaches $z_{t+1} = z_t + q_{t-L} - d_t$. For now, we only consider the case of back-ordered unsatisfied demand, i.e., we assume that customers unsatisfied by the lack of available inventory are willing to wait for a replenishment (the inventory level z_t may be negative). Finally, a new forecast demand distribution D_{t+H} becomes available.

The classical version of this problem proposed by Scarf (1960) includes: a linear holding cost h per positive unit of inventory carried between two periods; a linear backorder cost b per negative unit of inventory carried between two periods (i.e. unsatisfied demand); and a fixed order cost K if an order occurred in this period. A linear order cost c per unit ordered can be considered but it is often set to 0, as observed in Pochet and Wolsey (2006).

Let $y_t := z_t + \sum_{k=1}^{L} q_{t-k}$ be the *inventory position* in period t before ordering and $w_t := y_t + q_t$ the inventory position right after ordering but before the demand realizes. Given an order q_t, the expected holding and stock-out costs at period $t + L$, when the order arrives, is the convex function

$$I_{t+L}(w_t) = \begin{cases} \displaystyle\int_0^{w_t} h(w_t - \xi)\phi_{t:t+L}(\xi)d\xi + \int_{w_t}^{\infty} b(\xi - w_t)\phi_{t:t+L}(\xi)d\xi & w_t \geq 0 \\ \displaystyle\int_0^{\infty} b(\xi - w_t)\phi_{t:t+L}(\xi)d\xi & w_t < 0 \end{cases} \tag{2.1}$$

where $\phi_{t:t+L}$ is the probability density function of the sum of the demands from periods t to $t + L$. The ordering cost is

$$P(q_t) = \begin{cases} K + c \cdot q_t & q_t > 0 \\ 0 & q_t = 0. \end{cases} \tag{2.2}$$

The *Single-Item Stochastic Lot-Sizing Problem* (SISLSP) with a starting inventory level y_t can, then, be formulated with the following functional equation of the expected value of the discounted cost

$$C_t(y_t) = \min_{q_t \geq 0} \{P(q_t) + I_{t+L}(y_t + q_t) + \gamma E_{t+1}(y_t + q_t)\} \tag{2.3}$$

where

$$E_t(w_{t-1}) = \int_0^{\infty} C_t(w_{t-1} - \xi)\phi_t(\xi)d\xi \tag{2.4}$$

is the expected total cost for periods t to H if period $t-1$ reached the inventory position w_{t-1}. The boundary condition $C_{t+H-L}(\cdot) = 0$ is equivalent to assuming that the process stops beyond the horizon. When the forecast horizon H is long enough, this condition is unimportant as the most distant forecasts are insignificant in the computation of the policy. This, therefore, can be used as a good approximation of the infinite-horizon problem.

In the lost sales version of the SISLSP, negative inventory levels are brought to zero after the *shortage* costs (instead of the *backorder* costs) have been incurred. This change dramatically impacts on (2.1) by making its expression hard to compute in practical time. In that case, one may no longer use the convolution $\phi_{t:t+L}$ of periodic demand distributions and must instead resort to nested integrals to compute the expected inventory costs.

2.3.1 An approximate dynamic programming algorithm for the SISLSP

In the case of backorders, Scarf (1960) proved that the optimal policy to the finite-horizon version of the SISLSP consists of checking whether the inventory position y_t is below a threshold s_t: if this should be the case, then order up to the inventory level $q_t = S_t - y_t$; otherwise, do not order. Because the demand is non-stationary, the (s_t, S_t) values differ for each period (Iglehart, 1963).

The sequence of optimal (s_t, S_t) parameters to problem (2.3) can be approximated by means of a DP algorithm that was first briefly described by Bollapragada and Morton (1999) as a benchmark to evaluate their heuristic. This algorithm, whose pseudocode is outlined in Algorithm 1, starts at period $H-L$, by computing the policy parameters (s_{H-L}, S_{H-L}); then, it proceeds backward in time. The algorithm relies on the K-convexity of the function

$$G_t(w) = c \cdot w + I_{t+L}(w) + E_{t+1}(w), \tag{2.5}$$

which is the expected cost function of reaching the inventory position w when ignoring the fixed ordering cost (Scarf, 1960). At each iteration t, a line search on (2.5) with a fixed step size δ starts from an upper bound to S_t to find a (sub)optimal inventory position S_t; then it continues searching until finding s_t at the inventory position costing K more than that of S_t. The task of computing $C_{t+1}(w_{t+1})$ is simplified by the fact that the pair (s_{t+1}, S_{t+1}) is known from the previous iteration and the values of $E_{t+2}(w_{t+1})$ can be memoized from the previous iteration for all w_{t+1}. E_{t+1} can be computed with any numerical integration tool. We used a simple Riemann sum on an interval upper bounded by a large quantile of ϕ_{t+1}. The closed-form expression of Equation (2.1) can be obtained by means of a symbolic solver such as Wolfram Mathematica.

The introduction of the lost sales in the SISLSP breaks the K-convexity property at the core of Scarf's optimality proof of the (s_t, S_t) policy. The policy computed by assuming backorders, however, may constitute a good heuristic to approximate the optimal policy for the lost sales case (Axsäter, 2015). This is especially true when the h/b ratio is low, as stock-outs will be rare and negligible in the presence of a fixed order cost. In their experiments, Hill and Johansen (2006) concluded that the (s_t, S_t) policy is near-optimal, and likely optimal in many cases. In the rolling-horizon framework, only the (s_1, S_1) pair is used and the problem must be resolved at every period by using the new forecast window. Due to the boundary condition, the discretization of the units, and the approximate integration of E_t, this method is approximating the optimal policy as the length of the window increases and as the step size decreases. In the case of a short forecast horizon, the finite-horizon boundary condition is no longer a good approximation. We circumvent this issue by augmenting the forecast with 32 stationary periods equal to the average demand.

2.3.2 The Martingale model of forecast evolution

The *Martingale Model of Forecast Evolution* (MMFE) was introduced by Heath and Jackson (1994) as a framework to model the dynamics of forecasting. We will exploit the MMFE in the subsequent sections to model the periodic revision

Algorithm 1: Approximate Policy based on DP

Input: The instance parameters: $L, b, K, h, F_{1:t+H-1}, \gamma, c$; a step size: δ; an upper bound to S: UB_S

for $t \in [H-L, \dots, 1]$ **do**
  descending $\leftarrow$ true;
  $w \leftarrow UB_S$;
  $g \leftarrow G_t(w)$;
  $S_t \leftarrow w$;
  $g_{min} \leftarrow g$;
  while $g_{min} \leq g + K \vee descending$ **do**
    $w \leftarrow w - \delta$;
    $g \leftarrow G_t(w)$;
    if $g \leq g_{min}$ **then**
      $g_{min} \leftarrow g$;
      $S_t \leftarrow g$;
    else
      $descending \leftarrow false$;
  $s_t \leftarrow w$

of the demand that may occur in practice. For the sake of completeness, we briefly recall here the fundamental aspects of the MMFE and we introduce some basic notation and definitions that will prove useful in the remainder of the chapter.

Let $D_{k,t}$ denote the distribution that models the demand forecast for period k, given the information available at time $t \leq k$. The mean of $D_{k,t+1}$ (i.e., the revised forecast, hereafter denoted as $\mathbb{E}[D_{k,t+1}]$) is obtained by multiplying the mean of $D_{k,t}$ by an update factor u_k, i.e.,

$$\mathbb{E}[D_{k,t+1}] = \mathbb{E}[D_{k,t}]u_k, \quad \forall k \geq t+1.$$

In the context of the MMFE, this updating policy is referred to as the *multiplicative update* (Iida and Zipkin, 2006). Another type of updating policy available in the MMFE is referred to as the *additive update* and consists of adding a *revision term* $\Delta\mu_k$ to $\mathbb{E}[D_{k,t}]$, i.e., $\mathbb{E}[D_{k,t+1}] = \mathbb{E}[D_{k,t}] + \Delta\mu_k$. Because the MMFE only considers updates of the expected demand, we assume that the standard deviation is also updated such that the *Coefficient of Variability* (CoV) (i.e., the ratio $\mathbb{E}[D_{k,t}]/\sigma_{k,t}$, where $\sigma_{k,t}$ represents the standard deviation of $D_{k,t}$) remains constant. In general, u_k is a random variable with positive support and unit mean and may be correlated to $u_{j \neq k,t}$. In this work we will assume that u_k is log-normally distributed, i.e.,

$$u_k \sim Logn(-\frac{\Sigma_k}{2}, \Sigma_k), \tag{2.6}$$

where the exponential decaying of the variance

$$\Sigma_k = \Sigma_1 \cdot \varkappa^{k-t-1} \tag{2.7}$$

is used to model the fact that less information is available to update distant forecasts. In (2.7), Σ_1 denotes the variance of u_{t+1} and $\varkappa \in (0, 1)$ is a scalar representing the decay factor. As in Iida and Zipkin (2006), we consider no correlation between the update factors belonging to different periods.

Henceforth, we refer to the version of the SISLSP that includes a periodic revision of the forecast according to the above framework as SISLSP-MMFE. No solution method is currently described in the literature to solve the SISLSP-MMFE under fixed ordering cost and a base-stock policy would perform poorly under these settings. To identify a baseline policy for the SISLSP-MMFE, we used the DP method presented in Section 2.3.1.

2.4 A DRL approach to non-stationary versions of the SISLSP

In this section, we present a DRL approach to learn replenishment policies for the non-stationary versions of the SISLSP described in Section 2.3. Before proceeding, we recall that *Reinforcement Learning* (RL) methods require an interactive simulation of a Markov Decision Process, called an *environment*, to learn on. In Section 2.4.1, we present a possible approach to construct an environment in the case of a non-stationary demand and we provide a toy example in Section 2.4.2. In Section 2.4.3, we briefly recall the foundations of the *Proximal Policy Optimization* (PPO) algorithm, much used in the literature of DRL to learn neural network-based control policies. We rely upon this algorithm to carry out computational experiments on the non-stationary SISLSP. In Section 2.4.4, we discuss some issues introduced by the presence of a fixed order cost and propose a possible strategy to overcome them.

2.4.1 Encoding the SISLSP environment

An environment is a Markov Decision Process that simulates a sequential decision problem to solve and is associated with a set $\mathcal{X}$ of states that it may assume. An *agent* is an entity that is distinguished from the environment and that can learn from it by means of *interactions*, i.e., actions that cause the transition of the environment from the current state to a new one and the return of a reward. Specifically, let $\mathcal{A}$ denote the set of actions that an agent can undertake. Then, a transition of the environment occurs at time t when the agent observes the state $x_t \in \mathcal{X}$ and undertakes an action $a_t \in \mathcal{A}$ based on a policy $\pi : \mathcal{X} \to \mathcal{A}$. The environment then returns a scalar reward r_t drawn from a *reward function* $r : \mathcal{X} \times \mathcal{A} \to \mathbb{R}$ and moves to a new state x_{t+1}. The agent-environment interaction gives rise to a sequence of transitions (x_t, a_t, r_t, x_{t+1}) until a terminal state is reached, or the simulation is arbitrarily stopped in the case of an infinite Markov Decision Process. The sequence of states, actions and rewards from the initial to the last state is called an *episode*. The *transition dynamics* of the process describe how the environment will transition depending on its state x_t, the action a_t, and possibly some other stochastic component. The goal of an agent is to learn the policy that maximizes its expected discounted sum of rewards. Thus, in order to formulate an IC problem as a RL environment, we need to specify the sets of states and actions, the reward function, and the transition dynamics.

States

A particular state is an encoding that includes all the information about past agent-environment interactions (i.e. it has the *Markovian* property, Sutton and Barto (2018)). In this chapter, we explicitly embed the parameters of the forecast demand distributions in the state, in addition to the inventory position:

$$x_t = (M(D_t), M(D_{t+1}), \ldots, M(D_{t+H-1}), z_t, q_{t-L}, \ldots, q_{t-1})$$

where H is a forecast horizon and $M(D)$ returns all the necessary parameters to describe the distribution D (e.g. the mean and the standard deviation in the case of Gaussians).

Actions

The simplest encoding of the action is the direct embedding of the order quantity $a_t = (q_t)$. In this chapter, we instead consider the use of an alternative encoding $a_t = (s_t, S_t)$. The environment is programmed internally to reinterpret this pair as an order quantity q_t. As explained in Section 2.4.4, this is crucial to the stability of the DRL algorithm due to the discontinuous nature of the optimal policy of lot sizing problems with fixed costs.

Reward function

The environment returns the ordering and inventory costs paid at time t as the negative rewards r_t computed from the function $r(x_t, a_t)$ that depends on the state of the environment and the action undertaken by an interacting agent. Ordering costs are deterministic and given by Equation (2.2). The inventory costs equal $h \cdot \max(0, z_{t+1}) + b \cdot |\min(0, z_{t+1})|$ where $z_{t+1} = z_t + q_{t-L} - d_t$ where d_t is the realized demand at time t.

Although Scarf's results can be generalized to other cost functions, as shown in Porteus (1971), here we only consider the case of linear ordering, holding, and stock-out costs, mostly because the baseline DP policy parameters are easier to compute. Any different assumption would just involve changing accordingly the reward signal of the environment.

Transition dynamics

The dynamics of the of environment are characterized by a *state-transition function* that generates one next-state given an action. In our case, given the action (q_t), the next state is simply

$$x_{t+1} = (M(D_{t+1}), \ldots, M(D_{t+H}), z_{t+1}, q_{t-L+1}, \ldots, q_t).$$

That is, the forecast window is shifted by one period, the inventory level is updated according to the realized demand and the quantity ordered L periods ago, and the on-order queue is appended with the latest order (q_t) quantity. An implementation of an environment with lost sales needs only to set $z_{t+1} = 0$ if it is negative at the end of period t. In the case of a SISLSP-MMFE, the parameters of the forecast are multiplied by a random vector at the end of each period, as we explain in Section 2.3.2. Since there is no terminal state in infinite problems, interactions may optionally be interrupted to form episodes after a fixed number of periods.

Initial state distribution

RL algorithms typically learn from transition data generated by several episodes of interactions with an agent. When starting a new episode, the environment may be reset to a fixed or to a random initial state. Formally, the state variables may be partitioned into subsets characterized by their independent distributions (univariate for single-element subsets, multivariate otherwise). For example, the parameters of each forecast distribution may be independently drawn for each period or follow an autocorrelated process. For the sake of simplicity, we consider here just univariate initial distributions. This is without loss of generality as RL methods do not require this assumption. We denote $\bar{Z}_1$ as the distribution of the initial inventory level (z_1); $\bar{D}_t^n$ for the n^{th} parameter of the forecast distribution at period t; and $\bar{Q}_k$ for the on-order quantity ordered at time $1-k$ (q_{1-k}). The family of each distribution need not to be the same for each state variable and may be discrete or continuous. Notably, a subset of state variables may be initialized deterministically to a fixed number.

Randomizing the initial state of the forecast is a core feature of the proposed approach for two reasons.

First, the randomization of the initial state is an instance of an *exploring start* that allows the agent to explore the state space more efficiently and thus better generalize (Sutton and Barto, 2018). Some states may otherwise be unlikely to be reached with the current policy and consequently prevent the agent from learning their value.

Second, this ensures that the agent does solve the rolling-horizon problem with a forecast window of H periods and no more. If the forecasts beyond the horizon are predictable, because always identical from episode to episode, then the agent will learn it and its policy will solve a longer-horizon problem. Randomizing the initial state of the forecast parameters acts as an uninformative prior that provides little information to extrapolate beyond the horizon.

A key implication of this latter fact is that, once trained, the agent can be used to compute the order decision indefinitely, as long as the training environment remains relevant to the actual inventory system. To the best of our knowledge, this is the first method that achieves this in an infinitely-rolling-horizon setting with non-stationary demand. Consequently, as a by-product, the policy learned by the agent does not learn to solve a single instance of the original problem, but rather all instances that lie on the support of the initial state distribution.

2.4.2 A toy example

We show the construction and a transition of a rolling-horizon RL environment following the steps outlined in Sections 2.4.1. Consider an instance of the SISLSP problem arising when setting $H = 4$, $L = 1$, $K = 100$, $h = 1$, $b = 10$. The demand at period t is forecasted with a Poisson distribution $Poi(\lambda_t)$ with parameter λ_t. The state of the environment at time t will thus be

$$x_t = (\lambda_t, \lambda_{t+1}, \lambda_{t+2}, \lambda_{t+3}, z_t, q_{t-1})$$

The initial state distribution used to generate λ_t, $\forall t \geq 1$ is a single binomial distribution to randomly generate the Poisson parameter at each period. We choose the following initial state distributions:

- $\lambda_t \sim \bar{D}^\lambda := B(20, 0.5), \forall t,$
- $z_1 \sim \bar{Z}_1 := U(-20, 20),$
- $q_0 = 5,$

where U denotes a uniform distribution, B a binomial distribution, and q_0 is a deterministic initial state variable. Following the initial distribution outlined above, we generate the initial state

$$x_1 = (7, 13, 10, 15, -2, 5)\,.$$

The policy of the agent returns the action $a_1 = (19)$, giving rise to the transition that follows. The 5 on-order quantities become available and the realized demand $d_1 = 6$ is randomly drawn from the Poisson distribution $Poi(7)$, thus the new inventory level becomes $z_2 = -2 + 5 - 6 = -3$. Finally, a new forecast parameter for the demand in period 5 becomes available and generated from the initial distribution: $\lambda_5 \sim B(20, 0.5) = 5$. As a result, the new state is

$$x_2 = (13, 10, 15, 5, -3, 19),$$

and the reward is

$$\begin{aligned} r_1 := r(x_1, a_1) = &- (100 + 10 \cdot |\min(0, -3)| \\ &+ 1 \cdot \max(0, -3)) = -130. \end{aligned}$$

This completes a single transition.

Should the environment be that of a lost sales model, the transition to the next state finishes by setting $z_2 = \max(z_2, 0)$. In the case of forecast updates according to a MMFE and given Σ_k for all $k \in 2, \ldots, 5$, the transition requires to randomly generate update factors u_k for $k \in 2, \ldots, 5$ according to Equation (2.6). The revised forecast is obtained by setting $\lambda_k = \lambda_k \cdot u_{k+t}, \forall k \in 2, \ldots, 5$.

2.4.3 The Proximal Policy Optimization algorithm

The *Proximal Policy Optimization* (PPO) algorithm is a well-known DRL algorithm introduced by Schulman et al. (2017) to train neural networks. The PPO algorithm belongs to the family of *actor-critic* algorithms as it approximates two functions by using neural networks. The first network, called the *critic*, approximates the state value function that estimates expected discounted sum of future rewards of a state for a given policy π

$$V_\pi(x_t) = \mathbb{E}\left[\sum_{k=t}^{\infty} \gamma^{k-t} r\left(x_k, \pi\left(x_k\right)\right)\right] \tag{2.8}$$

where $0 < \gamma < 1$ is a discount factor and the expectation is with respect to the transition probabilities and the stochastic policy distribution.

The second neural network, called the *actor*, approximates the policy $\pi(x_t)$. The PPO algorithm learns a *stochastic policy*, in contrast to a deterministic policy. In particular, the actor does not directly predict the action, but the parameters of a distribution from which the action is sampled. This allows a local exploration of the actions during training. At test time, only the mode of the learned policy is used to approximate the optimal policy. The goal of this network (agent) is to find the policy π that maximizes $J(\pi) = \mathbb{E}\left[V_\pi(x_1)\right]$ with respect to the initial state distribution.

The policy distribution may be discrete or continuous. We differ from existing IC literature in that the actor policy is a continuous multivariate Gaussian distribution with diagonal covariance, whereas Vanvuchelen et al. (2020) use a discrete multinomial distribution where each class is a possible action from a discretized set of actions. We use this approach for two reasons. First, it is more data efficient because a continuous distribution naturally incorporates the fact that two close order quantities lead to similar results. In contrast, an actor with a multinomial distribution must try each action several times to learn their values separately. Second, it is more scalable in the dimensions of $\mathcal{A}$. With a multinomial policy distribution, the number of parameters to estimate grows exponentially and the approach becomes untractable.

We propose here that the actor approximates the parameters of a Gaussian distribution to sample a (s_t, S_t) pair. In other words, the output of the neural network is

$$\pi(x_t) = (\mu_{s_t}, \mu_{S_t}, \sigma_{s_t}, \sigma_{S_t})$$

and during training

$$a_t = (s_t \sim N(\mu_{s_t}, \sigma_{s_t}), S_t \sim N(\mu_{S_t}, \sigma_{S_t}))\,.$$

At test time, the action is $a_t = (\mu_{s_t}, \mu_{S_t})$. Actions that are not feasible are projected to the nearest feasible solution (e.g. negative order quantities become a no-order action). For more details on how the PPO algorithm trains its neural networks, we refer the reader to the Section 1.2 of this thesis and to the original publication by Schulman et al. (2017).

Continuous DRL is, however, typically less stable and harder to train, Engstrom et al. (2020) explain how one must carefully pay attention to implementation details. Additionally, this approach does not interact well with the presence of fixed order costs due to the discontinuous nature of the optimal policy function. We discuss this aspect in Section 2.4.4. As in Schulman et al. (2017), we use the Generalized Advantage Estimation (GAE) method of Schulman et al. (2016) to estimate the advantage of the actions sampled during training. This method uses an exponentially weighted average of the learned value of all the state encountered after an action to reduce the bias induced by the estimation error of the critic neural network.

2.4.4 Learning under fixed order costs

The presence of a fixed order cost makes much harder the task of learning the correct value estimates by using one-step bootstrapping. Let (x_t, a_t, r_t, x_{t+1}) be a transition where a non-zero quantity q_t was ordered. The incurred reward r_t is very low due to the large fixed cost whereas the benefit of an increased inventory will only be visible L periods in the future. The short-term advantages (computed with the GAE) are negative if the values of x_{t+1} and the subsequent states have not been well learned yet, as it is the case at the beginning of the training phase. This fact discourages the policy from ordering. Which in turn reduces the chances of encountering high inventory states and thus increases the time before the critic effectively learns that ordering is beneficial in the long run. This effect is amplified when the fixed order cost or the lead time increases. Furthermore, the exploration of the action-space via local sampling around a mean action discourages to ever move from ordering 0 to a large batch quantity. Around a mean of, say, $q_t = 0$, the most likely positive order quantities are small. These small batches will reinforce the agent towards never ordering because they incur the fixed cost without significantly improving the inventory position. It is therefore crucial that the critic rapidly learns the value of high inventory position states. For this reason, we train the critic for more epochs per iteration than the actor and use a warmup time to prevent the actor from diverging due to incorrect value estimation at the beginning of training, as detailed in Section 2.5.1. Finally, as known from the work of Scarf (1960), the optimal q_t policy with respect to the inventory position is discontinuous since q_t is a function of the (s_t, S_t) parameters and the inventory position. A neural network, on the other hand, is a continuous function approximator. Approximating discontinuous functions typically leads to extremely large gradients and is source of high instability in deep learning. A possible way to cope with this difficulty consists of parameterizing the actor network to output a distribution for (s_t, S_t) instead of (q_t). This parameterization has the benefit of having much smoother gradients. It is worth noting here that this reparametrization does not reduce the space of policies that the agent can approximate. In the literature on IC, the (s_t, S_t) policy parameters are computed without the knowledge of the inventory position w_t. This is a fundamental difference with this DRL approach where the output (s_t, S_t) distribution of the neural network is still conditioned on the full state vector, including on the inventory position, and thus the actor can output parameters that vary with respect to w_t. As such, it is still a universal policy approximator and it does not limit the method to the (s_t, S_t) policy in its classical IC meaning; it is merely a smoothed way to indirectly express a desired order quantity that allows a special treatment of the $q_t = 0$ decision.

2.5 Computational experiments

In this section, we empirically evaluate the ability of the DRL approach to learn a policy for the versions of the non-stationary SISLSP described in Section 2.3. We consider, in particular, three series of experiments. The first series aims to measure the performance of the approach on problem instances where the DP baseline is approximately optimal. We show that a single agent trained on an environment with an erratic demand prior is able to interpolate close-to-optimal policies for several realistic trends that are not encountered during the training phase. We also show that the performance of the DRL approach is not impacted by a reduction of the forecast horizon. In the second series, we experiment on the case where the demand forecast is periodically revised according to a MMFE. Finally, we show in the last series that the approach can outperform the heuristic DP baseline for some instances of the lost sales

Hyperparameters	Values
Critic learning rate	10^{-4}
Actor learning rate warmup bounds	$10^{-6} \rightarrow 10^{-4}$
Actor warmup factor	$\sqrt[2000]{100}$
Actor learning rate decay bounds	$10^{-4} \rightarrow 10^{-5}$
Actor learning rate decay	$\sqrt[8000]{0.1}$
Maximum gradient norm	0.5
Discount factor (γ)	0.99
GAE weight (λ)	0.9
Clip decay bounds (ϵ)	$0.2 \rightarrow 0.05$
Entropy weight (β)	10^{-2}
Episodes per iteration	30
Critic epochs	5
Actor epochs	1
Batch size	256
Training iterations	10000

Table 2.1: Values of the hyperparameters of the PPO algorithm.

version of the problem.

We start by presenting the details of our implementation of the PPO algorithm in Section 2.5.1. We then carry out the experiments on near-optimally solved instances in Section 2.5.2, the experiments on the SISLSP-MMFE instances in Section 2.5.3, and the experiments on lost sales instances in Section 2.5.4. Finally, we provide visual insights on the learned policies in Section 2.5.5.

2.5.1 Implementation details

The PPO algorithm is known to be fairly insensitive to the choice of hyperparameters. We used the recommended hyperparameters in most cases and we lowered the number of simulations and neural network update epochs as much as it was possible without losing performances (Schulman et al., 2017; Engstrom et al., 2020; Cobbe et al., 2021). They are detailed in Table 2.1. The only hyperparameter that required tuning was the gradient step size (learning rate) used to update the neural networks. We observed that different learning rates for the actor were needed depending on the problem parameters. To address all instances with the same hyperparameters, we used an exponentialy decaying learning schedule to vary its value during training. To ensure stability and better convergence of the actor, we also used an exponential warmup during the first 2000 training iterations. This warmup time is necessary to let the critic learn before letting the actor make too large updates, this addition proved to be critical for the stability of the method. The critic has a constant learning rate. We also linearly decayed the ϵ clipping parameter (see Schulman et al. (2017)) to help convergence at the end of training. We used the state-

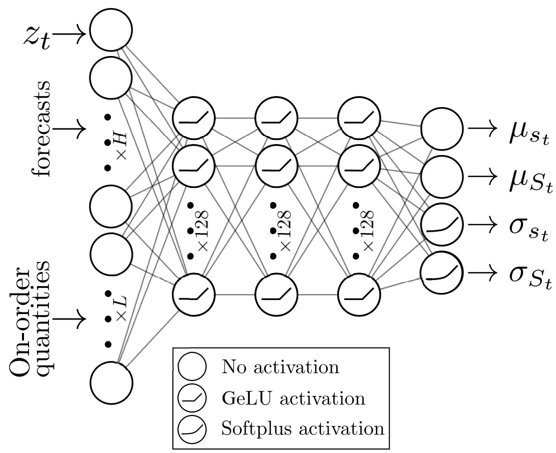


Figure 2.1: Architecture of the actor neural network.

of-the-art stochastic gradient descent optimizer to train the neural networks, Adam (Kingma and Ba, 2015).

We implemented both neural networks as Multi-Layer Perceptrons (MLP) composed of three hidden layers with 128 neurons having the GeLU activation function (Hendrycks and Gimpel, 2016) and randomly initialized by following the orthogonal scheme (Saxe et al., 2014). The rationale underlying the use of an MLP rather than more recent sequential architectures such as an LTSM or a Transformer is that the performance cost is not justified in our situation where the state is Markovian. Such architectures are mainly used in DRL when dealing with partially observable states. The critic network is characterized by a linear output activation. The output neurons of the actor network that estimate the mean of the policy parameters have linear activations and those that estimate the standard deviation have a standard *softplus* activation ($sp(x) := log(1+e^x)$). The complete architecture of the actor network is shown in Figure 2.1. The critic network is identical except for the output layer that has a single neuron with no activation.

At each iteration, episodes of 52 periods are simulated by using the latest policy to generate transitions. Then, both the actor and the critic networks are updated. The actor network is updated for a single epoch per iteration while additional epochs are performed for the critic network, as suggested by (Cobbe et al., 2021) and discussed in Section 2.4.4. The advantages are estimated by using GAE (Schulman et al., 2016) with $T = 52$, the full episode length. We scaled the rewards to zero mean and $1/T$ standard deviation with a running estimate of their moments to normalize the value network (Andrychowicz et al., 2021). The gradient is clipped to have a maximum norm of 0.5. The code is written in Julia 1.8.1 (Bezanson et al., 2017) (LLVM 13.0.1) and includes the deep learning package Flux 0.13 and its internal auto-differentiation package,

Experiment	GPU	CPU	DDR4 RAM	Time/agent
Near-optimality	4 NVIDIA RTX 3090	AMD EPYC 7352 24 cores (2.3 GHz)	515Gb	53
Updating Forecasts	2 NVIDIA A10	Intel Xeon Gold 6346 12 cores (3.1 GHz)	98 Gb	20
Lost Sales	2 NVIDIA RTX 3090	AMD EPYC 7352 12 cores (2.3 GHz)	98 Gb	25

Table 2.2: Hardware configurations of the different experiments and approximative time (expressed in minutes) to train one agent. Each GPU was dedicated to train one agent at a time.

Zygote (Innes, 2018). The three experiments ran on different cluster nodes, and thus with varying hardware configurations that are detailed in Table 2.2. Environment simulations were performed in parallel on the CPU cores while gradient computations were carried out on a GPU. We trained multiple agents in parallel by allocating one GPU to each task. The training time is highly hardware dependent: experiments that ran with more CPU cores per agent (number of GPUs divided by the number of cores) took up to 2.5 less time to complete. The implementation, the experimentation datasets, the experimental data, and the code of the fully reproducible experiments are available at `https://github.com/HenriDeh/DRL_MMULS/tree/single-item`.

2.5.2 Near-optimality

In this section, we describe the first series of experiments that we carry out on SISLSP instances that are approximately optimally solved by the DP baseline presented in Section 2.3.1. We first present the testing and the training setups and then we detail the results of the experiments.

Parameters	Experiment values	
Lost sales	false	true
h	1	1
b	$10, \mathbf{25}, 50$	$50, \mathbf{75}, 100$
CoV	$0.1, \mathbf{0.2}, 0.3$	$0.1, \mathbf{0.2}, 0.3$
K	$0, 80, \mathbf{360}, 1280$	$0, 80, \mathbf{360}, 1280$
c	0	0
L	$0, 1, \mathbf{2}, 4, 8$	$0, 1, \mathbf{2}, 4, 8$
H	$4, 8, 16, \mathbf{32}$	$4, 8, 16, \mathbf{32}$

Table 2.3: Parameters of the learning environments. The default values are emphasized in bold.

Training setup

To simplify the experimental setup, we considered the case where all demand distributions are Gaussians with identical Coefficients of Variation (CoV) across periods. This allowed to only embed the means of the Gaussians into the state, as the standard deviation can be inferred. We trained the agents on SISLSP environments with different *environment parameters* to evaluate the sensitivity of the method to their values. We changed the backorder/shortage cost (b), the CoV, the fixed order cost (K), the lead time (L), the forecast horizon (H), and whether sales are lost. The K values were selected to have respective deterministic EOQ of 0, 4, 8 and 16 periods. In the case of lost sales, the shortage cost (b) values were significantly higher than backorder values to ensure the DP baseline is near-optimal, and to avoid ill-posed problem instances where the optimal policy is to never-order. The variable production cost was kept to 0 for simplicity. We allowed one environment parameter to change at a time while the others were kept to default values. Ten agents were trained for each of the 30 combinations described in Table 2.3 (15 of both backorders and lost sales configurations). The initial distributions of the inventory and of the expected demands of the training environments were $z_1 \sim \bar{Z}_1 = U(-\mu{\cdot}(L{+}1), 2\mu{\cdot}(L{+}1))$ and $\mu_t \sim \bar{D^{\mu}} = U(0, 20)$, respectively. The on-order quantities were always initialized to 0.

Testing setup

In order to test the ability of the agents to predict near-optimal policy parameters given non-stationary demand forecasts they never encountered during the training phase, we generated a test dataset of 136-periods-long forecasts with five types of realistic trends, inspired from those used by Rossi et al. (2015): three constant trend, three seasonal trends with different frequencies, a linear growth trend and a linear decline trend. These trends are detailed in Table 2.4. The constant trends test the ability to perform when the average expected demand is not that of the training environment ($\mu = 10$); the linear decline and growth trends model a threefold decrease or increase in demand over the 136 weeks; and the seasonal trends represent sinusoidal demands with one, two and four periods over the two years, respectively. These forecasts were used in combination with the varying training environment parameters (Table 2.3) to generate unique test problem instances.

The policies learned by the agents are evaluated by comparing to two baselines. The first is the (s_t, S_t) policies obtained by using the DP method described in Section 2.3.1, with a step size δ of 1 and always assuming backorders. This baseline is approximately optimal when shortages are backordered and heuristic in the case of lost sales. The baseline policy parameters are computed in a rolling-horizon fashion by executing Algorithm 1 at each period to recover the current period (s_t, S_t) parameters. The second baseline, which we refer to as the *Simple* baseline, computes (s_t, S_t) policy parameters for every period as follows. The threshold s_t is the $b/(b+h)$ quantile of the distribution of the total demand during the lead time. The reorder level S_t is $s_t + EOQ$ where the EOQ is computed with the usual formula $EOQ = \sqrt{2 \cdot D \cdot K/h}$, where D is the average expected demand of the next H periods.

The average return of a policy (baseline or learned) on an instance is estimated with 1000 Monte Carlo simulations of the first 104 periods with an initial inventory of $z_1 = L \cdot \mu$ and no on-order quantities. The performance of a policy for a single instance is its average return over the 1000 simulations. The *gap* of an agent for one instance is the difference between its performance and the baseline performance, divided by the baseline performance.

A trained DRL agent is evaluated on the 8 test instances that combine the test forecasts and its training environment parameters. We used the median of the gaps of the individual instances as a measure of the overall performance of an agent. With 8 forecasts and a total of 15 different environment parametrizations (1 default configuration, 2 b variants, 2 CoV variants, 3 K variants, and 4 L variants, and 3 H variants), for both backorders and lost sales, we effectively tested the DRL approach on 120 unique backorder and 120 unique lost sales instances, each never encountered by the agents during the training phase.

Results

Figure 2.4 (see Appendix 2.A) shows the distribution of the median gap of the agents with respect to the DP baseline over the 8 forecasts. Agents where $H \neq 32$ are not included in this figure. The results showed that the DRL approach performs consistently over repeated trials and thus the boxes are narrow in most environment settings. We only observed agents that are less performing on average in the case of a low setup cost. Specifically, 2 agents out of 240 reach an average gap greater than 4%, one when $K = 0$ and one when $K = 80$.

Table 2.5 reports the average, best and worst median gaps achieved with

Trend	μ_t
Constant μ, $\mu \in \{5, 10, 15\}$	μ
Linear decline	$15 - t \cdot \frac{10}{136}$
Linear growth	$5 + t \cdot \frac{10}{136}$
Seasonal 1	$\left[1 + 0.5 \cdot \sin\left(1 \cdot 2\pi \frac{t}{52}\right)\right] 10$
Seasonal 2	$\left[1 + 0.5 \cdot \sin\left(2 \cdot 2\pi \frac{t}{52}\right)\right] 10$
Seasonal 4	$\left[1 + 0.5 \cdot \sin\left(4 \cdot 2\pi \frac{t}{52}\right)\right] 10$

Table 2.4: Time series used to generate the test forecasts for 8 different trends.

Backorders								Lost Sales							
				DP			Simple					DP			Simple
L	b	K	CoV	Average	Best	Worst	Average	L	b	K	CoV	Average	Best	Worst	Average
2	25	0	0.2	1.11	0.24	1.85	1.44	2	75	0	0.2	0.88	-0.51	5.73	1.29
2	25	80	0.2	1.35	0.73	2.29	-8.88	2	75	80	0.2	1.68	0.71	3.65	-7.04
2	25	320	0.1	1.95	1.47	2.41	-4.69	2	75	320	0.1	2.2	1.71	2.71	-3.33
2	10	320	0.2	1.15	0.74	1.56	-8.67	2	50	320	0.2	1.06	0.73	1.32	-5.59
0	25	320	0.2	0.9	0.72	1.13	-5.38	0	75	320	0.2	0.83	0.62	1.19	-4.1
1	25	320	0.2	0.96	0.65	1.5	-6.14	1	75	320	0.2	1.0	0.71	1.34	-4.64
2	25	320	0.2	1.02	0.87	1.2	-6.63	2	75	320	0.2	1.14	0.86	1.39	-5.04
4	25	320	0.2	0.68	0.5	1.0	-7.36	4	75	320	0.2	0.81	0.6	1.04	-5.85
8	25	320	0.2	1.8	1.36	2.23	-6.1	8	75	320	0.2	1.21	0.54	1.71	-5.26
2	50	320	0.2	0.99	0.74	1.5	-5.65	2	100	320	0.2	1.01	0.65	1.29	-4.88
2	25	320	0.3	0.82	0.7	0.96	-8.0	2	75	320	0.3	0.98	0.72	1.29	-6.41
2	25	1280	0.2	1.39	-0.33	2.22	-4.3	2	75	1280	0.2	1.46	0.48	2.61	-3.1

Table 2.5: Summary of the near-optimality experiments (Section 2.5.2). Each row summarizes the ten agents that were trained with the environment parameters in the four leftmost columns. The four rightmost columns give the average, best, or worst of the ten agents' median gaps (in %) with respect to either the DP method or the Simple heuristic (see the headers).

respect to the baseline methods for each environment parametrization. Overall, the average median gap achieved by the agents with respect to the DP baseline never exceeds 2.2% on each environment. All the agents converged to a policy competitive with the DP baseline and the simple heuristic is greatly outperformed in all settings excepted when $K = 0$, where the simple baseline even outperforms the DP baseline. This result shows that the proposed DRL approach approximates near-optimal policies for a wide range of forecasts trends, unknown to the agent. Figure 2.2 shows how the deep RL approach performs with reduced forecast horizons, ranging from 4 to 32 periods. This has no significant impact on the relative performance compared to the DP heuristic, which, as a reminder, solves for an instance with a forecast augmented with 32 stationary periods equal to the average in-horizon expected demand. This result shows that the choice of a uniform initial state distribution as a prior for the demand beyond the horizon is similar to the use of a stationary boundary condition for the DP method. In the opposite case, we would have observed an increase or a decrease in performance as the horizon shortens.

Figure 2.5 (see Appendix 2.A) shows that the agents appropriately learned most forecast trends. The obvious exception is the *constant 5* trend (see Table 2.4), where most agents approximated a suboptimal policy. Surprisingly, we cannot observe the same effect for the *constant 15* trend. A possible explanation for this difference is that the relative difference between 5 and 10 is greater than between 15 and 10. This suggests that the mean demand during training should at least be close to that of the average real demand (10 here). A second exception is that some agents did not generalize well to the *seasonal 4* trend. The uppermost outliers of this box are all from agents training in environments with a lead time of 8 periods. This indicates that it is due to a difficulty in

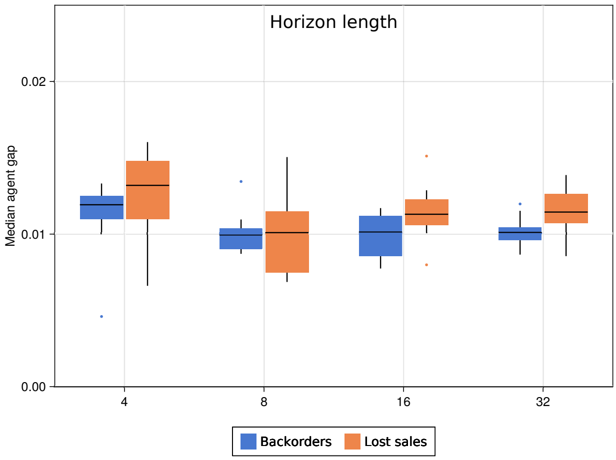


Figure 2.2: Whisker plots of the median agent gaps with respect to the DP baseline for the near-optimality experiments with varying horizons. Each data point represents one agent, each whisker plot thus represents ten agents.

generalizing to different forecasts in this type of environment, rather than a difficulty posed by the forecast trend alone, since we did not see this difference with shorter lead times.

2.5.3 Updating forecasts

We performed experiments with a similar setup to those described in Section 2.5.2 with the addition of forecast updates that follow an MMFE. The training and testing setups as well as the implementation details are almost identical to the first experiment. The only change is that the parameters in Table 2.3 were fixed to their default values. We instead varied the parameters $\Sigma_1 \in \{0.03, 0.05, 0.1\}$ and $\varkappa \in \{0.7, 0.8, 0.9, 0.95, 0.99\}$, this time in a full factorial fashion. The results are plotted in Figure 2.6 (see Appendix 2.A) for both backorders and lost sales. Table 2.6 contains the detailed performance results. The DP method remains a well-performing choice in the case of backorders. However, the DRL approach becomes more interesting in the case of lost sales as the variability increases. This is likely due to poorer performances of the DP heuristic as the likelihood of shortages increases with the variability induced by the forecast revisions.

Backorders					
		DP			Simple
Σ_0	$\varkappa$	Average	Best	Worst	Average
0.03	0.7	0.75	0.48	1.17	-5.65
0.03	0.8	0.6	0.31	0.87	-5.83
0.03	0.9	1.11	0.76	1.32	-6.51
0.03	0.95	1.44	1.12	1.67	-7.44
0.03	0.99	1.94	1.52	2.45	-8.72
0.05	0.7	0.51	0.38	0.67	-5.17
0.05	0.8	0.49	0.38	0.72	-5.46
0.05	0.9	0.84	0.5	1.27	-6.5
0.05	0.95	1.21	0.76	1.48	-7.56
0.05	0.99	0.6	0.01	1.56	-9.13
0.1	0.7	0.41	-0.01	0.6	-4.14
0.1	0.8	0.49	0.23	0.95	-4.28
0.1	0.9	0.47	0.13	1.23	-5.83
0.1	0.95	-0.47	-1.56	0.33	-7.41
0.1	0.99	1.93	-0.61	5.55	-7.95

Lost Sales					
		DP			Simple
Σ_0	$\varkappa$	Average	Best	Worst	Average
0.03	0.7	0.55	0.28	0.89	-3.4
0.03	0.8	0.51	0.23	1.06	-3.6
0.03	0.9	1.26	1.01	1.55	-4.13
0.03	0.95	1.52	1.22	1.8	-5.25
0.03	0.99	2.03	1.68	2.75	-6.53
0.05	0.7	-0.02	-0.39	0.22	-2.55
0.05	0.8	0.1	-0.23	0.57	-2.85
0.05	0.9	0.41	0.01	0.8	-3.88
0.05	0.95	0.8	0.39	1.33	-5.31
0.05	0.99	0.09	-1.17	1.94	-6.85
0.1	0.7	-1.02	-1.48	-0.23	-0.89
0.1	0.8	-0.82	-1.5	0.16	-1.32
0.1	0.9	-1.08	-2.0	0.35	-3.18
0.1	0.95	-2.87	-4.59	-1.14	-5.53
0.1	0.99	-3.81	-7.4	5.74	-8.9

Table 2.6: Summary of the Updating Forecasts experiments (Section 2.5.3). Each row summarizes the ten agents that were trained with the environment parameters in the two leftmost columns. The four rightmost columns give the average, best, or worst of the ten agents' median gaps (in %) with respect to either the DP method or the Simple heuristic (see the headers).

2.5.4 Lost sales

The high shortage costs in the Lost Sales column of Table 2.3 lead to DP policies with shortage rates of less than 1%. In this section, we specifically focus on lost sales environments with parameters that allow for a higher shortage rate at optimality. Lowering the shortage cost is, however, not sufficient as this quickly leads to instances where the optimal policy is to not participate (i.e. never ordering), due to the high order cost. We avoided this issue by using a Poisson demand distribution instead of a Gaussian. The higher uncertainty of the demand induced by the long tail of the Poisson distribution creates more frequent shortages and thus penalizes more a non-participation. In these experiments we varied the parameters in a full factorial fashion as follows: $L \in \{2, 4, 8\}, b \in \{5, 10\}, K \in \{10, 20, 30\}$. In these settings, the shortage rates of the corresponding DP policies ranged from 2% to 9%. The results are plotted in Figure 2.7 (see Appendix 2.A) and summarized in Table 2.7. The results shows that the DRL approach provides competitive policies as the lead time increases, albeit with a higher variance as can be seen with the height of the boxes. Longer lead times are harder to learn in general, due to the delay between an action and the resulting reward, and thus the agents do not all converge to similarly performing policies. Nevertheless, the deep RL approach is able to outperform both the DP and the Simple heuristic in some settings, by a large gap in the best cases.

			DP			Simple
K	L	b	Average	Best	Worst	Average
10	2	5.0	3.26	-1.56	5.73	-4.32
20	2	5.0	0.71	-0.93	4.48	-11.51
30	2	5.0	1.35	0.4	2.86	-11.55
10	2	10.0	4.36	2.65	6.03	1.06
20	2	10.0	-0.11	-0.53	0.25	-7.28
30	2	10.0	2.21	1.72	2.84	-5.64
10	4	5.0	-5.31	-8.3	3.4	-10.46
20	4	5.0	-3.3	-4.55	-2.13	-12.77
30	4	5.0	-3.63	-4.48	-1.97	-13.35
10	4	10.0	0.21	-3.98	3.86	-0.74
20	4	10.0	-1.95	-2.92	-1.06	-7.08
30	4	10.0	-0.05	-1.68	1.61	-6.28
10	8	5.0	-10.0	-16.54	-3.63	-12.81
20	8	5.0	-6.82	-10.9	1.18	-13.93
30	8	5.0	-6.47	-10.21	-0.43	-14.68
10	8	10.0	-4.77	-12.54	-1.32	-4.06
20	8	10.0	-3.18	-8.88	10.27	-6.31
30	8	10.0	-2.9	-7.89	3.85	-7.66

Table 2.7: Summary of the Lost Sales experiments (Section 2.5.4). Each row summarizes the ten agents that were trained with the environment parameters in the three leftmost columns. The four rightmost following columns give the average, best, or worst of the ten agents' median gaps (in %) with respect to either the DP method or the Simple heuristic (see the headers). The three rightmost columns give the count of agents with a median gap within selected intervals.

2.5.5 Visualization of the learned policies

Figure 2.8a (see Appendix 2.A) compares the DP policy and the output of a learned DRL policy with respect to the inventory position (y). For the visualization, we selected the best performing agent (1.00% median gap) for the instance with backorders, $K = 360, b = 10, CoV = 0.2$, $L = 4$, and the constant 10 forecast. The on-order quantities in the state are set to zero, thus the inventory position consists solely of the inventory level. The red lines refer to the DP method and the blue lines to the agent. The solid lines are the order quantities prescribed by the respective policies (given their (s, S) output), the dashed and dash-dotted lines are the s and S parameters respectively (both constant for the DP method).

Unlike the DP baseline, the s output of the agent decreases as a function of y. However, in effect, there is still an implicit unique ordering threshold at the point where the dashed blue line crosses the identity line (dotted black line). This behavior arises from the fact that, during training, s is sampled from the Gaussian distribution of the agent's stochastic policy. Therefore, a very low mean when the inventory position is high ensures that the chances of sampling an s that would trigger an order are low. Conversely, a higher mean for s when the position is low decreases the chances of not ordering when it is needed.

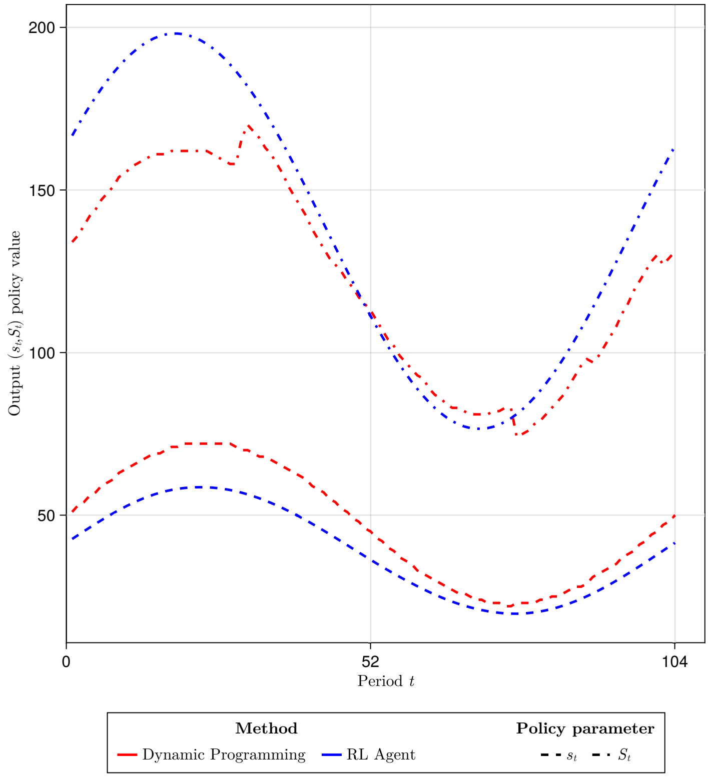


Figure 2.3: Evolution over time periods of the (s, S) predicted by an agent (blue) and the baseline DP policy (red).

Although the S prediction is slightly increasing with the inventory position, and therefore not constant with respect to y, the agent learned a policy with a reorder point close to a true (s, S) policy.

Figure 2.8b provides the same comparison of the DP baseline but with one of the agents of the Lost Sales experiments in Section 2.5.4 (-4.47% median gap). The environment parameters are $K = 30, L = 4, b = 5$, and the trend is the constant 10. The reorder threshold of the DRL policy is significantly lower than that of the DP baseline. This is expected in the case of lost sales because shortages are less expensive in this setting due to only being penalized once, whereas backorders accumulate for possibly several periods until resolved by a delivery. Since the DP baseline solves for a backorder policy, it is unsurprising that it over-conservatively holds more safety stocks than an agent trained in a lost sales environment.

Figure 2.3 compares how the baseline DP policy and the prediction of the agent used for Figure 2.8a evolve as the forecast window evolve over time under the seasonal 1 trend. The input state of the actor uses an inventory level z_t equal to the point where it outputs $s_t = z_t$. This reorder threshold is obtained with a line search on z_t (there are no on-order quantities). Evidently, the approximated policies vary as time progresses and the predicted (s_t, S_t) parameters follow a trend that resembles the demand trend, similarly to the DP baseline. The DRL policy is smoother than the DP baseline; this is due to the discretization of the DP algorithm where small errors accumulate during the backward computation of the policy and lead to a sudden jump at some iterations.

Finally, Figures 2.9a and 2.9b (see Appendix 2.A) show the heatmap of the gradients of the (s, S) outputs (top and bottom heatmaps respectively) of the same policy neural networks as in Figure 2.8, with respect to the elements in the forecast window. Each row of the heatmaps correspond to a forecast trend. The gradients are not significantly affected by the input trend. We can observe that the network of Figure 2.9a only uses the $L + 1$ first periods to decide the reorder threshold s, and around 7 more periods to decide S. In Figure 2.9b, period $L + 1$ is almost exclusively used by the neural network to define the reorder threshold.

2.6 Conclusion

In this chapter we presented a methodology that allows to extend the use of DRL approaches to IC problems that feature a non-stationary demand and a rolling forecast horizon. This methodology is based on the PPO algorithm and allows to approximate the optimal policies of versions of the Single-Item Stochastic Lot-Sizing Problem with non-stationary demand forecasts (i) by augmenting the state encoding with the forecast of the upcoming demand in a fixed-length window and (ii) by training the agents in environments with randomized demand forecasts. This strategy allowed the reuse of the trained neural networks without the need to retrain in subsequent periods. To the best of our knowledge, this is the first method that addresses non-stationary IC problems with a DRL algorithm. The proposed methodology also does not require forecast data to train agents. Specifically, because the forecast is embedded in the state encoding and only known at test time, this approach approximates a generalized replenishment policy given any forecast in the support of the environment initial state distribution.

The three series of experiments carried out in Section 2.5 showed that the agents can learn close-to-optimal policies on backorder and easy lost sales instances and that they can additionally learn in environments where the forecasts update periodically according to a MMFE. Moreoveor, the experiments also showed that the agents can learn policies that outperform a DP algorithm in the case of lost sales.

We also note that this work is also the first application of DRL to IC that uses a continuous policy. This feature paves the way to future research efforts aiming at effectively scaling the approach to large multi-item inventory problems with stochastic demand (Boute et al., 2021).

2.7 Future research

While deep reinforcement learning (DRL) theoretically offers the ability to learn complex inventory-control policies, its application remains confined to scientific studies, as several practical caveats have hindered its broader adoption.

Notably, DRL offers no guarantees on performance or convergence stability, necessitating further research to address this limitation. Ensembling multiple trained models—a common technique in other machine-learning domains—could enhance robustness in DRL. We briefly explored this possibility when considering a different candidate algorithm, the Deep Deterministic Policy Gradient (DDPG) (Lillicrap et al., 2016), to study the SISLSP. The critic network in DDPG learns the Q-function instead of the V-function. This enables us to let several independently trained agents to "vote" with their critic to select the best action among those collectively proposed by the actors. This ensembling yielded decisions that often outperformed the single best agent. Although we ultimately adopted the standard PPO algorithm, this finding suggests a promising avenue for future work. Additionally, pre-training agents to imitate simple heuristics (e.g., De Munck et al. 2025) or integrating reward-shaping techniques (e.g., De Moor et al. 2022) may further accelerate and stabilize learning.

Current practitioners instead favor Mixed-Integer Programming and well-established heuristics. The former, grounded in the mathematical-programming framework, readily models a wide range of problems; by contrast, DRL demands a simulation wrapped in a bespoke MDP interface—an often time-consuming, error-prone coding effort. Enabling the construction of MDP environments through a modeling language akin to those used in mathematical programming could substantially boost DRL's adoption in operational research. Such a tool might draw inspiration from existing stochastic dynamic-programming languages, for example SDDP.jl (Dowson and Kapelevich, 2021).

Finally, to our knowledge, robust optimization remains unexplored in DRL for operational research. Modeling risk aversion via Distributional Reinforcement Learning could enable the minimization of conditional value-at-risk or entropic risk (Bellemare et al., 2023), thereby bolstering practitioners' confidence in the resulting policies as well as expanding the range of adressable problems.

Acknowledgments

This research was carried out under a research mandate provided by the Université de Louvain. The second author acknowledges support from the Fondation Louvain via the research grant COALESCENS and the Belgian National Fund for Scientific Research (FRS-FNRS) via the grant PDR 40007831. Computational resources have been provided by the supercomputing facilities of the Université de Louvain (CISM/UCLouvain) and the *Consortium des Équipements de Calcul Intensif en Fédération Wallonie Bruxelles* (CÉCI) funded by the *Fond de la Recherche Scientifique de Belgique (F.R.S.-FNRS)* under convention 2.5020.11 and by the Walloon Region.

2.A Appendix: Figures

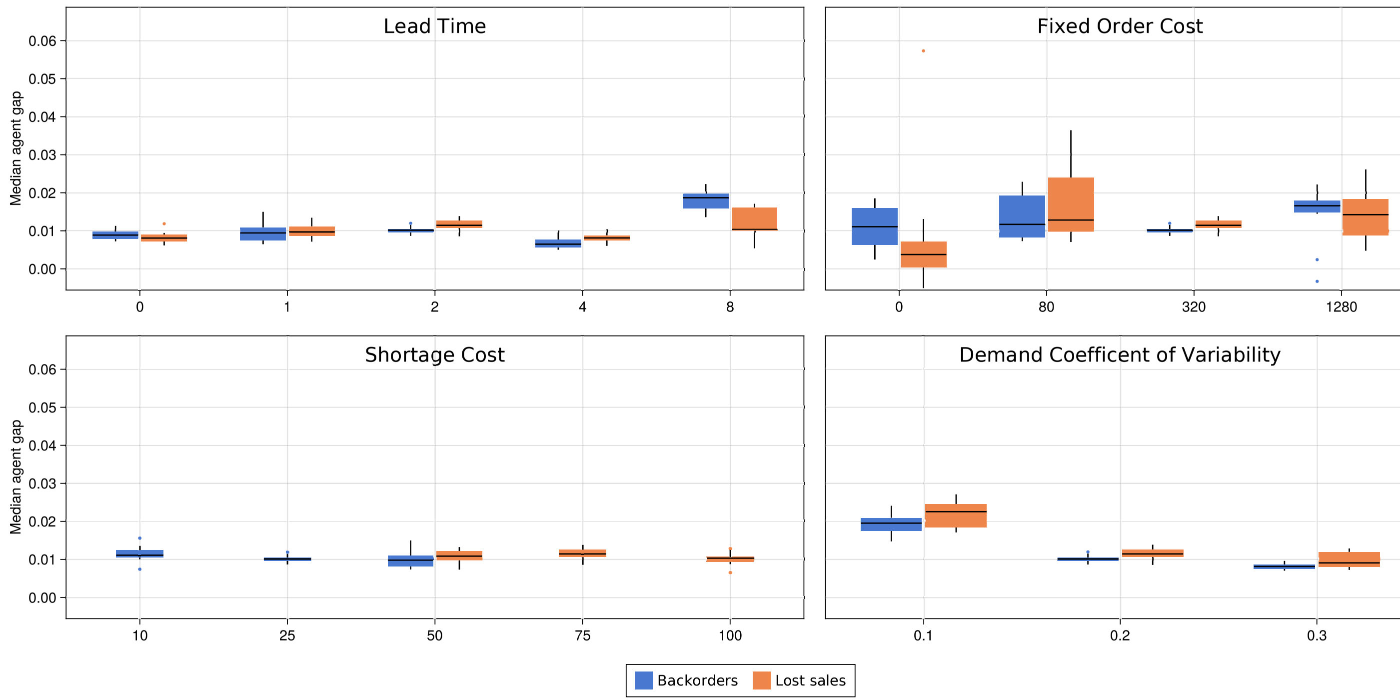


Figure 2.4: Whisker plots of the median agent gaps with respect to the DP baseline for the near-optimality experiments. Each data point represents one agent, each whisker plot thus represents ten agents.

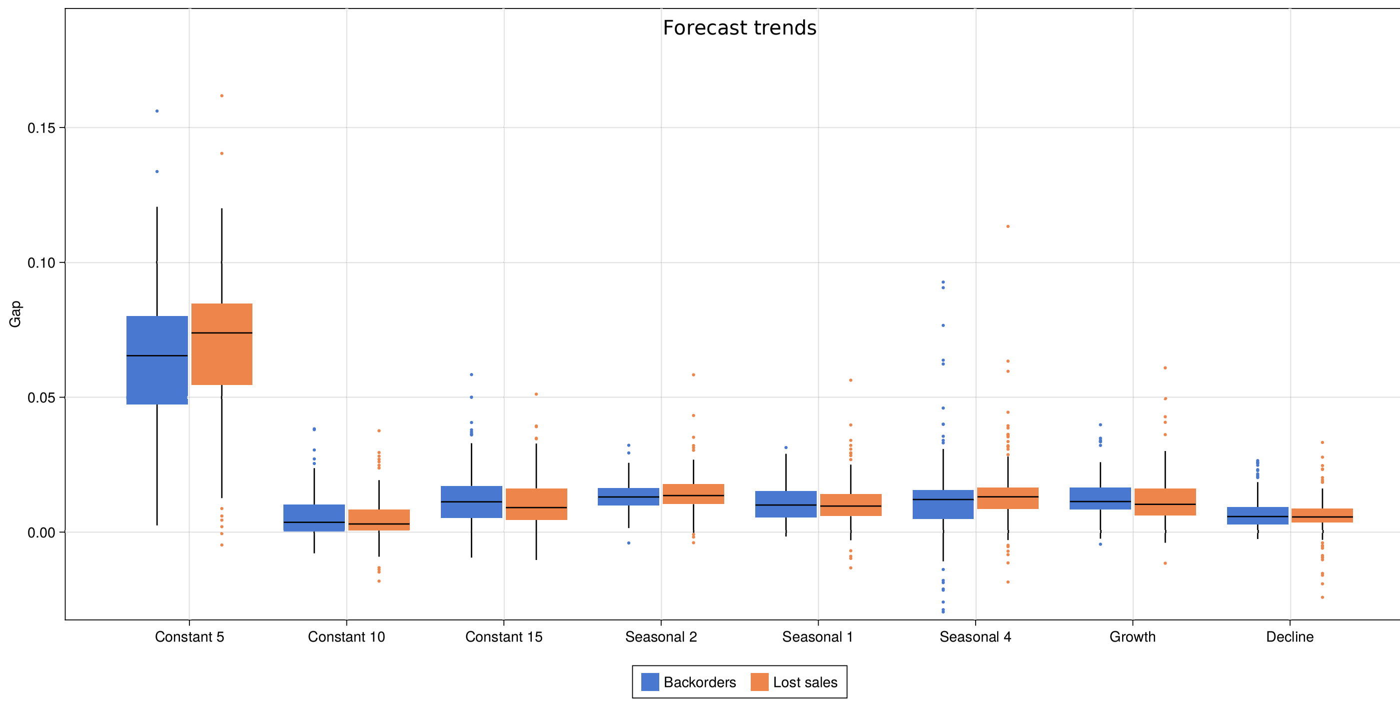


Figure 2.5: Whisker plots of the gap of the agents for each forecast with respect to the DP baseline. Each data point represents one test instance, each whisker plot thus represents the performance one the respective forecast trends of each of the 120 agents (backorders or lost sales).

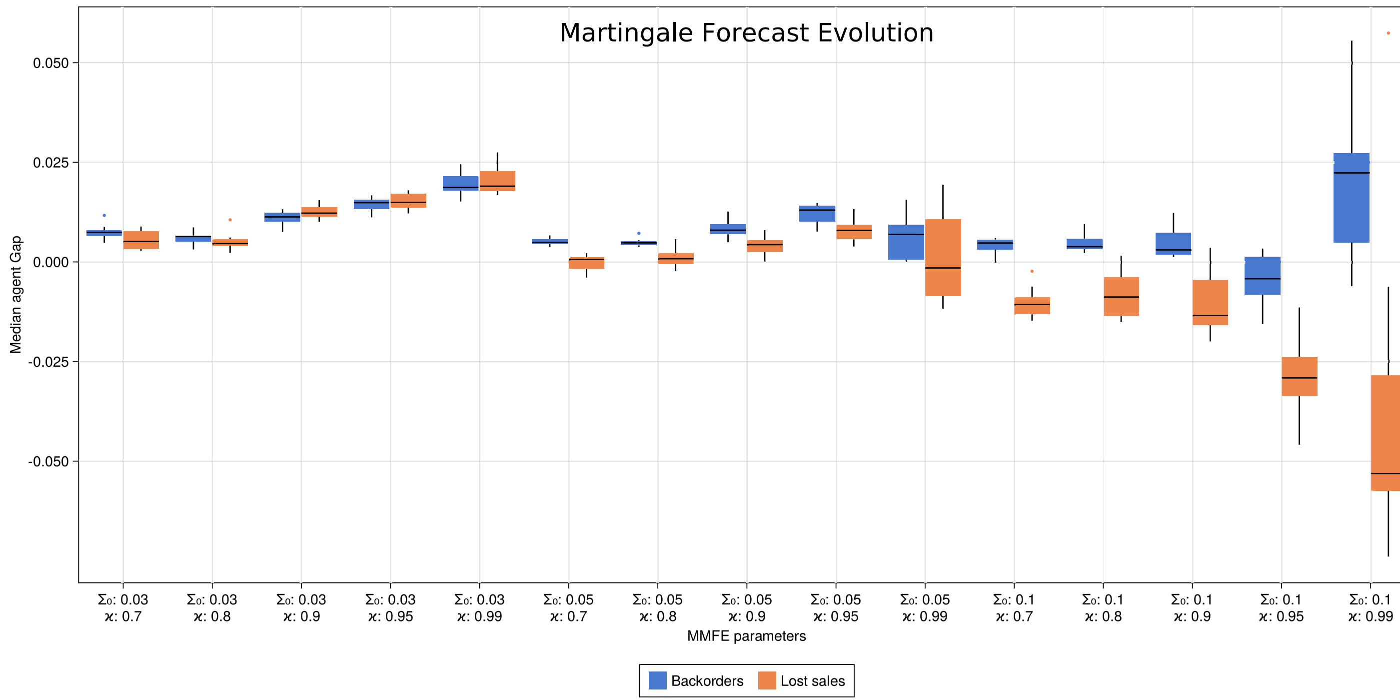


Figure 2.6: Whisker plots of the median agent gaps for the Updating Forecasts experiments with varying MMFE parameters. The gaps are computed with respect to the average cost of the dynamic programming (s_t, S_t) policy of each instance, recomputed at each step to account for the updated forecast. Each data point represents one agent, each whisker plot thus represents ten agents.

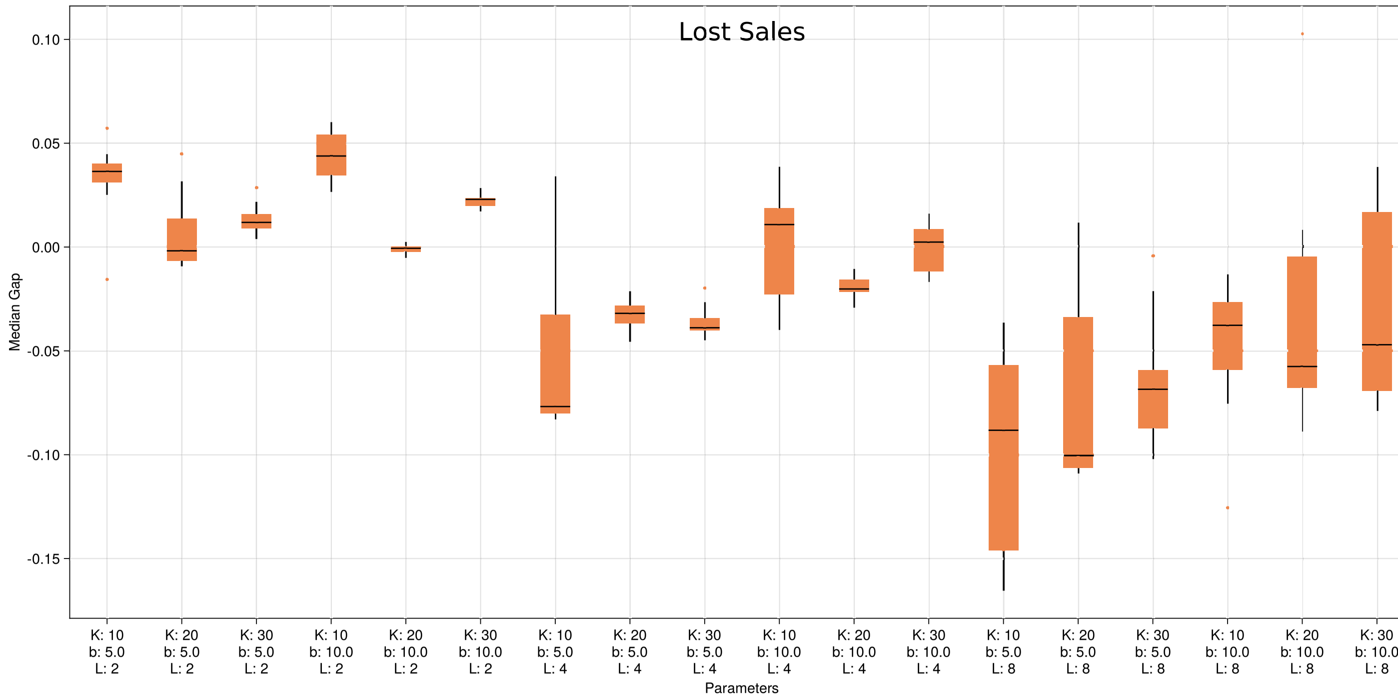


Figure 2.7: Whisker plots of the median gap for the Lost Sales experiments for varying environment parameters. The gaps are computed with respect to the average cost of the DP policy of each instance. Each data point represents one instance, each whisker plot thus represents ten instances.

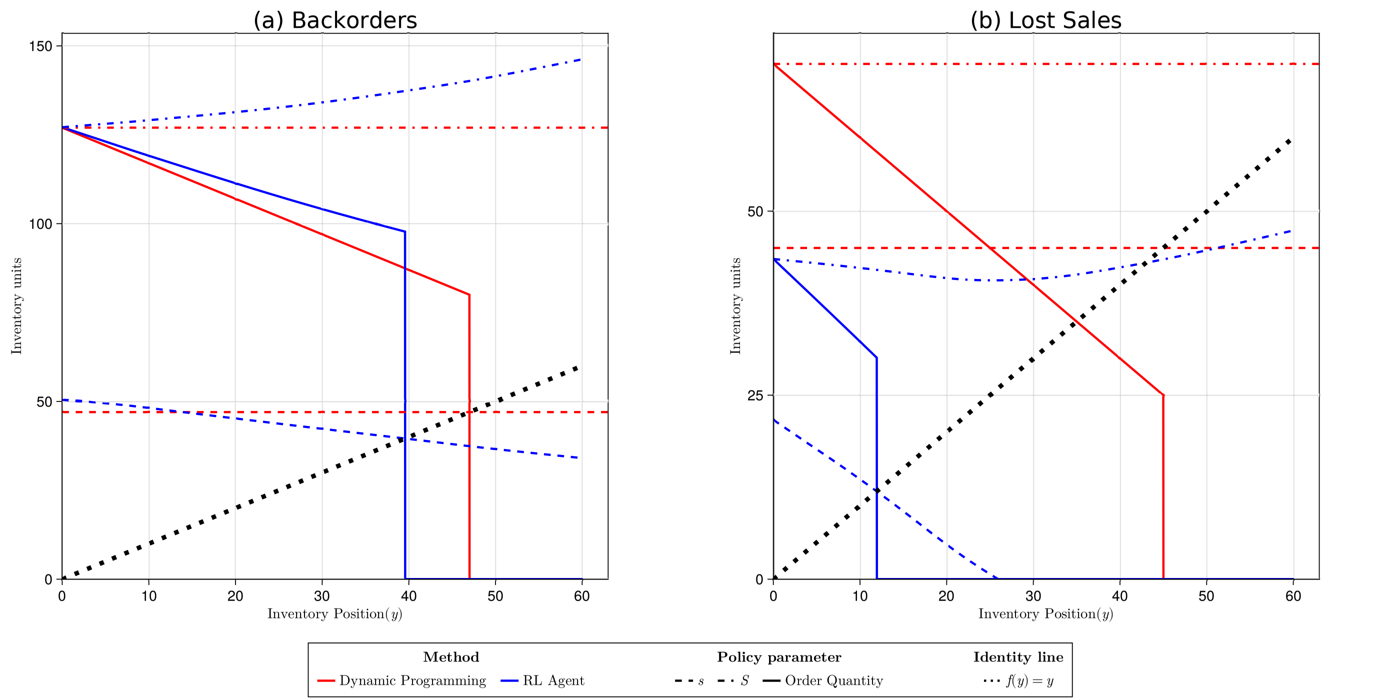


Figure 2.8: Comparison of the DP (s, S) policies (red) and learned deep RL policies (blue) in a backorders environment (left) and a lost sales environment (right). The dashed and dash-dotted lines are the predicted (s, S) parameters respectively, and the solid lines are the resulting order quantities with respect to the inventory position.

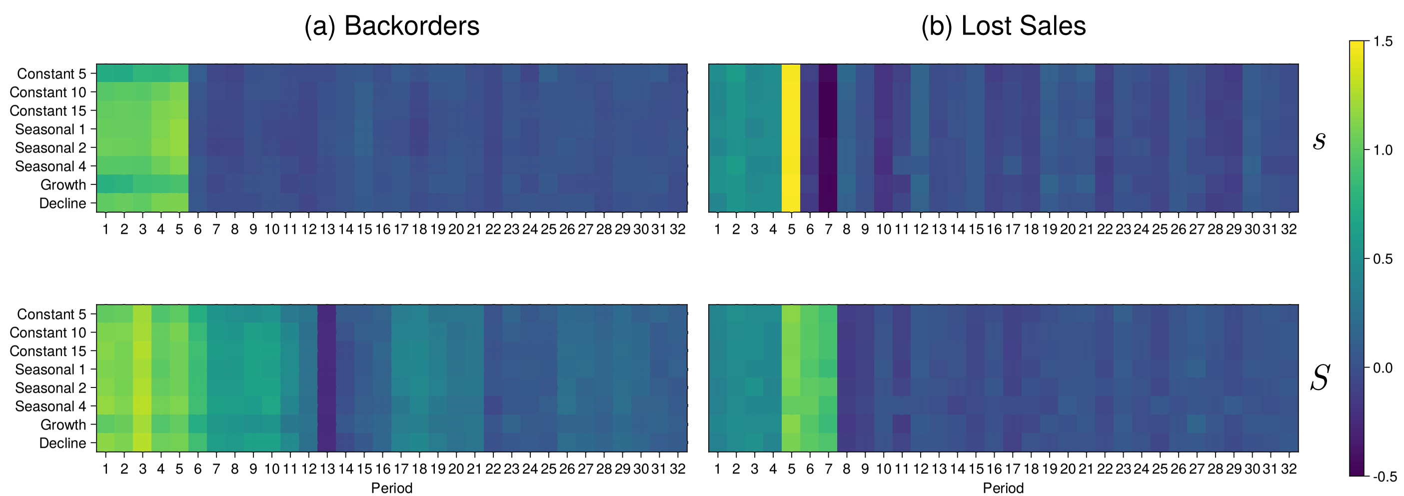


Figure 2.9: Heatmaps of the gradients of s (top) and S (bottom) with respect to the forecast input for each forecast trend. The agents inspected are those used in Figure 2.8a (left) and Figure 2.8b (right), trained in backorders and lost sales environments respectively.

Chapter

THREE

BALANCING SUPPLY SECURITY AND DECARBONIZATION: OPTIMIZING GERMANY'S LNG PORT INFRASTRUCTURE UNDER THE EUROPEAN GREEN DEAL

Abstract

The European Green Deal depends on a coordinated common policy for demand, renewable generation, supply security, and infrastructure. To ensure supply security after 2022 gas supply interruptions, the German government diversified energy imports into liquefied natural gas (LNG) through floating storage and regasification units (FSRU). We test whether the current plan for Germany is consistent with the Green Deal objectives for decarbonization. We use a cost-minimizing facility location model following European policy objectives, combined with a network flow model, to optimize the assignment of FSRUs among German ports to ensure gas supply. Six scenarios are investigated: three greenfield models, with different demand scenarios, in which all locations are possible, a brownfield model in which the existing terminals are imposed, a no-FSRU scenario, and a peace scenario between Ukraine and Russia. The optimal solutions for the base greenfield scenarios use the FSRU port infrastructures to expand the regasification capacity to 45 bcm per year. The model validates actual current decisions as optimal for supply security, but the two greenfield and long-term decisions differ in length and investment intensity, challenging policy consistency. Potentially, the impact of anticipatory hydrogen transport investments can distort the analysis. The European Union (EU) is embarking on an ambitious climate change response policy through the Green Deal (European Commission, 2019), aiming at a 55% reduction in

CO_2 emissions by 2030 and complete carbon neutrality by 2050. The plan concerns regulation, financing, access, and market conditions for a new revitalized energy sector, tightly monitored through the new Energy Union. Germany is a key player in the plan, being the largest emitter of CO_2 in Europe in absolute terms and above average per capita due to its industrial structure and dependence on fossil fuels for power generation, especially natural gas.

Natural gas has played an important role as a transition energy for heating and peak load management in Europe and elsewhere. Although combustion of this gas produces CO_2 emissions lower than those of coal or petroleum products, the convenience of using natural gas as a transition fuel to soften the temporary economic impact provoked by the adoption of renewable energy on a full scale has been widely debated. Gürsan and de Gooyert (2021) review the literature on the role of natural gas in reducing carbon emissions, acting as a bridge to renewable technologies. One of the main arguments against using natural gas as a transition energy source is that new investments in natural gas could displace future investments in renewable alternatives. In this context, the object of this work is pertinent, provided the versatility, lower unit investment costs, and mobility of the FSRUs, compared to onshore regasification plants.

Natural gas is delivered by long-distance maritime or land transport. On land, natural gas transportation mobilizes a massive pipeline network divided into transmission, regional transmission, and distribution systems with associated storage and compressor stations. By sea, transport is carried out in three steps. First, natural gas is liquefied in dedicated facilities (liquefaction plants) located in loading ports, obtaining LNG that is shipped at -162°C in LNG carriers, double-hulled ships with insulated tanks. At the destination ports, it is loaded into insulated storage tanks to be later regasified and injected into the pipeline distribution network. The regasification process is carried out in dedicated plants that can be of two types: onshore plants or FSRUs. The technology is the same in both types, the only difference being that in FSRUs all equipment is adapted to marine operations.

Due to its strong dependency on imports and the decrease in production volumes, Europe faces serious challenges in maintaining a balanced energy mix, aggravated by the geopolitical risks associated with the natural gas supply. The 2022 war in Ukraine changed the scenario in which Europe had evolved in the last few years, based mainly on gas imports via pipeline from Russian producers. The cuts in Russian gas supply to the EU, and especially to Germany, triggered a fossil fuels crisis of global proportions with a particular incidence in Europe, resulting in high volatility of the price of oil and gas in the international market (see Figure 3.1).

To date, the diversification of gas imports remains one of the greatest challenges facing Europe. European imports depend on a gas oligopoly with very few actors: pipeline gas from Norway, Algeria, and Russia, and LNG from the USA, Qatar, and Nigeria (Jaller-Makarewicz, 2024). Since the Russian attack

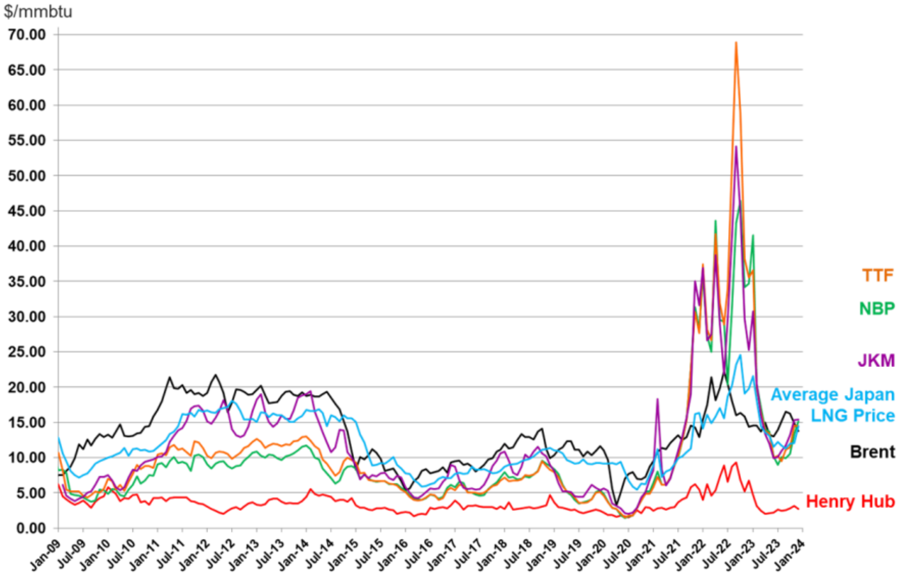


Figure 3.1: Global gas and Brent prices in US Dollars per million British thermal units: Jan 2009 – Dec 2023. (Heather, 2024)

on Ukraine, the EU has strongly reduced its gas imports from Russia (Figure 3.2), compensating for this reduction with demand reductions and increased LNG imports, especially from the United States.

3.0.1 Germany in the European Energy Union

Germany is by size, industrial structure, and electricity generation pool, the most important country in Europe in terms of CO2 emissions, since the country emits 24% of the total EU emissions derived from fuel combustion (IEA, 2023). It is also the third highest per capita emitter (7,501 tCO2 per capita), even though considerable efforts have been made to reduce this figure (-38% since 2000). Fossil fuel consumption has been equally reduced, particularly coal (-48% between 2000 and 2020) and oil (-29% between 2000 and 2022). In any case, natural gas remained a constant in the German energy mix until 2022. The supply shock triggered by the war resulted in a dual response: natural gas consumption declined, and a strategy was devised to expeditiously construct gas infrastructures, particularly LNG regasification plants. Following this strategy, Germany has already built three LNG terminals in record time at the Wilhelmshaven, Brunsbüttel, and Lubmin ports. Two additional facilities will be in operation at the ports of Mukran and Stade in 2024 and 2025, respectively. All of them operate FSRUs to supply the German gas distribution network. In parallel, plans have been launched to build onshore regasification

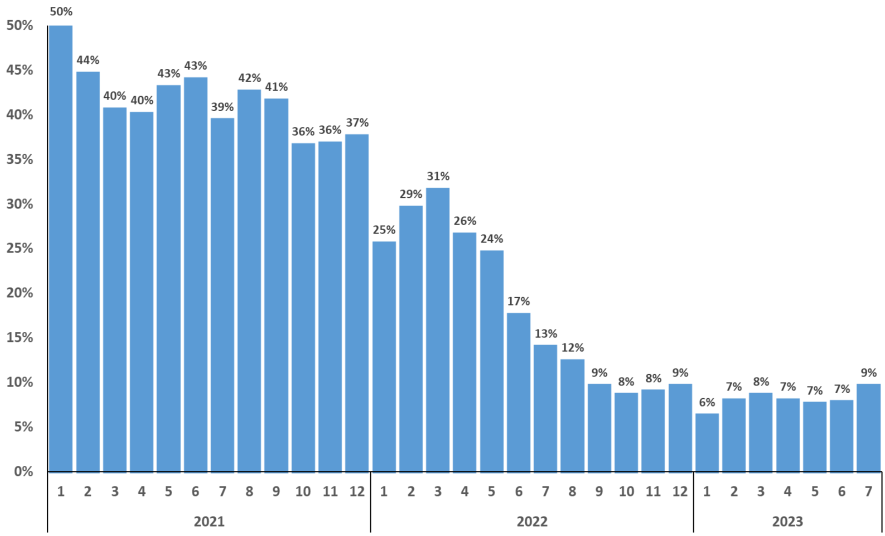


Figure 3.2: Share of Russian pipeline gas in total EU natural gas imports. Source: (European Commission, 2023)

plants, some of them in an advanced stage. The German government's master plan (European Commission, 2023) is to use natural gas as energy to transition to a green economy based on renewable energies. Thus, some of these gas facilities would eventually be retrofitted, enabling import and production of green hydrogen in the future. Given its decentralized gas transmission infrastructure with 12 independent (private) gas transmission system operators (TSO) (ENTSOG, 2024) with separated accounts, the economic energy system planning in Germany is ultimately dependent on the federal regulatory authority, Bundesnetzagentur (BNetzA) in collaboration with the EU Agency for the Cooperation of Energy Regulators (ACER). The choices made to meet legitimate short-term needs for supply security in terms of natural gas can therefore be seen as indications of the overall policy direction for Germany and Europe towards the Green Deal objectives.

3.1 Objective

The consistency of the policy of natural gas supply decisions in Germany has been the subject of critical discussion, based on various projections of demand, pricing, and competitive factors. However, the urgency to maintain the natural gas supply in the short term to ensure economic and social conditions is an obvious imperative for any government planner. Moreover, the complexity of the gas transmission system network, the demand locations, and the sequence of potential investments in port-side, maritime, or land-based infrastructure call for a deeper techno-economic analysis before passing to conclusions. However, the evaluation of the consistency of the claimed, potential and actual policy is often left to an arbitrary assessment based on hypothetical arguments and assumptions, rather than an objective framework.

The objective of this work is therefore to assess the policy consistency of the German short-term and medium-term natural gas supply investments in LNG import facilities, more specifically in FSRU port infrastructures, using operations research to model the optimal decision-making process under the stated constraints and comparing the model solutions to the actual policy. We investigate the deployment of a set of FSRU-operating LNG terminals located in 11 German candidate ports. These terminals can unload one or several ships depending on the infrastructure work carried out in each of them. The time frame of the deployment comprises 28 years, that is, 2023 to 2050. In this period, we are considering that the demand linearly decreases year by year, reaching zero by 2050. In other words, we assume a steady transition to carbon neutrality by 2050 consistent with the stated European policy objectives in the Green Deal and associated agreements, cf. (European Commission, 2019). Two assumptions have been made: (i) a linear demand decrease and (ii) no seasonality. Although it is unlikely that the exact demand development will follow a linear curve, the objective of this paper is still adherence to the final policy objective of the Green Deal for 2050. The exact phase-out pattern is unknown and any assumptions about the functional form for it would be as arbitrary as the simple linear assumption. Since the horizon involves no specific temporal events, the linear phase-out can be seen as an unbiased average from a series of stochastic outcomes converging to the final known point. Likewise, seasonality is managed with a cycle of storage injections and withdrawals each heating year, as a function of exogenous factors such as weather and business cycle. We address this limitation by modeling only the expected net consumption by (heating) year under the assumption that in-period storage effects net out in the time period. Seen in aggregate over a 28-year horizon, this assumption is likely relatively harmless. Finally, we recognize that climate changes that would significantly change heating demand in the area could have an impact on the overall slope and, ultimately, on the feasibility of the Green Deal economically and technically. However, since such large unknowns may also have an impact on LNG supply and pricing, natural gas pricing, and demand globally and regionally, it is beyond the scope of this paper.

Our contributions are twofold: First, we inform the energy policy discussion for collective European versus national (German) supply security objectives based on an ex-post investment review for recent LNG import choices. Second, we formulate and solve a gas infrastructure location model with a directional flow network to optimize a dynamic investment policy in FSRU and LNG terminals to meet demand objectives. The latter has to our knowledge not been presented in the literature, and the current application to Germany is without loss of generality; the model can be used to inform an individual operator, another country, or higher level planning, e.g. the European level.

The remainder of this paper is organized as follows: the relevant literature is reviewed in Section 3.2. The model is described in Section 3.3 and applied in Section 3.4; the obtained results are analyzed in Section 3.5. Section 3.6 compares our findings with the actual German planning, and finally Section 3.7 is dedicated to the final policy conclusions and possible further research.

3.2 Relevant literature

Among the existent literature about the European energy infrastructure and policy, and more particularly, the natural gas industry, three streams of research are relevant to this study: studies related to the consequences of the invasion of Ukraine, those related to the European energy policy, and literature dealing with optimization models. The energy shock caused by the Ukraine war in Central and Eastern Europe is analyzed by Gritz and Wolff (2024) studying the gas flow into the region, the consumption, and the diversification policies adopted by the European countries to mitigate the impact of the Russian gas cuts. Halser and Paraschiv (2022) discuss possible demand corrections and regulatory measures to cope with the post-Ukraine invasion energy crisis, highlighting the use of FSRUs to shelter the supply of gas and finally concluding that delaying the phase-out of coal and nuclear power could avoid the foreseeable German economic contraction. Kędzierski (2023) examines equally the post-invasion changes in the German gas sector and the adopted emergency measures, among others the construction of port LNG terminals and the control of part of the German gas infrastructure. Overland (2022) concludes that the post-invasion increase of fossil fuel prices will encourage global renewable energy investments, favor the perspectives of the nuclear industry, and reduce the demand for fossil fuels, whereas in Europe the high prices scenario will harm energy-intensive industries. Jirušek (2022) shows the EU's difficulties dealing with the Russian energy import restrictions and the post-invasion geopolitical scenario, highlighting problems for the adoption of a common policy, and indicating that even so, the EU has room to improve a policy through specific regulations for pipeline interconnections with third-party countries and promotion of international cooperation in infrastructure-building. Considering the available infrastructure, the long-term contracts, and the post-Ukraine invasion geopolitical situation in Europe, Chyong and Henderson (2024) propose different future Russian gas export scenarios to Europe, modeling a base case and comparing it with four scenarios with a wide range of flows for the years 2023 to 2030, concluding that although the total resumption of the Russian gas exports to Europe is highly unlikely, depending on the outcome of the war, its partial resumption cannot be ruled out. Analyzing European energy policy responses to the war and its implications for sustainable transformations

Kuzemko et al. (2022) study what those policies mean for environmental sustainability, energy equity, and social justice, revealing the strong potential for the implantation of clean energy supply, the difficulties for fossil fuel phase-out, the negative perspectives for sustainable transitions in the Global South, the implications for energy equity within Europe, and a partial return of the states as energy actors. Linking the invasion of Ukraine to the European energy policies, Emiliozzi et al. (2023) analyze the post-invasion gas markets and their incidence on the global economic landscape: macroeconomic deterioration, reduced growth prospects, inflationary pressures, severe strain on global value chains and increasing geo-economic fragmentation. Regarding Europe, they focus on implemented fiscal support to shield vulnerable households from high energy prices, ensure business continuity, and secure substitute energy sources. They conclude that Europe must adopt a clean-energy paradigm, not only increasing renewable energy investments but also pursuing a long-term strategy to avoid the risks derived from the potential scarcity of critical minerals necessary for the transition.

Examining the EU energy policy and specifically the impact of the National Energy and Climate Plans (NECP19) (European Parliament, 2018) and REPowerEU (European Commission, 2022), Ah-Voun et al. (2024) conclude that, by the application of the latter, Europe could be independent of Russian gas by 2030, indicating that since the execution of REPowerEU is based on the implementation of clean technologies, a substantial effort in the deployment of wind and solar installations in Europe is required. The uncertainties in the European natural gas market about demand and supply are analyzed by Riepin et al. (2022) who propose an adaptive robust optimization framework for the expansion planning of gas infrastructure according to these uncertainties. Using a mixed-integer linear model, they analyze the interest of strategic infrastructure projects supported by EU funds, the so-called "Projects of Common Interest", to warrant supply security by 2030, finding that up to date the investment preferences have been oriented to the most favorable projects from a business perspective. They also find that most projects are difficult to execute without public financial support. Regarding the role that hydrogen will play in a future decarbonized Europe, Durakovic et al. (2023), analyze whether blue and green hydrogen are complementary or competitive depending on electrolyzer costs and natural gas prices. Using a multi-horizon stochastic capacity expansion model, they show that early deployment of green hydrogen leads to higher hydrogen costs and CO2 prices in 2030 whereas the blue hydrogen option presents notable benefits in 2030, allowing cheaper hydrogen with less socioeconomic impact. However, the adoption of green hydrogen and the competitive aspects between blue and green hydrogen are short-term, and by 2050 green and blue hydrogen will be complementary. Investigating barriers and constraints in EU policies for natural gas decarbonization Khatiwada et al. (2022) examine the costs of hydrogen production applied to a specific case in Portugal, and find that with the natural gas prices of 2020, there is a significant gap between the cost of green (electrolysis fueled by renewable energy sources) and gray hydrogen (methane electrolysis without carbon capture) and that of natural gas, indicating that the inclusion of a carbon tax would be necessary to incentivize hydrogen production. The energy transition in Germany and the risk of gas lock-in under that transition are treated by Brauers et al. (2021) in connection with the LNG expansion plans of the German government, arguing that the plans for the construction of LNG import terminals are contradictory to the country's climate objectives and promote increases rather than decreases in natural gas consumption.

Optimization models for gas transmission networks focused on short-term storage (line-packing problems), gas quality satisfaction (pooling problems), and compressor station modeling (fuel cost minimization problems) are addressed in the survey in Ríos-Mercado and Borraz-Sánchez (2015). Sesini et al. (2020) use a minimum-cost linear programming model to study the resilience of the EU natural gas network in case of supply disruptions or unexpected peak demand periods by minimizing costs with respect to gas transmission, capacity and storage. The model enables the import of LNG to meet the demand. Using a mixed-integer programming formulation, Melkote and Daskin (2001) presents a capacitated facility location/network design problem as a generalization of the classical facility location problem, concluding that, compared to the models where capacity constraints are imposed, the network is denser and both link and transport costs decrease. Zwickl-Bernhard et al. (2024) study the development of the current Austrian gas grid by 2040 by using a mixed integer optimization model, considering different decarbonization scenarios and analyzing the optimal cost balance between underutilized gas pipelines and external supply alternatives such as trucking. The level of spatial resolution is very high to allow for detailed recommendations. Masudin (2013) uses a Mixed Integer Linear Programming (MILP) formulation to develop a model of the Indonesian gas distribution system, studying the relationship between distribution and facility location costs in that country. The formulation includes transportation and inventory decisions in a joint multi-echelon and multihierarchy location-allocation model that defines the optimal facility locations and customer allocations given a set of existing assets. Using the connections between the different elements of the transmission and distribution networks and introducing six levels for the natural gas supply chain, Hamedi et al. (2009) define a multi-period MILP model whose objective function minimizes total costs *a*ssociated to the distribution and utilization of the nominal and operational capacities of the system. Another facility location approach is used by Derse et al. (2020) to minimize the costs of the production, storage and transportation, safety, location, and staff assignment decisions of green hydrogen, developing a mixed-integer non-linear programming (MINLP) model to integrate all those aspects. Focusing on the resource planning and management problem of a biomass transportation and conversion network covering the entire syngas (synthesis gas) production process, Sarker et al. (2022) develop a

MINLP model minimizing the costs and optimally allocating the different elements of the productive process. The impact of LNG and the Nord Stream 2 pipeline on European gas trade and supply security by 2030 is examined by Eser et al. (2019). They find that increased LNG imports would require part of European gas pipelines to be bidirectional, leading to a substantial need for investments in the European gas infrastructure. Halfway between the three above-mentioned streams this paper aims to answer two research questions not sufficiently addressed as far as we know in the current literature: First, the development of an optimization model combining a facility location and a network flow model including candidate ports, regasification and pipeline investments for a given demand profile. Second, the application of optimization models to validate the strategic investment decisions in countries facing gas supply risks, while still respecting overarching demand reductions in the long term. Even if the body of literature on facility location and distribution models is rich, to our knowledge only one paper uses a similar approach for gas networks, the already mentioned Zwickl-Bernhard et al. (2024) for the Austrian networks but without the FSRU and port dimensions. Secondly, while the European and German energy policies, separately or jointly, are subject to a series of papers, no paper has been found comparing the empirical FSRU investments with the consistency of stated policy objectives using an optimization approach

3.3 Model

We model the dynamic investment decision problem as a multi-period facility location problem combined with a network flow problem. Specifically, the model represents the German pipeline network as a digraph $G = (V, E)$ where each node is associated with a natural gas demand, a supply source, or both. The facility location aspect of the model comes from the possibility of building the necessary infrastructure for berthing one or several FSRUs at a selected candidate *port node*. This investment is referred to as an *upgrade* in the remainder of this chapter.

The sets, parameters, and variables of the model are listed in Tables 3.1, 3.2, and 3.3, respectively. The model is detailed in the following formulation:

$$\begin{aligned} \min \sum_{t \in T} (1-r)^t \Bigg(\sum_{p \in V_p} U_{p,t} \text{capex}_p + \sum_{p \in V_p} A_{p,f,t} \text{opex}_p + \sum_{p \in V_p} U_{p,t} b_p \\ + \sum_{p \in V_p} Y_p^f \hat{p} + \sum_{n \in V_g} Y_{n,t}^d p_{\text{DE},t} + \sum_{c \in C_e} \sum_{n \in V(c)} Y_{n,t}^i p_{c,t} \Bigg) \end{aligned} \tag{3.1}$$

subject to

$$
\begin{aligned}
& Y^d_{n,t} + Y^i_{n,t} + Y^f_{n,t} + \sum_{u \in V(n)} l_{(u,n)} X_{(u,n),t} \\
& \quad = \sum_{u \in V(n)} X_{(n,u),t} + Y^e_{n,t} + d_{n,t}, && \forall t \in T, n \in V, && (3.2) \\
& G_{p,t} = \sum_{k=1}^{t} U_{p,k}, && \forall t \in T, p \in V_p, && (3.3) \\
& X_{e,t} \leq k_e, && \forall t \in T, e \in E, && (3.4) \\
& \sum_{f \in F} A_{p,f,t} \leq q_p G_{p,t}, && \forall t \in T, p \in V_p, && (3.5) \\
& \sum_{p \in V_p} A_{p,f,t} \leq 1, && \forall t \in T, f \in F, && (3.6) \\
& Y^f_{p,t} \leq \sum_{f \in F} A_{p,f,t} w_f, && \forall t \in T, p \in V_p, && (3.7) \\
& \sum_{n \in V(c)} Y^i_{n,t} \leq s_c, && \forall t \in T, c \in C_e, && (3.8) \\
& \sum_{n \in V(c)} Y^e_{n,t} = m_{c,t}, && \forall t \in T, c \in C_i, && (3.9) \\
& U_{p,t} \in \{0,1\}, G_{p,t} \in \{0,1\}, && \forall t \in T, p \in V_p, \\
& A_{p,f,t} \in \{0,1\} && \forall t \in T, f \in F, p \in V_p, \\
& X_{e,t} \geq 0 && \forall t \in T, e \in E, \\
& Y^i_{n,t} \geq 0 && \forall t \in T, c \in C_e, n \in V(c), \\
& Y^e_{n,t} \geq 0 && \forall t \in T, c \in C_i, n \in V(c), \\
& Y^f_{p,t} \geq 0 && \forall t \in T, p \in V_p
\end{aligned}
$$

Equation (3.1) minimizes the discounted sum of capital expenditures, operational expenditures, pipeline construction costs, and gas purchase costs. Constraint (3.2) ensures the conservation between input and output flows in a node. On the left-hand side, the input flows are the imports (if the node is in an exporting country $c \in C_e$), the imports from an FSRU facility (if the node is a port), and the incoming flow from neighboring nodes. On the right-hand side, the output flows are the outgoing flows to neighboring nodes, the exports (if the node is in an importing country $c \in C_i$), and the demand of the node (if the node is in Germany). The incoming flows from neighboring nodes via pipelines are multiplied by a factor $0 < l_e < 1$ that represents the losses due to leakage during transport and the gas used by the network for compressors. In practice, this constraint is generated per node by only including variables that are defined for the respective node. For example, for a node n in a country set $V(c)$ where $c \in C_e \setminus C_i$, the variables $Y^e_{n,t}$, $Y^f_{n,t}$, and the parameter $d_{n,t}$ are not defined and thus not present in the constraint. The same is true for the variable $X_{e,t}$ if $e \notin E$. Constraint (3.3) ensures that $G_{p,t} = 1$ if and only if an infrastructure upgrade was performed in the past. Constraint (3.4) ensures the respect of the capacity of each pipeline. Constraint (3.5) sets the limit on the number of FSRU ships that can be berthed in an upgraded port. Constraint (3.6) ensures that an FSRU can be assigned

Symbol	Label
C	Neighboring countries.
$C_i \subset C$	Countries with import demand.
$C_e \subset C$	Countries with export capacity.
E	Edges (pipelines).
F	Fleet of FSRUs.
T	Time periods.
V	Nodes.
$V(c) \subset V$	Nodes in country c.
$V_d \subset V$	Domestic demand nodes.
$V_g \subset V$	Domestic production nodes.
$V_i \subset V$	Industrial demand nodes.
$V(n) \subset V$	Neighbor nodes of node n.
$V_p \subset V$	Candidate port nodes.

Table 3.1: Description of sets and subsets.

to one port only in each period. Constraint (3.7) sets the processing capacity of the upgraded ports. Constraint (3.8) limits the total import from a country to the supply capacity of that country. Constraint (3.9) enforces the satisfaction of the demand of neighboring countries.

3.4 Application

3.4.1 Data collection

The main data source for this model is the open source IGGIELGN database, which provides data for the European gas transmission network (Diettrich et al., 2021; Pluta et al., 2022). From this dataset, we extracted the sets of arcs (E) and nodes (V) that constitute the German transmission network along with the associated metadata. This dataset provides geographical information on the European pipeline network in the form of pairs of coordinates that indicate the endpoints of each pipeline. Each pipeline is also accompanied by the following parameters relevant to the model: endpoint countries, diameter, length, and maximum flow capacity. It also provides a collection of nodes with their coordinates, country codes, and a NUTS2 regional identifier. We exploit pipelines with at least one endpoint in Germany and only the nodes that correspond to the endpoints of these arcs.

Based on their geographical position and technical characteristics, we identified 11 candidate ports (Bremerhaven, Brunsbüttel, Duisburg, Emden, Hamburg, Lübeck, Lubmin, Mukran, Rostock, Stade and Wilhelmshaven) suitable for the construction of LNG terminals. Then each port was assigned to the closest node with at least one outgoing pipeline to form the subset V_p. Each port can accommodate a single ship, that is, $q_p = 1$, except for Wilhelmshaven, where an extension of the capacity to host a second FSRU (i.e. $q_{\text{Wilhelmshaven}} = 2$) is planned for 2024 (ECOnnect, 2024).

Symbol	**Index sets**	**Label [units]**
b_p	$p \in V_p$	Cost of new pipeline construction if infrastructure investment at p [M€].
capex_p	$p \in V_p$	Net Present Value of capital expenditures to upgrade port p [M€].
$d_{n,t}$	$n \in V, t \in T$	Nodal demand at node n in year t [bcm].
h_n	$n \in V_g$	Domestic production capacity at node n [bcm].
k_e	$e \in E$	Flow capacity in pipeline e [bcm/year].
l_e	$e \in E$	Losses and compression gas usage through pipeline e [%].
$m_{c,t}$	$c \in C_i, t \in T$	Import demand of country c in year t [bcm].
opex_p	$p \in V_p$	Operational expenditures per FSRU in use for a year at port p [M€].
$p_{c,t}$	$c \in C_e, t \in T$	Price of exports from country c in year t [M€/bcm].
$\hat{p}$	-	Price of LNG imported via FSRU [M€/bcm].
q_p	$p \in V_p$	Berthing capacity of port p [# ships].
s_c	$c \in C_e$	Supply capacity of country c [bcm/year].
w_f	$f \in F$	Processing capacity of FSRU f [bcm/year].
r	-	Yearly discount rate.

Table 3.2: Description of parameters.

The investment costs for connection pipelines and port facilities are based on flow volume and connection points to the existing network. The local site characteristics such as slope, soil, land cover, humidity, accessibility for construction, and others, as mentioned in Agrell et al. (2023), are not considered in the cost function since the exact location has not been determined for the pipeline construction and the cost data for all port facilities are not public. The distance between a port's coordinates and that of its assigned node was used to estimate the construction cost of the pipeline (b_p) with 450 mm diameter (Khoiriyah et al., 2023) by multiplying this length by a construction cost of 455 k€/km (Sumicsid, 2019) times the number of ships that can be accommodated at the port (q_p).

We modeled the cost of gas transportation through the pipeline network as an internal gas usage for compressor stations equal to 0.005% of the volume transported per km, as suggested in Molnar (2022). To this, we add the losses that occur during transport due to leakage, estimated to be 0.0425347% of the transported volume (Abrell et al., 2016). This is modelled by the parameter $l_e = 1 - \text{length}_e \cdot 0.00005 - 0.000425347$ that multiplies the pipeline inputs of a node in Constraint (3.2) of the model.

We assume that the investment required to equip each port with FSRU infrastructure, that is, to *upgrade* it, depends on the number of FSRU that can be berthed (q_p) and is estimated according to the investment breakdown in Table 6.1 in Songhurst (2017), from which we removed the vessel investment (leasing fee in our case), and adjusted for 22.8% of inflation since 2017. We thus have the *upgrade investment cost*

$$\mathrm{up_p} = 1.228 \times ((80q_p + 30) \times 1.1 + 54).$$

Symbol	Index sets	Label
$A_{p,f,t}$	$p \in V_p, f \in F, t \in T$	1 if FSRU f is assigned to port p in year t. Used to compute the available capacity at the ports.
$G_{p,t}$	$p \in V_p, t \in T$	1 if port p was upgraded for FSRU berthing during the current or a past year. Used to define whether $A_{p,f,t}$ may be 1.
$U_{p,t}$	$p \in V_p, t \in T$	1 if port p is upgraded for FSRU berthing during the year t. Used to define the year when the capex are paid.
$X_{e,t}$	$e \in E, t \in T$	Flow of gas in pipeline e during year t.
$Y^d_{n,t}$	$n \in V_g, t \in T$	Quantity of gas produced at domestic node n.
$Y^e_{n,t}$	$n \in V_i, t \in T$	Quantity of gas exported via the foreign node n.
$Y^i_{n,t}$	$n \in V_e, t \in T$	Quantity of gas imported via the foreign node n.
$Y^f_{p,t}$	$p \in V_p, t \in T$	Quantity of gas imported via FSRU at port p.

Table 3.3: Description of variables

We model the capital investment cost ($\mathrm{capex_p}$) as the Net Present Value of this investment, paid with ten equal annuities and a discount rate of $r = 0.0175$ (Agrell et al., 2024). This gives the formula

$$\mathrm{capex_p} = \sum_{t=1}^{10} \frac{\mathrm{up_p}}{10 \times (1+r)^{t-1}}.$$

This amounts to a capex of 214.9M€ for a one-FSRU infrastructure, and 322.9M€ for a two-FSRUs one. Annual operating expenses for an FSRU include the annual lease cost of 46M€ (Reuters, 2022), plus 2.5% of the undiscounted upgrade investment cost (Songhurst, 2017), that is,

$$\mathrm{opex_p} = 71.5 + 0.025 \times \mathrm{up_p}/q_p.$$

Note that the investment value is divided by q_p as the opex is paid annually per FSRU *in use*. The final values are 76.9M€ and 75.5M€ per year for $q_p = 1$ and $q_p = 2$, respectively. Without loss of generality, we assumed a fleet of 12 identical FSRU with $w_f = 5$ (LNG Prime, 2024).

3.4.2 Demand estimation

The objective of the model is to represent the gas flow in Germany to explain the decision about the location of LNG port terminals. As the flow is driven by domestic consumption in addition to transit engagements, we assign an estimate of demand to each node in the country. This demand is divided into two categories: residential and industrial load. Industrial and commercial demand, dominated by power plants, accounts for 59 % of the total demand for the country, which was 86.7 bcm in 2022 (Enerdata, 2023). The share of each industrial node $n \in V_i$ from this dataset is estimated proportionally to their power in MW (Pluta et al., 2022).

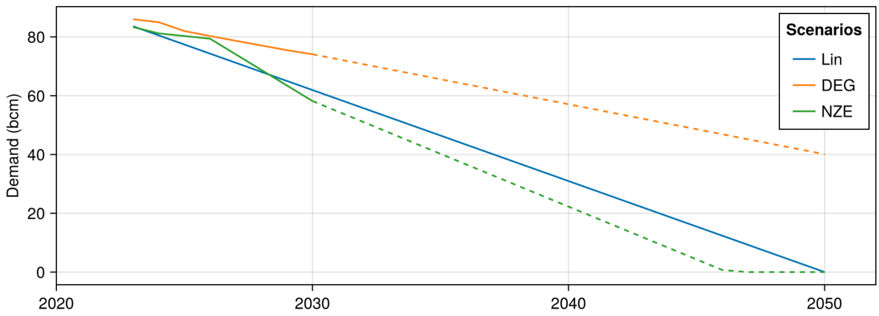


Figure 3.3: Comparison of the demand scenarios from 2023 to 2050. The Model scenario demand decreases linearly. Forecasts from the DE government (DEG) (BMWK, 2023a) and Net Zero Emission (NZE) (David et al., 2023) are limited to 2030. The dashed lines correspond to the linear projections from 2030 to 2050.

Geographic data for the local residential load are not available. Therefore, we base our estimations on the assumption that residential consumption is proportional to the residents' local Gross Domestic Product (GDP) per capita. To do this, we use public Eurostat data on the average GDP per capita aggregated at the geographical level NUTS2 and the population of each NUTS2 area to estimate their respective GDP (Eurostat, 2023a,b). This demand is distributed between all nodes within the NUTS2 region, proportional to the total input capacity of their respective incoming arcs (Pluta et al., 2022).

Finally, since the goal of this paper is to model the supply of gas in the context of the green transition, we analyzed three demand scenarios. First, we considered the *linear* scenario (LIN) where we linearly decreased the national demand ($d_{n,t}$) and the international demand ($m_{c,t}$) starting from the demand observed in 2022 to reach zero in the last period (2050). Second, we linearly projected the gas demand prediction of the German government (BMWK, 2023a) beyond 2030 by assuming a yearly decrease in consumption equal to the average decrease between 2023 and 2030. Specifically, this scenario (DEG) entails a decrease of 1.7 bcm per year to reach a yearly consumption of 40 bcm in 2050. We applied the same extrapolation method to the Net Zero Emissions Scenario (NZE) from David et al. (2023). The decrease is the steepest in this scenario, leading to an end of consumption as soon as 2046. The three scenarios are shown in Figure 3.3. We note that in all scenarios, the decrease in demand from other countries remains linear for simplicity.

3.4.3 Import and export

The current main supply source of natural gas for Germany is imports from neighboring countries through its pipeline network. Germany is also an exporter of gas that transits through its network. We obtained the sets of exporting (C_e) and importing (C_i) countries along with their respective volumes from Bundesnetzagentur (2023).

The base supply capacity from Russia is set to zero. We constructed the corresponding subsets of nodes, $V(c), \forall c \in C_e \cup C_i$, by taking all non-German nodes from the IEGGELGN dataset that are connected by an arc to at least one German node. We assume that, unlike the dynamic demand ($m_{c,t}$), supply capacities ($s_{c,t}$) of the relevant connected countries remain constant over the years and equal to the amount that was actually supplied in 2022. The t index exists in the supply parameter as we consider a change of supply capacity for a set countries in the "Peace" scenario later in section 3.5.

3.4.4 Domestic production

The IGGIELGN dataset provides a set of nodes where natural gas is produced. We the extracted nodes on the German territory to account for domestic production. Each node is accompanied by a production capacity. However, the capacities of the six nodes sum up to 3490 bcm per year according to the dataset. We rescaled the values to a realistic estimate assuming that the production sites were used up to their desired maximum capacity in 2022, to a total of 4.68 bcm (Federal Statistical Office Germany, 2024). We maintained the relative capacities between the production sites identical to that provided by the dataset.

3.4.5 Market prices

We used a constant price ($p_{c,t}$) for each country with a supply capacity, again, the t index exists as we consider a change of price in the "Peace" scenario later in section 3.5. To estimate the price at which each country supplies Germany, we separated them into three categories: those that sell regasified LNG, those that sell domestically produced natural gas, and those that only act as transit countries. Table 3.4 summarizes this classification.

We based the prices of the LNG category on the Total Transfer Facility (TTF) virtual trading point. We used the Henry Hub (HH) natural gas spot price for the Producers category, including Germany's domestic production. To obtain the prices, we used average historical values available from January 2023 to September 2024, obtained from ACER (2023) and EIA (2023), for the TTF and HH indices, respectively. The price of LNG paid by Germany in its ports is equal to the "EU LNG spot price" also provided by ACER (2023), averaged over the same dates. We obtained a price for FSRU imports ($\hat{p}$) of 303 M€/bcm, a TTF price of 325 M€/ bcm, and an HH price of 90 M€/bcm. For the last category, that of transit countries, we averaged the historical prices for the Austrian imports of 397 M€/bcm over the same period, as reported in E-Control (2024), and we assumed an identical price for the other countries in the same category.

Furthermore, we assumed that the real market prices would remain constant over time. As we do not consider alternative sources of energy, this only implies the weak assumption that the supply prices from different countries and LNG are strongly correlated. This is confirmed in IEA (2021), where the authors note a 0.81 correlation between HH and TTF prices, with an increasing trend. However, the respective mean levels of the two prices are different due to the higher number of market actors and

LNG	**Producers**	**Transit**
Belgium, France, Netherlands, Switzerland	Denmark, Germany, Norway, Russia	Austria, Czechia, Poland

Table 3.4: Classification of countries exporting to Germany. The LNG category is that of countries that sell regasified LNG; Producers that sell domestically produced natural gas; and Transit that only act as transit countries.

transactions operating with the international HH price compared to the Continental European TTF price index. Although the higher volatility of the TTF index may temporarily change the arbitrage between the two indexes, the observation that the TTF index is bounded below by the HH index confirms the ordinal preference ranking of the two indices that is underlined in the model. Thus, temporal price changes are not likely to change the qualitative outcome of the model.

3.4.6 Graph construction

The set V of nodes that make up the network is the union of three sets: the sets of nodes with German NUTS2 identifiers ($V_i \cup V_d$) and the set of foreign nodes connected to a German node with an arc. The set E of arcs is the set of all arcs that connect any pair of nodes in V. However, the data provided by the IGGIELGN dataset does not immediately construct a usable graph due to missing pipelines, incorrect flow capacities, and some pipelines being incorrectly labeled as unidirectional. We address these issues as follows. First, missing pipelines create disconnected components in the graph due to some nodes not having any arc linking them to another node, or due to some subset of nodes being linked together but not to the rest of the graph. We identified such nodes with the Floyd-Warshall shortest path algorithm (Floyd, 1962) on an undirected version of the graph. The nodes were then grouped into subsets for which there exists a path from each member to every other member. We identified 10 groups of disconnected nodes for a total of 13 nodes. We manually added pipelines to reconnect each component to the main network to avoid removing important import corridors and industrial consumption nodes. Second, some nodes remain unattainable due to the incorrect directionalities of some pipelines, and therefore have no path from any gas import source on the directed graph. For each unattainable node u that we identified, all of its outgoing arcs were changed to be bidirectional. Finally, the pipeline capacities provided by the IGGIELGN dataset do not allow for a feasible solution. To fix this issue while manipulating the provided data as little as necessary, we relaxed the pipeline capacity by allowing the model to violate constraints (3.4), only for arcs with both endpoints in Germany. This is addressed with a two-step multiobjective lexicographic model, first minimizing the violation of the constraint, then minimizing the cost while ensuring that the violation is kept at most at 120% of the minimal level found in the first step. This ensures the feasibility of the model while minimizing the use of capacity relaxation. The 20% leeway in the capacity violation is there to ensure that the model is not restricted to a single optimal solution found at the first step.

Scenario	Demand	Port upgrades	Import capacities and prices of Russia and Transit countries
Greenfield DEG	DEG	Free	Base
Greenfield NZE	NZE	Free	Base
Greenfield LIN	LIN	Free	Base
Brownfield LIN	LIN	German plan	Base
No-FSRU	LIN	None	Base
RUS-UKR Peace	LIN	Free	Uncapacitated and HH priced in 2027

Table 3.5: Summary of scenario-specific changes to the model and parameters.

The resulting graph network has 674 nodes and 1047 arcs and can be visualized in Figure 3.4 on top of a German map colored by NUTS2 regions. The resulting spatial granularity is similar to that in Zwickl-Bernhard et al. (2024). The arcs highlighted in red are those that required a penalized capacity increase in our base scenario solving.

3.5 Results

In this Section, we analyze the policies obtained in six different scenarios. Three "Greenfield" scenarios, where the model is free to upgrade any candidate port, are analyzed in Section 3.5.1. Each uses one of three demand decrease predictions described in Section 3.4.2. A "Brownfield" scenario, in Section 3.5.2, where the model is forced to upgrade the ports in which Germany has already invested. Additionally, we investigate a "no-FSRU" scenario in Section 3.5.3 where Germany decides not to invest in FSRU port infrastructures; and a "Peace" scenario in Section 3.5.4 where we assume a resumption of trade with Russia in 2027, following a hypothetical end of the war. Finally, we strengthen our findings by presenting the results of a sensitivity analysis in Section 3.5.5. Table 3.5 summarizes the differences between each scenario.

All models converged in less than five minutes on a laptop computer using the open-source HiGHS solver (Huangfu and Hall, 2018). The source code to fully reproduce the results, tables and figures is available at `https://github.com/HenriDeh/FSRU.jl`.

3.5.1 Greenfield scenarios

The unrestricted greenfield scenarios aim to validate the actual FSRU import capacity expansion plan for Germany. The model invests for a total of 45 bcm of regasification capacity through FSRUs in the three scenarios. This is achieved by upgrading the same selection of eight ports. The exact plans obtained by the model are shown in the three upper plots of Figure 3.5.

The three scenarios exhibit a short-term investment behavior: the installed capacity (5 bcm) becomes unused after only three to four years. After that, the use of FSRUs remains constant until 2029 (NZE), 2030 (LIN), or 2033 (DEG), when progressive decommissioning begins. The port of Lubmin is the last to cease operations

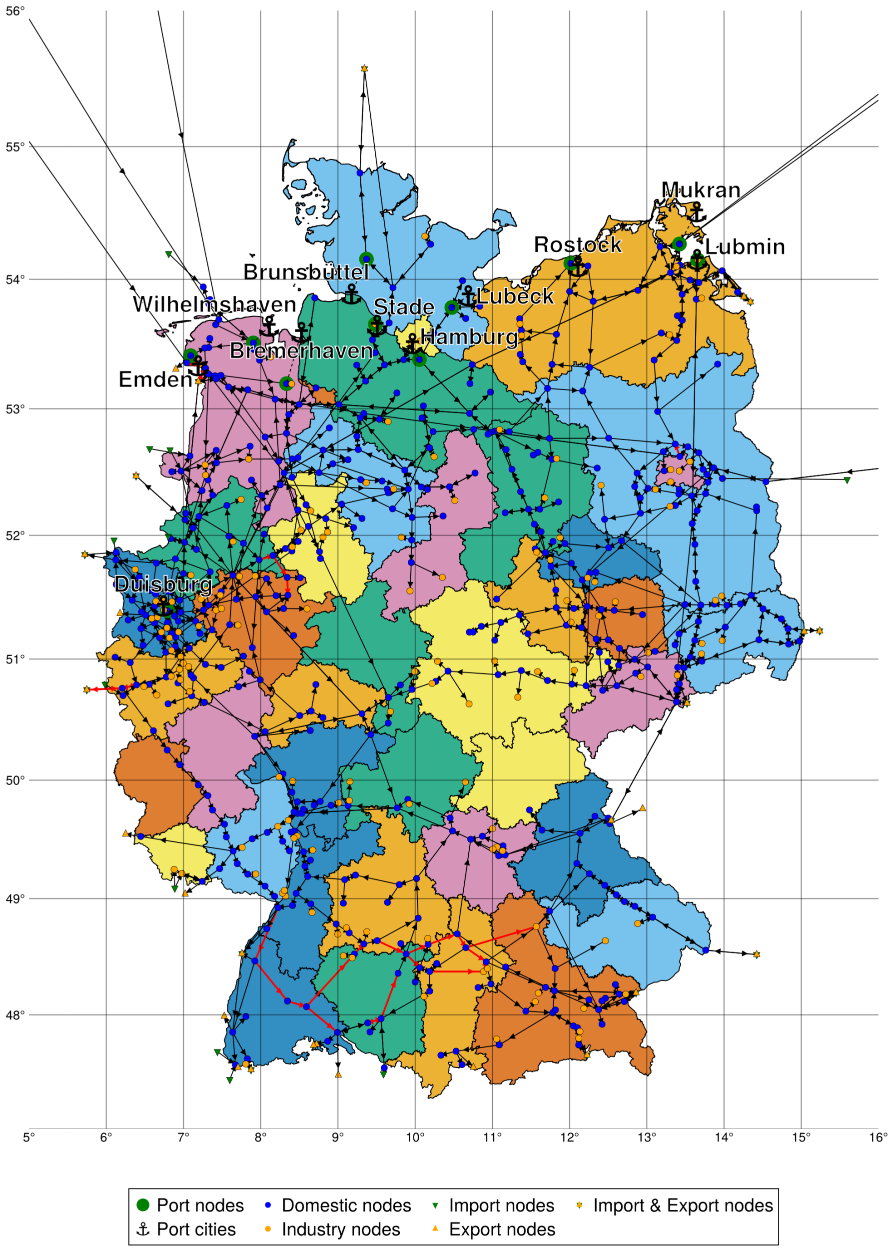


Figure 3.4: Map of Germany, divided in NUTS2 regions, with the modelled gas transmission network. Arcs with an overcapacity relaxation are emphasized in red. Cropped-out nodes are import nodes.

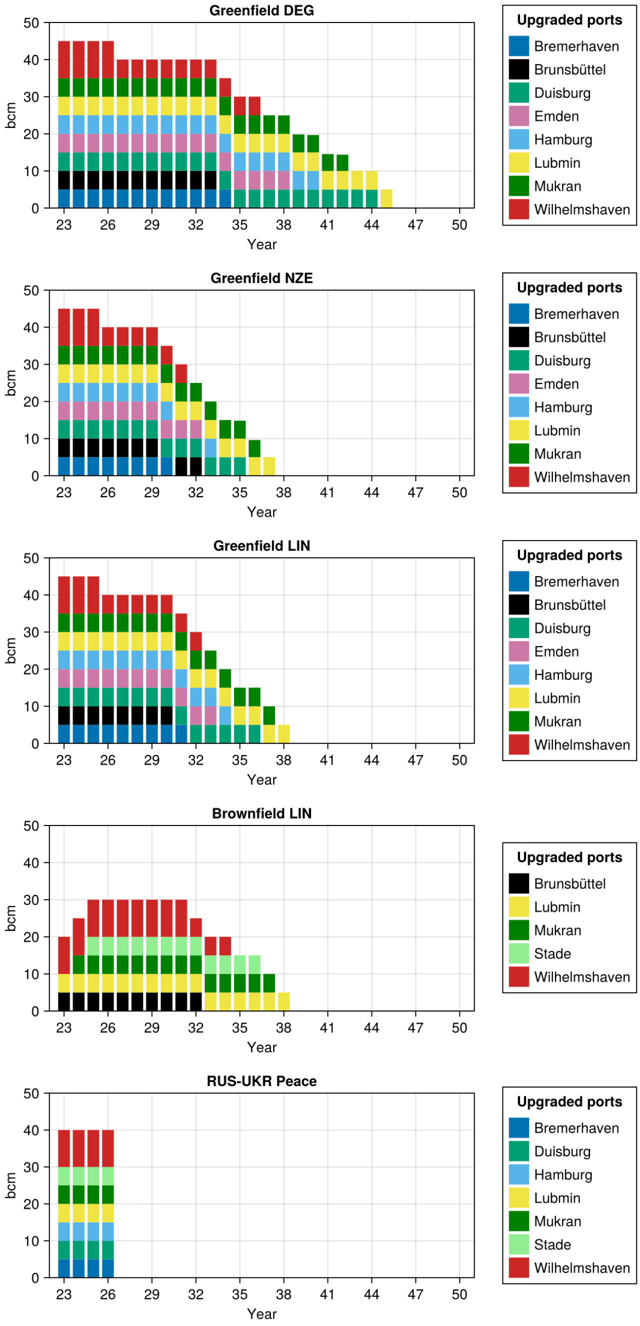


Figure 3.5: FSRU imports over time by port and scenario.

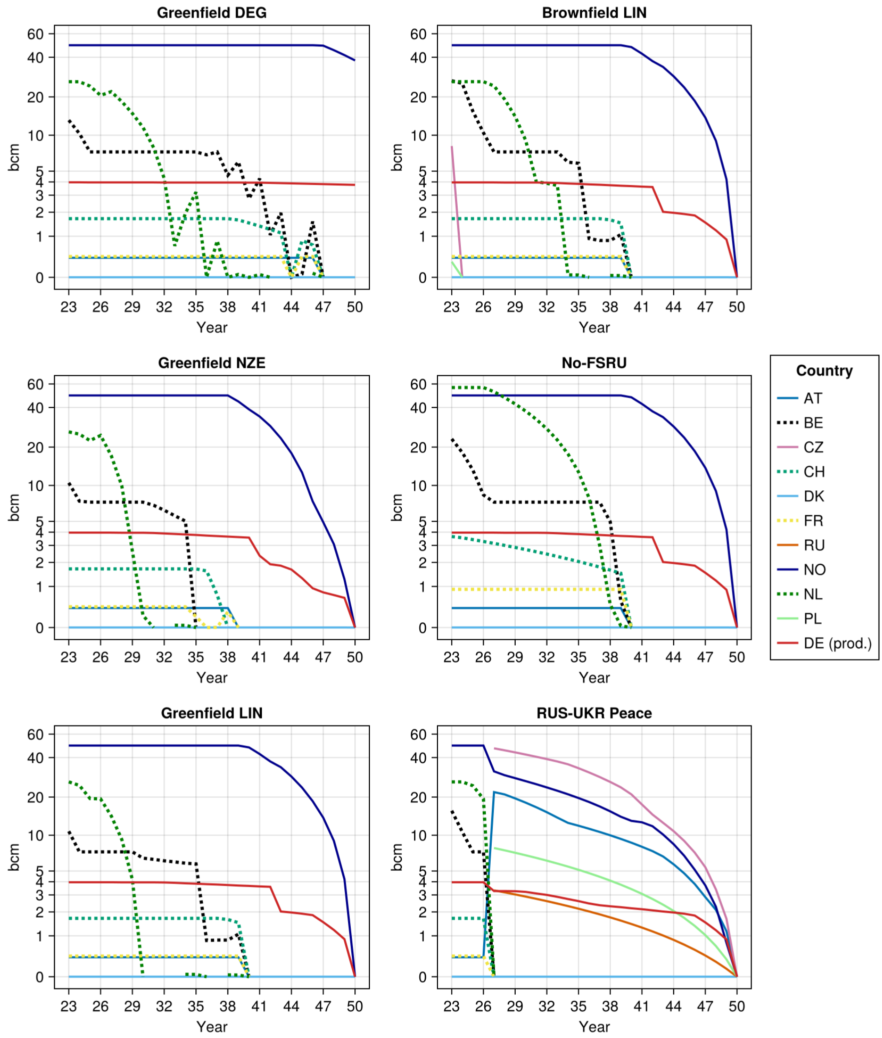


Figure 3.6: Pipeline import from connected countries over time for each scenario. Countries with a dotted supply line are that of the LNG category in Table 3.4. For clarity, the lines are rendered only where strictly positive.

Scenario	Total	Capex	Opex	FSRU imports	LNG imports	Producers imports	Transit imports	Domestic production
Greenfield DEG								
M€	476,922	1,720	7,168	212,093	119,975	122,345	3,716	9,905
bcm	2,541	-	-	698	368	1,355	9	110
Greenfield NZE								
M€	322,723	1,720	4,607	136,494	79,374	90,712	2,478	7,338
bcm	1,785	-	-	449	244	1,005	6	81
Greenfield LIN								
M€	345,829	1,720	5,017	148,811	80,675	98,994	2,632	7,979
bcm	1,929	-	-	490	248	1,097	7	88
Brownfield LIN								
M€	346,931	1,110	3,722	110,848	118,285	98,994	5,993	7,979
bcm	1,928	-	-	365	363	1,097	15	88
No-FSRU								
M€	350,121	0	0	0	240,515	98,994	2,632	7,979
bcm	1,930	-	-	0	738	1,097	7	88
RUS-UKR Peace								
M€	243,015	1,512	1,633	48,592	47,361	50,883	86,693	6,340
bcm	1,894	-	-	160	145	564	955	70

Table 3.6: Comparison between the optimal solutions of the scenarios. Capex includes pipeline construction costs. Pipeline gas import costs are divided between the source categories of Table 3.4 in addition to FSRU imports. All costs are nominal (undiscounted) over the 2023-2050 horizon.

in 2037 and 2038 in the NZE and LIN scenarios, respectively. In the DEG scenario, the FSRUs remain in use significantly longer, with a cease-of-operations in 2047.

As can be seen in the three left plots in Figure 3.6, the logic followed by the solutions obtained in the flow model is to use import sources in increasing price order, beginning with the least expensive HH nodes, the FSRU ports, the import nodes indexed by TTF and finally the expensive *Transit* countries. The model avoids the expensive imports from Czechia and Poland altogether by increasing the FSRU regasification capacity higher than what is strictly necessary to satisfy the demand (which would be 20 bcm). These results indicate that in addition to being a source of supply security, the installation of FSRU terminals is also an economical choice in the context of onerous alternative sources. However, the model does not choose to upgrade all ports and is still based on supply from Austria and the countries in the LNG category (Table 3.4). This is because all port locations are concentrated in the north of the country, meaning that pipeline capacities and the costs associated with longer transportation to the southern demand nodes (e.g., due to compression and leakage usage) are enough to justify purchasing from more expensive suppliers.

Unsurprisingly, all infrastructure investments are made in the first period, as the regasification capacity is the most needed when the demand is highest.

3.5.2 Brownfield scenario

In the brownfield scenario, we impose that the upgrading of the ports' infrastructures aligns with the actual plan of the German authorities, i.e., the upgrading of the ports of Lubmin, Brunsbüttel, and Wilhelmshaven in 2023, Mukran in 2024 and Stade in 2025. The model is not allowed to upgrade additional ports. The results are shown in Figures 3.5 and 3.6. As shown in Table 3.6, the total cost difference between this scenario and the *Greenfield LIN* scenario is minimal and mainly entails a different balance between FSRU imports and LNG imports. We note that the only new port selection compared to the Greenfield case is Stade. In the Greenfield scenario, Hamburg is instead selected, the difference is likely due to Hamburg sharing less pipeline capacity with Brunsbüttel, another port selected in all scenarios.

3.5.3 No-FSRU scenario

The No-FSRU scenario assumes that investments in regasification capacities are not considered by German authorities. Instead, to ensure feasibility in supply quantities, we suppose that such increases in regasification capacities will be made by neighboring LNG exporters (Belgium, France, the Netherlands, and Switzerland). Specifically, the parameter $s_{c,t}$ for the four countries is increased by a volume equal to a share of the 63.43 bcm formerly delivered by Russia, before the war in Ukraine (Bundesnetzagentur, 2023, averaged from January 1st to May 30th, 2022). This share is equal to the relative supply capacities of the countries in the base *Greenfield LIN* scenario. All other parameters are set to the values in the *Greenfield LIN* scenario.

Again, the total cost is not significantly higher than that in the *Greenfield LIN* scenario.

3.5.4 Peace scenario

The peace scenario assumes a resumption of trade with Russia following a peace agreement with Ukraine in 2027. This timing is purposely very optimistic. Further delaying this event would gradually converge this scenario towards the *Greenfield LIN* scenario. We modeled the resumption of trade by relaxing the supply capacity, $s_{c,t}$, of Russia and transit countries (namely Austria, Czechia, and Poland) from 2027. These countries also see their selling prices set to the HH index from 2027 on. All other parameter values remain equal to that of the *Greenfield LIN* scenario. A renewed dependence on Russian gas would naturally reduce the total expenditure by 29% compared to the *Greenfield LIN*, using FSRU only for the initial period until 2026. As the scenario neither seems politically desirable nor environmentally compatible with the EU policy objectives, the calculations can be seen as a lower bound to the attainable cost for the gas supply until 2050.

3.5.5 Sensitivity analysis

To test the robustness of our policy results versus our data estimates, we performed a sensitivity analysis on several parameters. First, the most uncertain parameter is probably the price of imported gas from neighboring countries. We tested the sensitivity of our estimates by fixing the HH price and simultaneously changing the value of the LNG, TTF, and Austrian prices by equal relative changes. Second, our estimates of capex and opex costs are based on the 2017 article by Songhurst. Although we corrected for inflation from 2017, the estimate does not account for

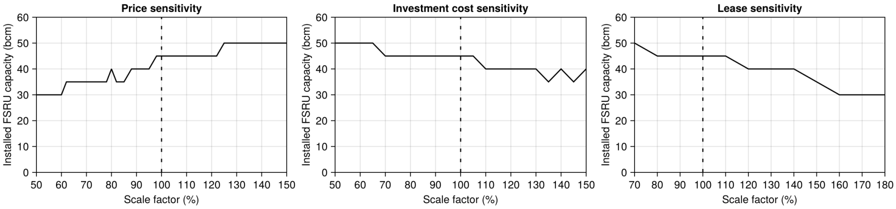


Figure 3.7: Sensitivity of the optimal investment policy to four parameters. All horizontal axes are in percentages relative to the default values of the *Greenfield LIN* scenario (vertical dashed lines).

several location-specific factors, as mentioned earlier. Thus, we test the sensitivity to the value of up_p (Equation 3.4.1). This also affects the opex by a small margin. Third, the lease cost of 46M€per year we used is that of the agreement disclosed by Finnish authorities (Reuters, 2022). It is possible that the increased attractivity of FSRUs due to the Ukrainian conflict resulted in an increase in charter rates since 2022. Finally, we tested the sensitivity to the discount rate (r) and to the number of annuities in the calculation of the capex parameter. For the latter, we only tested a change from 10 to 28 annuities (the horizon length).

The sensitivity we tested is that of the total installed FSRU capacity, that is, how the value of

$$\sum_{p \in V_p, t \in T} U_{p,t} \times q_p \times 5$$

varies when changing parameter values by small increments. All tests were performed relative to the *Greenfield LIN* scenario. The results are shown in Figure 3.7.

The analysis reveals that a decrease of 5% in price, an increase of 10% in the investment cost, or 20% of the lease cost would result in a reduction of the optimal installed capacity by 5 bcm. This observation is not surprising considering that the last 5 bcm of capacity are only used for three years in the base scenario (see Figure 3.5). An increase of 60% of the lease cost or a reduction of 50% of prices is necessary to reach an optimal installed capacity equal to that of the *Brownfield LIN* scenario (30 bcm). We found no sensitivity to the discount rate (r) for values between 0.25% and 9.25%, nor to the number of annuities for the estimation of the capex. We note that the non-monotonicity of the sensitivity that can be observed at the 80% and 140% marks of the left and center plots of Figure 3.7 is due to the inability of the model to always converge to the optimal solution under the allowed five minutes of compute per iteration.

3.6 Comparison with actual planning in Germany

As a consequence of the war in Ukraine, the German government issued in March 2023 a report on the planning and capacities of the FSRUs and onshore LNG terminals, (BMWK, 2023a). Prepared by the Federal Ministry for Economic Affairs and Climate Protection (BMWK) and based on a study of the Energy Economics Institute at the University of Cologne (EWI) (David et al., 2023), the report highlights the main milestones of the implementation process for regasification terminals in Germany. By 2025, five federal FSRUs (Wilhelmshaven I and II, Brunsbüttel, Stade, and Lubmin) should be operating all year round with a joint (nominal) regasification capacity of around 27 bcm/ year. To substitute the corresponding FSRUs, between 2026 and 2027, three onshore terminals (Stade, Brunsbüttel, and Wilhelmshaven) would be implemented to reach a total (nominal) regasification capacity of the system of 54 bcm/year, although the final investment decisions for these last three terminals were not made at the publishing date of the report. Currently, it seems that the Stade terminal will be the only one operational in 2027, as it is expected to receive the final investment decision (FID) before the end of 2024 (Global-LNG, 2024). Of the three terminals, Brunsbüttel and Stade are alleged to be prepared for later operation with hydrogen derivatives, whereas the onshore terminal at Wilhelmshaven would be designed as a green gas terminal for synthetic methane derived from green hydrogen. In February 2024, a terminal on the island of Mukran, close to Lubmin, planned for two FSRUs, started its operations with one FSRU (Energy, 2024). The second FSRU is scheduled to be transferred from Lubmin, where it is currently operating, to Mukran by the summer of 2024 (Energy, 2024). We note two critical differences in the assumptions in our model compared to the German government plans.

First, the demand scenario in BMWK (2023a) differs from both the one in our model and the background data in David et al. (2023). As can be observed in Figure 3.3, we consider an initial demand (2022) of 86.7 bcm versus 81.5 and 85.5 bcm in the BMWK and EWI studies, respectively. By 2030 the figures are 61.92, 74.1, and 58.2 bcm respectively, that is, the German government assumes a final demand that is 19.6 % to 27 % higher than the policy predictions and the EWI estimates. From 2030 on, we consider our model scenario decreasing linearly to reach zero demand in 2050 under the European policy objectives. The slow phase-out is in contradiction with the actual demand reactions in 2022/2023 to the supply reductions, as well as with the Green Deal projections.

Second, our brownfield model investment budget is estimated to be 4,832 M€ (capex + opex) compared to an initial budget of 6,560 M€ (later expanded to 9,800 M€ for the period 2022-28) in BMWK (2023a). This may be due to the stated hydrogen refitting, but given a lack of clear indications of the future infrastructure for hydrogen, it may also be a signal of overinvestment.

Third, a significant cause of the demand differences between our model and the public plans stems from the supply safety margin, planned to increase from 3,5 bcm in 2024 to 34,4 bcm in 2030. Hence, the German government effectively plans to increase the LNG import capacity radically to create energy autonomy in natural gas.

3.7 Conclusions and policy implications

The underpinnings of the ambitious European Green Deal (European Commission, 2019) and related policies resemble a dynamic game-theoretic problem with a common Pareto-optimal final equilibrium. By coordinating national energy policies for decarbonization of the economy, electrification of transport and heating, massive investments in renewable generation electricity and strengthening the cross-border infrastructure, Europe can achieve a quantum leap in terms of limiting climate change. However, on the path to 2050, each member state needs to ensure a socially and economically acceptable national trajectory to implement these objectives. The radical and unpredicted interruption of the natural gas supply for Europe is not only a socio-economic challenge for the highly indebted continent after COVID-19, but also an acid test for the consistency of the common versus individual objectives, beyond the political boundaries. Initially, few believed that the European energy sector would be able to replace the shortfall with other sources and simultaneously dampen the price impact by resolute demand shocks using the price instruments.

We put ourselves in the position of the German government and solve the complex mathematical optimization to meet the national supply objectives, both with *ex-ante* options open before the 2022 Ukraine war and the 2024 options after the initial investments, while keeping the common decarbonization objectives in 2050.

As an initial scenario, we model the Brownfield scenarios as the future path when taking the already decided investments in FSRU leases and port infrastructure for given. Our results for this scenario confirm the German decisions, both in terms of capacity (number of FSRUs) and locations (ports fitted with LNG terminals). Although short-term costs may seem excessive, the alternative costs resulting from curtailing industrial and commercial load and the social costs of price increases are surely even more discouraging and less politically manageable. One may, therefore see the first scenario as a sign of confidence in the rationality and unprecedented rapidity of the decisions made to maintain the energy balance through legal, financial, and regulatory means.

To compare the Brownfield solutions, we calculate the investment trajectory under three unrestricted (Greenfield) scenarios to design the supply network, using the German government's demand estimates (DEG) (BMWK, 2023a), the Net Zero Emission (NZE) study (David et al., 2023) and a linear reduction of demand (LIN). Whereas the latter two respect the Green Deal commitments for a phase-out in 2050, we extrapolate the DEG demand curve to a mere 50% reduction in the gas demand for 2050. Our results for the Greenfield scenarios LIN and NZE are similar and robust to a series of assumptions for cost and structural parameters. Both scenarios invest in eight ports and deploy FSRU until 2036 and 2037, gradually reducing the intensity from 2029 and 2030, respectively. The DEG scenario shows significant differences in terms of the depth of the FSRU deployment. However, the locations, capacities, and costs in the NZE and LIN models are close to those decided by the German government. The main differences result from a longer and more intensive use of FSRUs for LNG import to cover the large residual gas demand. Interestingly, even in a somewhat pessimistic DEG scenario, falling short of EU objectives, the existing infrastructure without FSRUs and Russian gas imports satisfies the forecasted gas demand by the end of the period.

Contrasting scenarios in the form of a No-FSRU policy and a Peace scenario with resumed Russian low-cost import of pipeline gas provide additional information about the robustness of the policy. Without FSRUs, Germany would have to rely on three times as much LNG imports from neighboring states at a higher cost and lower supply security. A hypothetical reactivated gas import from Russia at HH prices from 2027 would bring economic advantages but with significant political and energy supply security risks. Our findings are confirmatory and positive to the extent that given equal demand assumptions, the German government's Brownfield scenario and our Greenfield scenarios validate the same infrastructure investments in ports and similar FSRU deployment policies at almost the same cost (LIN) or at most 6% lower (NZE). Compared to earlier estimates for investments in port and gas infrastructure, the actual outcome of the Brownfield scenario shows that permits, construction, and commissioning are being carried out faster than expected. We note that the German authorities foresee a slower-than-projected phase-out of natural gas in the official projections and, consequently, an increased supply reserve capacity compared to the optimized scenario. Likely, the economic shocks to competitiveness and the socio-economic tensions that would result from the previously projected decarbonization of the German energy sector have gotten a second thought. So far, the idea of a common European energy policy and market is still a necessary, but not sufficient, condition for environmental and economic sustainable growth. Germany is no unique case in pursuing a radical LNG policy, other countries such as Spain, Greece, France, and Italy have increased their use of LNG terminals, invested in more infrastructure, or leased FSRUs. Potentially, the use of flexible resources such as FSRUs does enable redeployment to other countries and areas if the local demand reduction is stronger than the forecast, that is, closer to the European commitments.

3.7.1 Input on current policy discourse

Summarizing, the current policy discussion has expressed concern about four important economic and energy policy issues: (i) how much extra import capacity is needed to ensure supply security until 2050, (ii) what proportion of the future gas imports should be flexible (FSRU and/or LNG), (iii) for how long do we need the extra LNG import capacity, and finally, (iv) are the German government actions compatible with the European Green Deal? Our paper provides clear answers to all questions: (i) the capacity ramp-up will require the installation of eight ports for FSRU and LNG import as early as possible, with the associated minimal investments in gas infrastructure to connect the new ports, (ii) about 38% of the expected gas demand during the period 2023-2050 will be covered by flexible import from FSRUs and LNG in regasification plants, decreasing over time, and (iii) the scale-down of the FSRU capacity is planned already from 2030 and fully dismantled in 2037. Indeed, (iv) the initiated decisions in port investments, FSRU deployment versus fixed regasification assets, and network reinforcements are close to the optimal decisions under the Green Deal demand trajectory.

3.7.2 Lessons for policy makers

The analysis of the initial narrative, the actual decisions made, and the model findings provide several insights for policymakers in the sector. First, in terms of crisis preparedness and mitigation, the actual response by the German government is an example of how a serious supply crisis can catalyze faster-than-expected permitting, construction, and commissioning of critical infrastructure. The strategic investments should be designed with built-in flexibility to respond rapidly to geopolitical disruptions without derailing long-term climate commitments. Second, while national energy security is paramount for individual member states, its pursuit must not overshadow broader decarbonization efforts. Policymakers should prioritize mechanisms that harmonize short-term national actions with collective EU objectives, ensuring that the Green Deal's long-term goals remain achievable. Our model findings (that were developed without connections to the German government) show general consistency between the investment profile and the Green Deal assumptions that we imposed. However, the official demand predictions and storage requirements signal a discrepancy between the European decarbonization objectives and the actual supply planning for Germany, an issue that merits attention in the future. Third, strategic investments under the European energy transition policy should already anticipate the future hydrogen infrastructure planning to avoid lock-ins and costly conversions of stranded prime-age assets. Here, the use of FSRU rather than stationary regasification facilities or gas pipeline investments to other transit countries or LNG ports is a flexible and pragmatic choice, considering the analysis limitations discussed below.

3.7.3 Limitations and future research

Some caveats must be applied to extrapolate the findings. Our models are stylized in the sense that supply and demand are deterministic and linearly decreasing according to the policy objectives, although it is not likely to change the qualitative findings or the policy comparison. However, actual decisions may involve explicit or implicit risk analyses on supply, demand, or market developments resulting from national and European infrastructure choices. Although our sensitivity analyses for economic parameters such as capital, leasing, and investment costs confirm the qualitative robustness of the greenfield findings, the critical assumption in our calculations has been the 2050 phase-out of fossil fuels. The exact shape of the demand curve to this endpoint is not critical to the conclusions for the gas infrastructure, FSRU deployment would naturally adapt to the rate of change. The regasification dimensioning may also partially be explained by seasonality in demand, which is not considered in our model, where annual demand is used, and storage is assumed to cover seasonal differences. Another important limitation concerns the conversion of the gas transmission network for hydrogen transport. We do not model any refitting or expansion of the gas network for hydrogen use, locations, or volumes. The German National Hydrogen Strategy BMWK (2020) hardly mentions the existing gas infrastructure, but later updates in BMWK (2023b) after the gas crisis outline the hydrogen core network, which is further detailed in the Hydrogen Acceleration Act BMWK (2024b). The current plans for the network foresee a network of 9,700 km to be ready by 2032, of around 5,800 km of refitted gas pipelines and 3,900 km of new installations, 15 interconnectors, and generalized hydrogen import capacities for LNG ports. The Import Strategy for Hydrogen BMWK (2024a) explicitly supports the German government permitting for the port of Wilhelmshaven, mentioned also by investors in the case of port upgrades in Mukran and Lubmin. In addition, the national strategy also contains provisions for tendered investments in hydrogen-powered plants for electricity generation. The horizon impact of such plants is not clear, as the commissioning probably will involve a transition period using natural gas, given the very low worldwide production of hydrogen at the outset. In summary, the roll-out of the hydrogen core network could likely change the load, the geographical location of the load, and the transport capacity of the residual gas network, which may have an impact on the optimality of our solutions. Fundamental work is needed to formulate greenfield and brownfield models for hydrogen conversion of gas infrastructure in scenarios like the Green Deal with existing asset bases and sites. Future research on the cost of asset conversion and the optimal transition for fuel switching in Europe will be needed to fully answer these questions.

Finally, our model assumes a cost-minimizing decision-maker; however, as noted earlier, a more realistic formulation should incorporate the risk-minimizing behavior that likely drives actual decision-making in practice. Several axes of uncertainty merit consideration: robustness against further major supply disruptions of the same nature or scale as that of the conflict in Ukraine; stochasticity in the pace of natural-gas phase-out; the as-yet undetermined schedule for investment in hydrogen distribution infrastructure; and the increased volatility of international markets following recent commercial tensions between the EU and the US. Possible modeling approaches include making risk minimization an explicit objective or introducing chance-constrained formulations.

Chapter

FOUR

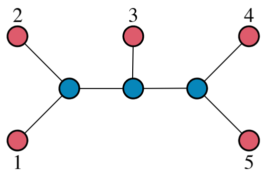
NEW HEURISTICS FOR PHYLOGENY ESTIMATION UNDER THE BALANCED MINIMUM EVOLUTION CRITERION

Abstract

Recent advances in the combinatorics of the *Balanced Minimum Evolution Problem* (BMEP) enabled the characterization of the mathematical properties that a symmetric integer matrix of order $n \geq 3$ must satisfy to encode the *Path-Length Matrix* (PLM) of an *Unrooted Binary Tree* (UBT). This result, together with the identification of fundamental facet-defining inequalities for the convex hull of BMEP solutions, has led to an integer linear programming formulation that currently serves as the reference exact solution algorithm. Here, we show how to exploit these advances to improve the approximation algorithms for the problem. We first leverage the tight linear programming relaxation of this formulation to develop an enhanced Neighbor Joining–like heuristic. Next, we embed this heuristic into a *Beam Search* framework to further improve the quality of the solutions. Computational experiments show that the proposed algorithms outperform existing heuristics, making their use highly desirable in practice.

4.1 Introduction

Consider a set $\Gamma = \{1, 2, \ldots, n\}$ of $n \geq 3$ distinct, aligned molecular sequences, hereafter referred to as *taxa*, and a $n \times n$ symmetric matrix $\mathbf{D} = \{d_{ij}\}$ of evolutionary distance estimates between pairs of taxa in Γ, such that $d_{ii} = 0$, for all $i \in \Gamma$, and $d_{ij} > 0$, for all $i \neq j \in \Gamma$. Let T denote a *phylogeny* of Γ, i.e., an *Unrooted Binary Tree* (UBT) having Γ as set of leaves and let $\mathcal{U}_n$ denote the set of the possible $(2n-5)!!$ phylogenies of Γ (Catanzaro et al., 2020b). For a given phylogeny $T \in \mathcal{U}_n$, let P_{ij}^T



$$\tau = \begin{pmatrix} 0 & 2 & 3 & 4 & 4 \\ 2 & 0 & 3 & 4 & 4 \\ 3 & 3 & 0 & 3 & 3 \\ 4 & 4 & 3 & 0 & 2 \\ 4 & 4 & 3 & 2 & 0 \end{pmatrix}$$

Figure 4.1: An example of a phylogeny of five taxa represented by a UBT with 5 leaves and the associated PLM $\boldsymbol{\tau}$.

denote the (unique) path in T connecting taxa $i, j \in \Gamma$ and let $\boldsymbol{\tau} = \{\tau_{ij}\}$ denote the *Path-Length Matrix* (PLM) associated to T, i.e., a $n \times n$ symmetric integer matrix, whose generic entry $\tau_{ij} \in [2, n-1]$ represents the number of edges on P_{ij}^T. For example, Figure 4.1 shows a phylogeny of five taxa and the associated PLM. Finally, let Θ_n denote the set of PLMs associated with the phylogenies in $\mathcal{U}_n$. Then, the *Balanced Minimum Evolution Problem* (BMEP) is the problem

$$\min_{\boldsymbol{\tau} \in \Theta_n} \ell(\tau) = \sum_{i \in \Gamma} \sum_{\substack{j \in \Gamma \\ i \neq j}} d_{ij} 2^{-\tau_{ij}}. \tag{4.1}$$

The BMEP formalizes, in terms of a nonlinear combinatorial optimization problem, the *Balanced Minimum Evolution* (BME) criterion of phylogenetic estimation, introduced in the literature on distance-based methods by Desper and Gascuel in 2004. BME originates from Rzhetsky and Nei's *Minimum Evolution* (ME) criterion and primarily differs from ME in its use of a specific branch-length estimation model introduced by Pauplin in 2000. This model was designed to address key limitations of the classical Ordinary Least-Squares method, the most critical of which was the occurrence of negative branch lengths (see Catanzaro et al., 2022 for a recent historical review of BMEP). BME posits that the phylogeny of a given set of molecular sequences (taxa) is the one that minimizes the sum of their *cross-entropies* (Catanzaro et al., 2020a). This phylogeny is provably *statistically consistent*, i.e., it converges to the true phylogeny of taxa if the distance estimates d_{ij} approximate the true evolutionary distances of taxa when considering longer and longer molecular sequences (Desper and Gascuel, 2004; Gascuel, 2005; Gascuel and Steel, 2006). BME figures among the most accurate estimation criteria in distance-based methods (Lefort et al., 2015a); it can be easily adapted to any kind of molecular data for which a dissimilarity measure is available (Schwartz, 2019) and because it works on distance estimates between taxa rather than on their molecular sequences, it results to be computationally less demanding than alternative estimation criteria based on maximum parsimony, maximum likelihood, or Bayesian inference (Gascuel, 2005). These features make BME particularly suitable for general-purpose phylogenetic analyses.

Inferring phylogenies under the BME criterion implies solving the BMEP. Unfortunately, this problem is $\mathcal{NP}$-hard and inapproximable within c^n, for some constant $c > 1$, unless $\mathcal{P} = \mathcal{NP}$ (Fiorini and Joret, 2012). This negative result holds even in the case where the structure of the UBT is fixed and only the assignment of the n

taxa to the n leaves of the UBT needs to be identified (Frohn, 2021). The BMEP can however be approximated within a factor of two if $\mathbf{D}$ is *metric*, i.e., if its entries also satisfy the triangle inequality (Fiorini and Joret, 2012).

The pursuit of practical solution algorithms for the BMEP has driven two decades of extensive research, focusing on its combinatorial properties (Semple and Steel, 2004; Pardi, 2009; Pardi et al., 2010; Cueto and Matsen, 2011; Haws et al., 2011), optimization aspects (Forcey et al., 2016, 2017; Catanzaro et al., 2020b), and the development of both exact solution algorithms (Pardi, 2009; Catanzaro et al., 2022) and heuristic approaches (Gascuel, 2005; Pardi, 2009; Fiorini and Joret, 2012; Lefort et al., 2015a; Gasparin et al., 2023) to tackle its instances. Particularly relevant for this chapter are some recent theoretical advances – briefly summarized in the next section for the sake of completeness – that led to the characterization of the set of the PLMs of UBTs, of basic properties and facets of their convex hull, and to the development of an *Integer Linear Programming* (ILP) formulation that currently serves as the reference exact solution algorithm for the problem (see Catanzaro et al., 2024). Here, we show that these findings can be leveraged to design matheuristic methods capable of constructing phylogenies with shorter tree lengths than current state-of-the-art methods. This chapter is organized as follows. In the next section, we briefly recall both some classic results on the *Tree Realization Problem* (TRP), introduced by Buneman (1974) and Waterman et al. (1977), and some recent results on the characterization of Θ_n that constitute the theoretical foundation of the ILP formulation presented in Catanzaro et al., 2024. In Section 4.3, we present the construction heuristic for the BMEP. In particular, we discuss first the core matheuristic and its rationale and then we show how it can be embedded within a Beam Search to enhance the quality of the resulting solution. In Section 4.4, we discuss the computational experiments obtained when running the proposed algorithms in combination with the local search method in FastME 2 (Lefort et al., 2015a) on a number of benchmark instances of the BMEP. Finally, in Section 4.5, we present some open questions that may inspire future developments.

4.2 Some background on the TRP and the BMEP

Given a matrix $\mathbf{D} = \{d_{ij}\}$ on a target set S of $n \geq 3$ points, the TRP asks whether there exists a *tree realization* of $\mathbf{D}$, i.e., a weighted tree (T, w) having S as set of leaves and nonnegative branch-lengths $w = \{w_e\}$ satisfying the following *path-length constraint*:

$$\sum_{e \in P_{ij}^T} w_e = d_{ij} \quad \forall\, i < j \in S. \tag{4.2}$$

A classical result for metric distance matrices states that

Proposition 1. *(Buneman, 1974)*
There is a tree realization for a metric distance matrix $\mathbf{D}$ on a target set S of n points if and only if for any four distinct $i, j, p, q \in S$ the following four-point condition *holds:*

$$d_{ij} + d_{pq} \leq \max\left\{d_{ip} + d_{jq}, d_{iq} + d_{jp}\right\}. \tag{4.3}$$

Observe that the tree that realizes a metric distance matrix $\mathbf{D}$ is not necessarily unrooted and binary, but can be transformed into a UBT by adding dummy nodes and edges with possibly null branch-lengths.

Determining the properties that an integer metric matrix $\boldsymbol{\tau}$ must satisfy to encode the PLM of a UBT in $\mathcal{U}_n$ is equivalent to determining the properties that ensure a tree realization for $\boldsymbol{\tau}$ when also imposing that the weighted tree (T, w) is a UBT with branch-lengths all equal to 1. This further requirement, unfortunately, makes the four-point condition necessary to ensure a *UBT realization* but not sufficient (Catanzaro et al., 2024). Denoted by $[\alpha, \beta]_{\mathbb{N}}$ the interval of the integers between α and β (included), a possible characterization of a UBT realization is provided by the following result:

Proposition 2. *(Catanzaro et al., 2024)*
A square matrix $\boldsymbol{\tau}$ of order $n \geq 3$ encodes the PLM of a UBT $T \in \mathcal{U}_n$ if and only if it satisfies

$$\tau_{ii} = 0 \qquad \forall\, i \in \Gamma \tag{4.4}$$

$$\tau_{ij} \in [2, n-1]_{\mathbb{N}} \qquad \forall\, i < j \in \Gamma \tag{4.5}$$

$$\tau_{ji} = \tau_{ij} \qquad \forall\, i < j \in \Gamma \tag{4.6}$$

$$\tau_{1i} + \tau_{1j} \geq 2 + \tau_{ij} \qquad \forall \textit{ distinct } i, j \in \Gamma \setminus \{1\} \tag{4.7}$$

$$\tau_{1i} + \tau_{ij} \geq 2 + \tau_{1j} \qquad \forall \textit{ distinct } i, j \in \Gamma \setminus \{1\} \tag{4.8}$$

$$\sum_{\substack{j \in \Gamma \\ j \neq i}} 2^{-\tau_{ij}} = \frac{1}{2} \qquad \forall\, i \in \Gamma \tag{4.9}$$

and, for all distinct $j, p, q \in \Gamma \setminus \{1\}$, exactly one of the following strong four-point conditions*:*

$$\tau_{1j} + \tau_{pq} + 2 \leq \tau_{1p} + \tau_{jq} = \tau_{1q} + \tau_{jp} \tag{4.10a}$$

$$\tau_{1p} + \tau_{jq} + 2 \leq \tau_{1j} + \tau_{pq} = \tau_{1q} + \tau_{jp} \tag{4.10b}$$

$$\tau_{1q} + \tau_{jp} + 2 \leq \tau_{1j} + \tau_{pq} = \tau_{1p} + \tau_{jq}. \tag{4.10c}$$

Conditions (4.4)-(4.6) impose that $\boldsymbol{\tau}$ must be a symmetric matrix with null diagonal and integer nondiagonal entries in $[2, n-1]$. Conditions (4.7) and (4.8) impose $\boldsymbol{\tau}$ to be metric. Condition (4.9) impose *Kraft's equality* that ensure the degree constraint on the internal nodes of the weighted tree (Catanzaro et al., 2024). Finally, Conditions (4.10) impose Buneman's four-points conditions, that ensure the connectivity and acyclicity of the weighted tree. Proposition 2 allows to reformulate the BMEP in terms of an ILP problem. Specifically, let $L = [2, n-1]_{\mathbb{N}}$ denote the set of values that can be taken by a generic path-length τ_{ij}. In addition, let x_{ij}^{ℓ} denote a decision variable equal to 1 if and only if $\tau_{ij} = \ell \in L$, and let y_{pq}^{j} denote a decision variable equal to 1 if and only if the subtree containing 1 and j is disjoint from that containing p and q. Then, the following formulation is valid for the BMEP (Catanzaro et al., 2024):

$$\min \sum_{i \in \Gamma} \sum_{j \in \Gamma \setminus \{i\}} d_{ij} \left(\sum_{\ell \in L} 2^{-\ell} x_{ij}^{\ell} \right) \tag{4.11a}$$

$$
\begin{aligned}
s.t. \sum_{\ell \in L} x_{ij}^{\ell} &= 1 \qquad \forall\, i \neq j \in \Gamma && \text{(4.11b)}\\
x_{ij}^{\ell} &= x_{ji}^{\ell} \qquad \forall\, i \neq j \in \Gamma,\ \ell \in L && \text{(4.11c)}\\
\tau_{ij} &= \sum_{\ell \in L} \ell x_{ij}^{\ell} \qquad \forall\, i \neq j \in \Gamma && \text{(4.11d)}\\
\sum_{j \in \Gamma \setminus \{i\}} \sum_{\ell \in L} 2^{-\ell} x_{ij}^{\ell} &= \frac{1}{2} \qquad \forall\, i \in \Gamma && \text{(4.11e)}\\
\tau_{1i} + \tau_{1j} - \tau_{ij} &\geq 2 \qquad \forall\, i \neq j \in \Gamma \setminus \{1\} && \text{(4.11f)}\\
\tau_{1i} + \tau_{ij} - \tau_{1j} &\geq 2 \qquad \forall\, i \neq j \in \Gamma \setminus \{1\} && \text{(4.11g)}\\
y_{pq}^{j} + y_{jq}^{p} + y_{jp}^{q} &= 1 \qquad \forall \text{ distinct } j,p,q \in \Gamma \setminus \{1\} && \text{(4.11h)}\\
\tau_{1j} + \tau_{pq} &\geq 2(1 - y_{jp}^{q}) + \tau_{1p} + \tau_{jq} - (2n-2) y_{pq}^{j} \\
&\qquad \forall \text{ distinct } j,p,q \in \Gamma \setminus \{1\} && \text{(4.11i)}\\
\sum_{i \in \Gamma} \sum_{j \in \Gamma \setminus \{i\}} \sum_{\ell \in L} \ell 2^{-\ell} x_{ij}^{\ell} &= 2n - 3 && \text{(4.11j)}\\
x_{ij}^{\ell} &\in \{0,1\} \qquad \forall\, i \neq j \in \Gamma,\ \ell \in L && \text{(4.11k)}\\
y_{pq}^{j} &\in \{0,1\} \qquad \forall \text{ distinct } j,p,q \in \Gamma \setminus \{1\} && \text{(4.11l)}\\
\tau_{ij} &\in L \qquad \forall\, i \neq j \in \Gamma. && \text{(4.11m)}
\end{aligned}
$$

The objective function (4.11a) of the ILP formulation for the BMEP linearizes (4.1). Constraints (4.11b) impose that for each distinct pair of taxa $i, j \in \Gamma$, the path-length τ_{ij} takes precisely one value in L. Constraints (4.11c) impose the symmetry condition on path-lengths τ_{ij}. Constraints (4.11d) tie τ_{ij} variables to x_{ij}^{ℓ} variables. Constraints (4.11e) impose (4.9). Constraints (4.11f) and (4.11g) impose (4.7) and (4.8). Constraints (4.11h) and (4.11i) impose Buneman's four-points conditions (4.10). Constraint (4.11j) impose that path-lengths lie on a particular manifold known as *UBT manifold* (Catanzaro et al., 2020b, 2024). Finally, constraints (4.11k)-(4.11m) impose the integrality of all variables. A more compact ILP formulation can be obtained by declaring x_{ij}^{ℓ} variables only for all $i, j \in \Gamma$ such that $i < j$ and substituting τ_{ij} variables with the right-hand side of (4.11d). Here, we prefer to keep them for the sake of clarity. Formulation (4.11) is characterized by $O(n^3)$ variables and constraints. A Branch-&-Cut algorithm based on it exactly solves BMEP instances containing up to 32 taxa within the reference time of 1 hour and in the computational environment described in Catanzaro et al. (2024). For larger instances, however, this algorithm may prove too slow for practical use. Hence, in the next sections we leverage the very tight lower bounds provided by the *linear programming* relaxation of Formulation (4.11) to approximate the optimal solution $\boldsymbol{\tau}^*$ to BMEP instances.

4.3 A matheuristic for the BMEP

4.3.1 Agglomeration methods

The matheuristic described in this section belongs to the family of *agglomeration methods* (Saitou and Nei, 1987; Felsenstein, 2004b; Gascuel, 2005; Gascuel and Steel,

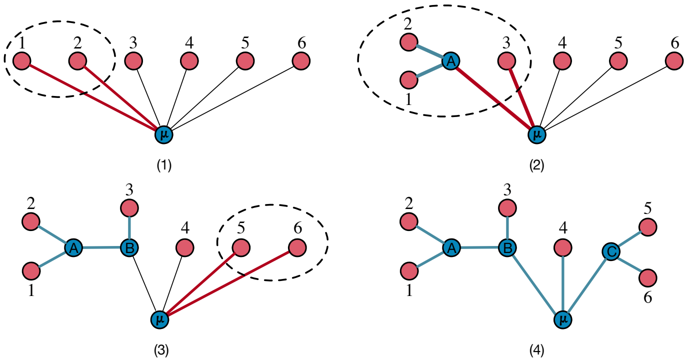


Figure 4.2: The three iterations of an agglomeration method required to generate a phylogeny of six taxa: iteration 1) merges singletons $\{1\}$ and $\{2\}$ and introduces internal node A, iteration 2) merges cluster $\{1, 2\}$ with the singleton $\{3\}$ and introduces node B; iteration 3) merges singletons $\{5\}$ and $\{6\}$ and introduces node C, resulting in the final 6-taxon phylogeny.

2006). The core idea of these methods consists of constructing first an initial star-tree constituted of a central node μ and n distinct clusters of taxa, each containing initially a single taxon in Γ. Then, at each step, two clusters of taxa are iteratively merged, with each merge introducing a new internal node that serves as the *common ancestor* of the merged clusters. This new internal node is linked to the central node μ, while the edges connecting μ to the common ancestors of merged taxa are removed. The process is repeated for $n - 3$ steps, resulting in a UBT that represents the phylogeny of Γ upon termination. Figure 4.2 shows an example of this agglomeration method.

Agglomeration methods differ in how they select clusters to merge at each step r, based on the current set N^r of common ancestor nodes and the matrix D^r of the distances d^r_{pq} between each pair of elements $i, j \in N^r$. Initially, $N^1 = \Gamma$ and D^1 is the taxa (evolutionary) distance matrix $\mathbf{D}$. At the end of step r, N^{r+1} is set equal to $N^r \cup \{v\} \setminus \{p, q\}$, where p and q are the common ancestors of the two merged clusters, and v is the common ancestor of the cluster created by the merge; D^{r+1} is derived from D^r by a *reduction operation*, i.e., by deleting the respective p-th and q-th rows and columns and by adding a new row and column whose entries encode the distances between v and the remaining clusters. For example, referring again to Figure 4.2, D^2 is obtained from D^1 by removing the rows and columns corresponding to taxa 1 and 2 and inserting a new row and column representing the distances from node A to nodes in $N^2 = N^1 \cup \{A\} \setminus \{1, 2\} = \{A, 3, 4, 5, 6\}$. In the remainder of the chapter, we denote by T^r the tree generated by the agglomeration method at step r and by s^r_{ij} the *topological distance* between the leaves i and j on T^r, i.e., the number of edges on the path P^r_{ij}. Note, in particular, that when $r = 1$, T^1 is the initial star-tree, hence $s^1_{ij} = 2$, for all $i, j \in \Gamma$. At the end of step $r = n-3$, T^{n-2} is a UBT in $\mathcal{U}_n$, hence $s^r_{ij} = \tau_{ij}$. We also define the *generalized tree length formula* $\ell(T^r)$

as (Semple and Steel, 2004):

$$\ell(T^r) = \sum_{i\in\Gamma}\sum_{\substack{j\in\Gamma\\ i\neq j}}\prod_{k\in V_{P^r_{ij}}} (\delta(k)-1)^{-1}\, d_{ij} \tag{4.12}$$

where $V_{P^r_{ij}}$ is the set of internal nodes in the path P^r_{ij} and $\delta(k)$ the degree of node k. Note that when $r = n-2$, (4.12) is precisely the objective function of the BMEP.

The *Neighbor-Joining* (NJ) algorithm (Saitou and Nei, 1987; Gascuel, 2005) is an example of a possible agglomeration method which is particularly interesting in the context of the BMEP because it greedily minimizes (4.12) (Gascuel and Steel, 2006). We recall that if p and q are the common ancestors of the two clusters that are merged at the r-th step of the NJ algorithm, then the values of the distances of the new node v from the common ancestors of the other clusters are defined by means of the following reduction:

$$d^{r+1}_{vk} = d^{r+1}_{kv} = \frac{d^r_{pk} + d^r_{qk}}{2} \qquad \forall k \in N^r \setminus \{p,q\}. \tag{4.13}$$

The common ancestors p and q of the two clusters to merge at the r-th step are instead identified by solving the following problem:

$$(p,q) = \operatorname*{argmin}_{\substack{i,j\in N^r\\ i\neq j}} (|N^r|-2)\, d^r_{ij} - \sum_{k\in N^r}\left(d^r_{ik} + d^r_{jk}\right). \tag{4.14}$$

4.3.2 The LPNJ algorithm

Building on the notation and definitions introduced in the previous sections, we now present a novel NJ-like agglomeration method, hereafter referred to as the *Linear Programming Neighbor-Joining* (LPNJ) algorithm, that leverages the *Linear Programming* (LP) relaxation of Formulation (4.11) to guide the cluster selection step. We will first introduce an agglomeration method that, given a path-length matrix, constructs the associated phylogeny. We then describe the LPNJ algorithm. Finally, we prove the equivalence between the UBT reconstruction method and a version of LPNJ that uses the optimal integer solution of Formulation (4.11) in place of the linear programming relaxation. This result serves as a theoretical justification for the proposed LPNJ algorithm.

The UBT reconstruction algorithm

Consider the following agglomeration method, hereafter referred to as the *UBT Reconstruction Algorithm*, that can be used to reconstruct the UBT T associated to a path-length matrix.

This algorithm operates with a matrix of topological distances $\boldsymbol{\tau}$, for example identified by the optimal integer solution of Formulation (4.11), instead of evolutionary distances, or, equivalently, assumes that evolutionary distances coincide with topological distances, i.e., $\mathbf{D} = \tau$. At each iteration, it merges two clusters whose distance τ^r_{ij} between the common ancestors i and j is equal to 2, and updates the distance matrix according to the following reduction operation rule

$$\tau^{r+1}_{vk} = \tau^{r+1}_{kv} = \tau^r_{ik} - 1 = \tau^r_{jk} - 1 \qquad \forall k \in N^r \setminus \{p, q\} \tag{4.15}$$

where v is the new common ancestor of the cluster created by the merge. At the end of its execution, the UBT reconstruction algorithm generates a UBT because if $\boldsymbol{\tau}^r$ is the PLM of a UBT, then there are at least two entries in $\boldsymbol{\tau}^r$ such that $\tau^r_{ij} = 2$ for $i, j \in N^r$. Furthermore, by the triangular inequalities (4.11f), if $\tau^r_{ij} = 2$, then $\tau^r_{ik} = \tau^r_{jk}$ for all $k \in N^r$. Now, observe that, for all r, $\boldsymbol{\tau}^r$ is the PLM of a UBT. First, $\boldsymbol{\tau}^1 = \boldsymbol{\tau}$ is the PLM of a UBT by assumption. Furthermore, if $\boldsymbol{\tau}^r$ is the PLM of a UBT T, then rule (4.15) makes $\boldsymbol{\tau}^{r+1}$ be the PLM of the UBT obtained by pruning the leaves i and j and their adjacent edges from T. Finally, the UBT generated by the UBT reconstruction algorithm is the phylogeny associated with $\boldsymbol{\tau}^1$, since rule (4.15) makes the distance between each pair p, q of leaves be equal to τ^1_{pq}.

The LPNJ algorithm

The UBT reconstruction algorithm described in the previous section is an agglomeration method that uses topological information (i.e., the topological distances between the nodes associated with the common ancestors of the clusters of taxa) to select the clusters to merge, as opposed to standard agglomeration methods, such as the NJ algorithm, which rely just on the evolutionary distances. The UBT reconstruction algorithm would return the optimal UBT T^* if the optimal path length matrix $\boldsymbol{\tau}^*$ were given.

The LPNJ algorithm that we present here leverages the features of both aforementioned agglomeration methods. Specifically:

- Like the NJ algorithm, it initially sets D^1 equal to the taxa evolutionary distance matrix and, at each subsequent step r, defines the distances d^{r+1}_{vk} between the common ancestor v of the newly generated cluster and the other common ancestors $k \in N^{r+1} \setminus \{v\}$ according to rule (4.13).
- Like the UBT reconstruction algorithm, at each subsequent step r, it selects the two clusters to merge based on an estimate of the topological distances of the optimal UBT T^* between their common ancestors i and j. Specifically, it estimates the distances between the common ancestors in N^r by computing the matrix $\tilde{\boldsymbol{\tau}}^r$, which is the solution to the LP relaxation of Formulation (4.11) when applied to the evolutionary distance matrix D^r. Thus, the selection operation is given by:
$$(i, j) = \operatorname*{argmin}_{\substack{p,q \in N^r \\ p \neq q}} \tilde{\tau}^r_{pq} \tag{4.16}$$

The rationale behind the LPNJ algorithm is based on the following theorem.

Theorem 1. *Consider the following version of the LPNJ algorithm. Let $\boldsymbol{\tau}^{*r}$ be an optimal integer solution of Formulation* (4.11) *applied to the evolutionary distance matrix $D^r = \{d^r_{ij}\}_{i,j \in N^r}$. Apply the cluster selection rule* (4.16) *and the reduction rule* (4.13)*. Then, there exists an optimal solution $\boldsymbol{\tau}^{*r+1}$ of* (4.11) *when applied to the evolutionary distance matrix D^{r+1} which can be derived from $\boldsymbol{\tau}^{*r}$ using the matrix reduction rule* (4.15) *of the UBT reconstruction algorithm on $\boldsymbol{\tau}^{*r}$.*

Proof. Without loss of generality assume that the LPNJ algorithm merges the clusters whose common ancestors are associated with the first and the second rows of matrix D^r. Then, $\tau^{*r}_{12} = 2$ and $\tau^{*r}_{1j} = \tau^{*r}_{2j}$ for all $j \in N^r$. In addition, $N^{r+1} = N^r \cup \{v\} \setminus \{1,2\}$, where v represents the common ancestor of the newly formed cluster. Finally, by definition $\boldsymbol{\tau}^{*r}$ is an optimal solution to the following problem (with constraints omitted for brevity).

$$
\begin{aligned}
\min_{\boldsymbol{\tau}} z^{*r} &= \sum_{i \in N^r} \sum_{j \in N^r} d^r_{ij} 2^{-\tau^r_{ij}} \\
&= 2d^r_{12}2^{-2} + \sum_{i \in N^r \setminus \{1,2\}} \sum_{j \in N^r \setminus \{1,2\}} d^r_{ij} 2^{-\tau^r_{ij}} \\
&\quad + 2 \sum_{j \in N^r \setminus \{1,2\}} (d^r_{1j} + d^r_{2j}) 2^{-\tau^r_{1j}} \\
&= 2d^r_{12}2^{-2} + \sum_{i \in N^{r+1} \setminus \{v\}} \sum_{j \in N^{r+1} \setminus \{v\}} d^r_{ij} 2^{-\tau^r_{ij}} \\
&\quad + 2 \sum_{j \in N^{r+1} \setminus \{v\}} \frac{d^r_{1j} + d^r_{2j}}{2} 2^{-(\tau^r_{1j} - 1)}.
\end{aligned}
$$

Now, consider the optimal solutions $\boldsymbol{\tau}^{*r+1}$ of the problem

$$
\min_{\boldsymbol{\tau}} z^{*r+1} = \sum_{\substack{i \in N^{r+1} \\ i \neq v}} \sum_{\substack{i \in N^{r+1} \\ j \neq v}} d^{r+1}_{ij} 2^{-\tau^{r+1}_{vj}} + 2 \sum_{\substack{i \in N^{r+1} \\ j \neq v}} d^{r+1}_{vj} 2^{-\tau^{r+1}_{vj}},
$$

where by hypothesis $d^{r+1}_{vj} = (d^r_{1j} + d^r_{2j})/2$ for all $j \in N^r$, and observe that $z^{*r} = 2d^r_{12}2^{-2} + z^{*r+1}$ holds. Now suppose, by contradiction, that there is no solution $\boldsymbol{\tau}^{*r+1}$ such that $\tau^{*r+1}_{vj} = \tau^{*r}_{1j} - 1$ for all $j \in N^r$ while $\tau^{*r+1}_{ij} = \tau^{*r}_{ij}$ for all $i,j \in N^{r+1} \setminus \{v\}$. This fact implies that $z^{*r} \neq 2d^r_{12}2^{-2} + z^{*r+1}$, which leads to a contradiction. Thus, the statement follows. □

An immediate consequence of the above theorem is that if we were able to estimate $\boldsymbol{\tau}^{*r}$ exactly at each step r of the LPNJ algorithm, it would return the optimal UBT T^*. In fact, only the minimal entries of $\tilde{\boldsymbol{\tau}}^r$ need to be exact for LPNJ to return T^*. Although this may be not the case in general, the solution $\tilde{\boldsymbol{\tau}}^r$ to the LP relaxation of Formulation (4.11) considered by the LPNJ algorithm constitutes a good approximation of $\boldsymbol{\tau}^{*r}$ (Catanzaro et al., 2024).

4.3.3 Beam search

To improve the quality of the solution provided by the LPNJ algorithm we introduce in this section a *Beam Search* approach (Lowerre, 1976) that explores multiple alternative cluster merges at each step of an agglomeration algorithm. Beam Search balances exploration and computational efficiency by maintaining a fixed number of alternatives, denoted by the beam width parameter B, at each level of the search.

Let Ξ^r be the set of alternatives considered at step r in a Beam Search approach applied to an agglomerative method. The algorithm proceeds as follows:

- `Initialization`: Set $\Xi^1 = \{T^1\}$, where T^1 is the initial star-tree corresponding to the taxa set (e.g., Figure 4.2.1).
- `For each step` $r = 1, \ldots, n-3$:
  - Compute the set C^r of all possible trees obtained from the trees in Ξ^r by merging two clusters of taxa.
  - Set Ξ^{r+1} as the set of the B-best trees in C^r according to a given *ranking criterion.*
- `Return`: The best tree in Ξ^{n-2}.

We refer to the algorithms obtained by applying Beam Search to NJ and LPNJ as *BeamNJ* and *BeamLPNJ*, respectively. We will compare the performance of these algorithms for different values of B in Section 4.4.

BeamNJ

The natural ranking criterion for selecting trees at each step in the BeamNJ method is given by the generalized tree length formula (4.12).

In the following, we show an efficient approach for computing the value of a tree $T^{r+1} \in C$, obtained from a tree $T^r \in \Xi^r$ by merging two clusters whose common ancestors are nodes p and q, respectively. To this end, let Γ_v denote the taxa cluster whose common ancestor is $v \in N^r$. By (4.13), we obtain:

$$d^r_{pq} = \sum_{i\in\Gamma_p}\sum_{j\in\Gamma_q} d_{ij} 2^{-s_{ij}-1}. \tag{4.17}$$

This result allows us to express the tree length as

$$\ell(T^r) = \hat{\ell}(T^r) + \sum_{i\in N^r}\sum_{j\in N^r} (|N^r|-1)^{-1}\, d^r_{ij} \tag{4.18}$$

where

$$\hat{\ell}(T^r) = \sum_{v\in N^r}\sum_{i\in\Gamma_v}\sum_{j\in\Gamma_v} d_{ij} 2^{-s^{r-1}_{ij}}, \tag{4.19}$$

a result that can be also easily derived from Gascuel and Steel (2006). Now, assume that p and q are the merged common ancestors at step r. Then, from (4.17) and (4.18) we obtain

$$\hat{\ell}(T^{r+1}) = \hat{\ell}(T^r) + \frac{d^r_{pq}}{2}$$

which leads to

$$\begin{aligned}\ell(T^{r+1}) = {} & \hat{\ell}(T^r) + \frac{d^r_{pq}}{2} \\ & + \sum_{i\in N^r\setminus\{p,q\}}\ \sum_{j\in N^r\setminus\{p,q\}} (|N^r|-2)^{-1}\, d^r_{ij} \\ & + \sum_{v\in N^r\setminus\{p,q\}} (|N^r|-2)^{-1}\, \frac{d^r_{vp}+d^r_{vq}}{2}.\end{aligned} \tag{4.20}$$

This expression can be rewritten as

$$\ell(T^{r+1}) = \hat{\ell}(T^r) + \frac{d^r_{pq}}{2} + \frac{\sum_{i\in N^r}\sum_{j\in N^r} d^r_{ij} - \sum_{v\in N^r} d^r_{vp} - \sum_{v\in N^r} d^r_{vq}}{2(|N^r|-2))}. \tag{4.21}$$

Observe that the three terms in the numerator, each expressed as a sum, can be precomputed for each $T^r \in \Xi^r$ to avoid redundant computation. Equation (4.21) is used as the ranking criterion for BeamNJ. It is worth noting here that Neighbor-Joining (NJ) is mathematically equivalent to BeamNJ when $B = 1$.

BeamLPNJ

The optimality criterion for selecting trees at each step in the BeamLPNJ method is based on a specific underestimate of the value $\ell(T^{n-2})$ of a UBT T^{n-2} that can be derived by each tree $T^r \in \Xi^r$ considered at step r. Specifically, we define:

$$\underline{\ell}(T^{r+1}) = \hat{\ell}(T^r) + \frac{d^r_{pq}}{2} + \sum_{i\in N^{r+1}} \sum_{\substack{j\in N^{r+1}\\ j\neq i}} d^{r+1}_{ij} 2^{-\tilde{\tau}^{r+1}_{ij}} \tag{4.22}$$

where, as usual, $\{p, q\}$ is the selected pair of common ancestors, and $\tilde{\boldsymbol{\tau}}^{r+1}$ is the solution to the linear programming (LP) relaxation of Formulation (4.11) when applied to the evolutionary distance matrix D^{r+1}. Equation (4.22) is a lower bound on the value of $\ell(T^{n-2})$ and would be an ideal ranking criterion for BeamLPNJ. Unfortunately, it requires computing $\tilde{\boldsymbol{\tau}}^{r+1}$ for each tree in C^r. Instead, we propose to use an underestimate of $\underline{\ell}(T^r)$, that is based solely on the solution $\tilde{\boldsymbol{\tau}}^r$. We derive it based on the following rationale. By Theorem 1, if $\tilde{\boldsymbol{\tau}}^r = \boldsymbol{\tau}^{*r}$, then, at step r, selecting any tree $T^{r+1} \in C^r$ such that $\underline{\ell}(T^{r+1}) = \ell(T^*)$ provides the optimal selection of common ancestors to merge. In this case, it holds that

$$\begin{aligned}\underline{\ell}(T^{r+1}) = \underline{\ell}(T^r) &- \sum_{i\in N^r} d^r_{ip} 2^{-\tilde{\tau}^r_{ip}} - \sum_{i\in N^r} d^r_{iq} 2^{-\tilde{\tau}^r_{iq}} \\ &+ \sum_{i\in N^{r+1}} \frac{d^r_{ip}+d^r_{iq}}{2} 2^{-\tilde{\tau}^r_{ip}-1} + \frac{d^r_{pq}}{2} - 2d^r_{pq} 2^{-\tilde{\tau}^r_{pq}}\end{aligned} \tag{4.23}$$

which only requires $\tilde{\boldsymbol{\tau}}^r$. By (4.13), (4.23) can be simplified to

$$\begin{aligned}\underline{\ell}(T^{r+1}) &= \hat{\ell}(T^r) + \frac{d^r_{pq}}{2} + \sum_{i\in N^r}\sum_{j\in N^r} d^r_{ij} 2^{-\tilde{\tau}^r_{ij}} - d^r_{pq} 2^{-(\tilde{\tau}^r_{pq}-1)} \\ &= \underline{\ell}(T^r) + \frac{d^r_{pq}}{2} - d^r_{pq} 2^{-(\tilde{\tau}^r_{pq}-1)}.\end{aligned} \tag{4.24}$$

However, in general, $\underline{\ell}(T^r) \leq \underline{\ell}(T^{r+1}) \leq \ell(T^{n-2})$, as $\tilde{\boldsymbol{\tau}}^r \neq \tilde{\boldsymbol{\tau}}^{*r}$. This choice of ranking criterion makes BeamLPNJ with $B = 1$ a slightly different version of LPNJ where (4.16) is replaced by

$$(i, j) = \operatorname*{argmin}_{p,q} d_{pq} \left(\frac{1}{2} - \frac{1}{2^{\tilde{\tau}_{pq}-1}}\right)$$

which, like LPNJ, equally favors pairs of taxa when $\tilde{\tau}_{pq} = 2$ but introduces the value of d_{pq} in other cases.

4.4 Computational experiments

4.4.1 Data sets

We carried out computational experiments on four sets of evolutionary distance matrices. Set A consists of the nine matrices derived from real aligned DNA datasets proposed by Catanzaro et al. (2006). Set B includes the RDPII and ZILLA matrices (Stamatakis, 2005; Guindon et al., 2010) as used by Gasparin et al. (2023) in their experiments. Set C is a set of symmetric Random Doubly Stochastic Matrices (RDSM) with zero diagonals generated with the Sinkhorn-Knopp algorithm (Sinkhorn and Knopp, 1967). Set D consists of symmetric Random Integer Matrices (RIM) with entries in the range $[1 \ldots 10]$, again excluding the diagonal, which is set to zero. The instances for the experiments were derived from Sets A and B by selecting the relative distances of the subset consisting of the first n taxa of each matrix, with n ranging from 10 to 60 (or the largest multiple of 5 for matrices with fewer than 60 taxa), incremented by 5. Data sets C and D were directly generated to include ten matrices for each considered value of n.

4.4.2 Experimental setup

We compare the tree lengths produced by the proposed Beam Search algorithms with those generated by the methods implemented in the state-of-the-art tree inference software, FastME 2.0 (Lefort et al., 2015a). FastME offers five construction algorithms: NJ, BIONJ, Unweighted NJ, TaxAdd OLS, and TaxAdd BME. The trees generated by each construction method considered in the experiments are further optimized using the Subtree Pruning and Regrafting (SPR) local search method, which is also provided by FastME. In the case of BeamNJ and BeamLPNJ, the B resulting phylogenies undergo the local search procedure before returning the best one. In practice, even the linear relaxation of Formulation (4.11) can be prohibitively time-consuming to solve for medium-sized instances. To address this, we further relax the formulation by removing the Buneman conditions in constraints (4.11h)-(4.11i) and the associated y variables. Additionally, we compare two variants of BeamLPNJ, one that includes the triangular inequalities (4.11f) and (4.11g) (BeamLPNJ+tri), and one that excludes them (BeamLPNJ-Tri). All variants of the Beam Search algorithm (BeamNJ, BeamLPNJ-Tri, and BeamLPNJ+Tri) are tested for values of B in $\{1, 5, 10, 15\}$. We further accelerate the solving of the linear programs by reducing the maximum path-length from $n-1$ to $\log_2(n-1)^2$ and by adding the maximum path-length to the exponent in the objective function in order to improve the numerical stability of the solver.

4.4.3 Results

Table 4.1 shows for each data set which methods most consistently found the best tree across all considered methods and the average optimality gaps. The gap for a single instance is calculated as the difference between the length of the tree found by the method and a lower bound on that length, divided by the lower bound. The lower

		Agglomeration								**Agglomeration + SPR**							
		Set A		Set B		Set C		Set D		Set A		Set B		Set C		Set D	
Method	**B**	**Best**	**Gap**	**Best**	**Gap**	**Best**	**Gap**	**Best**	**Gap**	**Best**	**Gap**	**Best**	**Gap**	**Best**	**Gap**	**Best**	**Gap**
BeamLPNJ+Tri	15	34.3	5.43	35.2	4.2	33.6	39.17	27.3	66.3	55.6	5.07	**85.2**	4.0	**45.5**	38.47	40.0	64.0
BeamLPNJ-Tri	15	**37.4**	5.6	31.8	4.16	**39.1**	**39.06**	**34.5**	**66.19**	57.6	5.08	83.0	**3.99**	43.6	**38.4**	**42.7**	**63.71**
BeamNJ	15	**37.4**	**5.26**	**63.6**	**4.1**	36.4	39.77	27.3	67.92	54.5	5.07	68.2	4.0	40.0	38.99	24.5	65.38
BeamLPNJ+Tri	10	33.3	5.53	33.0	4.19	24.5	39.27	27.3	66.42	56.6	5.09	73.9	4.0	37.3	38.49	30.0	64.16
BeamLPNJ-Tri	10	32.3	5.77	35.2	4.18	30.0	39.23	24.5	66.31	**58.6**	**5.06**	81.8	4.0	41.8	38.43	39.1	63.92
BeamNJ	10	33.3	5.3	43.2	4.12	32.7	39.88	20.9	68.32	54.5	5.09	65.9	4.0	38.2	39.13	23.6	65.45
BeamLPNJ+Tri	5	21.2	5.62	28.4	4.29	22.7	39.33	20.9	66.72	54.5	5.08	67.0	4.01	29.1	38.63	24.5	64.53
BeamLPNJ-Tri	5	28.3	5.72	34.1	4.2	27.3	39.28	22.7	66.81	56.6	5.09	71.6	4.01	32.7	38.67	26.4	64.41
BeamNJ	5	28.3	5.35	34.1	4.12	26.4	40.37	15.5	69.05	54.5	5.08	59.1	4.01	31.8	39.27	20.0	65.88
BeamLPNJ+Tri	1	19.2	5.61	20.5	4.28	15.5	39.94	5.5	68.73	49.5	5.11	63.6	4.02	20.9	39.31	10.0	66.11
BeamLPNJ-Tri	1	22.2	5.63	23.9	4.23	12.7	39.88	8.2	68.84	52.5	5.11	61.4	4.01	18.2	39.16	14.5	66.33
BeamNJ	1	11.1	5.67	18.2	4.23	8.2	41.81	9.1	71.72	48.5	5.1	52.3	4.03	16.4	39.92	10.0	67.29
BIONJ	-	6.1	6.33	9.1	4.39	5.5	42.21	0.0	80.99	37.4	5.2	26.1	4.19	20.0	39.82	0.9	78.58
NJ	-	11.1	5.67	18.2	4.23	8.2	41.81	9.1	71.61	48.5	5.1	52.3	4.03	16.4	39.92	10.9	67.28
TaxAdd BME	-	2.0	6.99	9.1	4.49	0.9	50.03	0.0	84.31	44.4	5.16	77.3	4.01	15.5	40.29	8.2	68.03
TaxAdd OLS	-	1.0	8.01	9.1	4.78	0.9	51.25	0.9	85.66	43.4	5.15	55.7	4.03	12.7	40.52	10.9	67.95
Unweighted NJ	-	8.1	6.43	13.6	4.41	4.5	42.29	4.5	72.81	31.3	6.13	29.5	4.26	16.4	39.9	14.5	66.9

Table 4.1: Comparison of the tested methods for each data set. The *Best* columns indicate the percentage of instances in the respective data sets where the best tree across all methods was identified by the method. The *Gap* columns report the average optimality gap (in percents) over the data set. Results are presented separately for standalone agglomeration algorithms and after applying Subtree Pruning and Regrafting (SPR) optimization. Bold values highlight the best results in each column.

bound is determined by the value of the linear programming relaxation of Formulation (4.11), excluding Buneman's constraints. The detailed result data is available at `https://github.com/HenriDeh/BME_BeamSearch.git`. Several key observations can be drawn from the results. In general, Beam Search algorithms with beam width $B > 1$ show superior performance in identifying solutions that are close to optimal, consistently achieving the smallest optimality gaps on average. Among the standalone agglomeration algorithms (without SPR improvements), BeamNJ emerges as the most effective choice for biological instances (datasets A and B), showing progressively better performance as the beam width B increases. In contrast, for synthetic datasets, the BeamLPNJ-Tri method proves to be the most efficient. Notably, the inclusion of triangular inequalities in the formulation tends to hinder the performance of BeamLPNJ methods, both in terms of solution quality and computational speed. However, when $B = 1$, Beam Search algorithms simplify to standard greedy algorithms: in this scenario, BeamNJ becomes equivalent to the NJ algorithm, while BeamLPNJ closely resembles LPNJ. In such cases, the advantage of employing an LP-based algorithm becomes more evident, as solving the linear relaxation of the optimization model yields a valuable metric for guiding the pairwise joining process. We hypothesize that the relatively loose lower bound provided by estimate (4.24) is the main factor behind the declining performance of BeamLPNJ as the beam width B increases, while BeamNJ benefits from the exact tree length as its ranking criterion. When the phylogenies generated by the agglomeration methods are further refined using local SPR moves, BeamLPNJ methods slightly outperform BeamNJ. BeamLPNJ-Tri then appears to be a robust choice. Table 4.2 reports the average run times of the methods as a function of the instance sizes. Beam Search methods are, unsurprisingly, more computationaly demanding. Their run time and memory usage scale linearly in B and that of BeamLPNJ methods also increases cubically in n due to the increase in formulation size.

4.5 Final remarks and open questions

The approximation algorithms proposed in this study—based on the recent findings presented in (Catanzaro et al., 2024)—successfully deliver improved near-optimal solutions for instances of the BMEP that are otherwise impractical to tackle using the current state-of-the-art exact solution algorithms. However, since these methods rely on solving a linear program, the application of LPNJ and BeamLPNJ algorithms remains constrained to small instances. Further exploration of enhanced formulations and their mathematical properties could potentially extend the applicability of these approaches to larger problem instances. Moreover, the results presented in this chapter raise an important question: why does the inclusion of triangular inequalities in the formulation hinder the performance of BeamLPNJ algorithms? Developing tighter lower bounds for the BeamLPNJ criterion could improve the pair selection process, potentially leading to even better performance. Addressing these challenges presents an important direction for future research.

4.6 Future research

In this study, we leveraged the convenient mathematical properties of the BME criterion—stemming from the recursive nature of the implicitly induced edge weights in the UBT (see Section 1.4)—which could be further exploited to develop enhanced local search heuristics (as opposed to the constructive heuristics proposed here) and more effective branching schemes in exact solution methods.

To illustrate this, recall from Theorem 1 that if T is a UBT with n leaves and

		Instance size (n)										
Method	**B**	10	15	20	25	30	35	40	45	50	55	60
BeamLPNJ+Tri	15	0.12	0.75	2.81	7.75	18.17	37.7	74.38	139.01	251.18	391.5	636.3
BeamLPNJ-Tri	15	0.07	0.39	1.4	3.45	7.12	13.04	21.99	34.81	51.84	75.62	106.34
BeamNJ	15	< 0.01	0.01	0.03	0.05	0.08	0.13	0.2	0.34	0.49	0.65	0.84
BeamLPNJ+Tri	10	0.09	0.53	1.89	5.42	12.39	27.01	51.12	92.72	170.06	265.29	423.55
BeamLPNJ-Tri	10	0.05	0.28	0.95	2.45	5.07	9.27	15.37	24.16	36.91	53.14	74.96
BeamNJ	10	< 0.01	< 0.01	0.02	0.03	0.05	0.08	0.12	0.22	0.31	0.42	0.55
BeamLPNJ+Tri	5	0.05	0.3	1.08	2.98	7.06	13.92	26.07	47.84	84.31	125.41	206.35
BeamLPNJ-Tri	5	0.03	0.17	0.64	1.42	3.02	5.39	9.1	14.45	21.81	30.6	43.0
BeamNJ	5	< 0.01	< 0.01	< 0.01	0.02	0.03	0.04	0.06	0.11	0.16	0.2	0.26
BeamLPNJ+Tri	1	0.02	0.12	0.42	1.0	2.11	4.13	7.05	11.25	19.03	25.64	38.1
BeamLPNJ-Tri	1	0.02	0.08	0.25	0.65	1.27	2.33	3.78	6.09	9.05	13.62	18.79
BeamNJ	1	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	0.01	0.02	0.03	0.04	0.05
BIONJ	-	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	0.01	0.01
NJ	-	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	0.01	0.01
TaxAdd BME	-	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	0.01
TaxAdd OLS	-	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	0.01
Unweighted NJ	-	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	< 0.01	0.01	0.01

Table 4.2: Average run time of each method in seconds per instance size (number of taxa).

its associated distance matrix $\boldsymbol{D}$, and $\check{T}$ is the UBT with $n-k$ leaves and distance matrix $\check{\boldsymbol{\tau}}$ obtained by iteratively collasping k cherries into their common ancestor via rule (4.13) (so that $\check{T}$ is a subtree of T). Then, optimizing the instance of the BMEP induced by the $(n-k) \times (n-k)$ $\check{\boldsymbol{D}}$ matrix amounts to reoptimizing the subtree $\check{T}$ while keeping the remainder of T unchanged. By Theorem (1), any reduction in $\ell(\check{T})$ yields an identical improvement in $\ell(T)$. Consequently, one could implement a local search scheme that, starting from a UBT T, repeatedly samples a subtree $\check{T}$, refines it by solving the BMEP for $\check{D}$, and then applies a simple local search——such as SPR moves——on the updated tree. Our preliminary exploration of this idea suggests it could extend the local methods in FastME 2.0 (Lefort et al., 2015b) via a large-neighborhood search framework (Pisinger and Ropke, 2018).

As noted in Section 4.3.2, if the minimal entries of $\tilde{\boldsymbol{\tau}}$ were correct at each iteration, i.e. $\tilde{\tau}_{ij} = 2 \iff \tau^*_{ij} = 2$, then the LPNJ algorithm returns the optimal UBT. The Beam Search extension we discussed in this chapter can be viewed as a non-exhaustive breadth-first search algorithm that branches on $x^2_{ij} = 1$. Given the heuristic's strong performance, it is natural to ask whether a specialized Branch & Bound/Cut algorithm designed along the same principles as our Beam Search procedure could improve current state-of-the-art exact methods by replacing the use of generic commercial solver.

Acknowledgements

The first author acknowledges support from the Belgian National Fund for Scientific Research (FNRS) via the grant FNRS PDR 40007831, and the Fondation Louvain via the grant "COALESCENS". Part of this research has been carried out during the research visit of D. Catanzaro at the Venice School of Management of the University Ca Foscari of Venice, Italy, Spring 2024.

Chapter

FIVE

CONCLUSION

This thesis brings together three studies that, while focused on distinct domains (inventory management, energy infrastructure planning, and phylogenetic inference), are unified by their reliance on optimization as a central methodological tool. In what follows, we summarize the contributions of each chapter and reflect on their broader implications.

In Chapter 2, we applied Deep Reinforcement Learning (DRL) to classical lot-sizing problems, showing how modern machine-learning algorithms can rediscover optimal dynamic inventory-replenishment policies. By choosing a problem whose properties have been analyzed for decades, we focused on methodological guidance for embedding non-stationary demand and continuous action spaces into a DRL framework. We began by benchmarking the DRL approach on a simplified lot-sizing instance, which exposed critical training failures tied to the value function's discontinuous shape. Understanding this connection led us to design tailored solutions that stabilized learning and enabled the neural network to achieve near-optimal policies. Three key methodological takeaways emerged: (1) the importance of curriculum training design to avoid early, catastrophic suboptimal convergence of the actor network; (2) the need to correctly decouple the discrete and continuous dimensions of the action space to overcome value-function discontinuities; and (3) the benefit of augmenting state inputs with forecast statistics coupled with randomized training instances to avoid inadvertent learning of post-horizon demand information. These results emphasize the necessity of avoiding a blind application of DRL algorithms to OR without a proper understanding of the problem at hand. With confidence in its robustness, we then explored scenarios in which more difficult versions of the problem were better solved by DRL than by simpler heuristics. We hope these insights will broaden DRL's applicability in OR, for example, to multi-echelon lot sizing, by offering generalizable best practices for algorithmic implementations.

In Chapter 3, we studied the long-term deployment strategy for Floating Storage and Regasification Units by Germany following the disruption of Russian natural gas supplies in the context of the conflict in Ukraine. The core contribution of this study is to show how optimization can support large-scale policy analysis under several alternative geopolitical and energy-transition scenarios. This was achieved by embedding the strategic capacity expansion and facility location problem into a flow model, allowing us to incorporate the existing pipeline network topology as a structural constraint influencing decision-making. The resulting analysis enabled us to provide quantitative insights on the alignment of Germany's infrastructure investment plan with the European Green Deal objectives. By benchmarking the model's recommendations against the government's actual choices, we show how OR can inform and critique public policymaking. Importantly, while the model captures technical and economic constraints, it cannot fully reproduce the rationale of real-world decision makers, whose choices may be shaped by political, social, or strategic considerations that are not quantifiable or even publicly known. Certain emerging elements, such as the potential future role of hydrogen infrastructure, are excluded from the model due to the current lack of reliable data or mature planning frameworks. Their exclusion reflects a trade-off between speculative modeling and analytical robustness. These limitations highlight the interpretive boundary of the model: its results should not be read as predictive, but as structured insights conditional on clearly defined assumptions.

In Chapter 4, we moved from the optimization of decisions taken in the context of human activities (lot-sizing and energy policy) to the modeling of a purely biological process. Specifically, we cast the task of phylogeny inference under the Balanced Minimum Evolution (BME) criterion as a network design optimization problem. We employed the well-known OR tool of linear programming to design a matheuristic (LPNJ) that provides an alternative agglomeration method to the ubiquitous Neighbor Joining (NJ) algorithm. The use of linear programming ensures that, similar to NJ, each step of the LPNJ tree construction is based on a principled decision guided by a model that explicitly aims to minimize the BME criterion. We then proposed to extend the search space explored by the NJ and LPNJ methods by embedding them into a beam search. Our experimental results show that using beam search is significantly more likely to yield improved trees compared to classical approaches in the bioinformatics literature.

Despite their drastically different contributions, whether in terms of methodology, algorithms, or applications, these three chapters are more deeply linked than they initially appear. Obviously, they each involve optimization, albeit with varying purposes. Interestingly, although it pertains to phylogeny, an unusual topic in OR, Chapter 4 employs optimization in the most traditional way: a task is abstracted as an optimization problem and tools from the OR toolbox are used to attempt to obtain solutions of the best possible quality in a reasonable time. Conversely, in Chapter 3, while the algorithm used is perfectly standard, optimization is employed not as a goal but as a means to an end, that of reproducing the behavior of decision makers. Notably, in Chapter 2, optimization appears both as the means and the end. While the purpose of the study was to investigate a new approach to inventory control, the goal ultimately remains to solve a lot-sizing problem. However, in Reinforcement Learning, this is achieved indirectly by optimizing the weights of a policy neural network whose objective is to solve the considered lot-sizing problem, by minimizing a surrogate loss given by the critic neural network.

Furthermore, the three studies also share the issue of scalability, probably the most common problem in OR in general. Again, the issue appears in a different form each time. In Chapter 4, we face the usual issue of a rapidly increasing size of the mathematical program, here due to the cubic increase in the number of constraints and variables. In Chapter 2, scalability concerns arise this time not due to the size of the decision space, but from that of the state space. Because we specifically dealt with the case of non-stationary demand, and because an infinite number of realizations are possible for each forecast, every episode generates a drastically different scenario. The DRL algorithm must sparingly decide whether to exploit its current policy to reinforce value-approximation or to explore more solutions. This problem is well known as the exploration vs. exploitation problem. Unsurprisingly, this challenge is exacerbated when scaling the inventory control to multiple items due to the introduction of a new forecast for each item. It is therefore important that future approaches aiming to tackle that problem take good care of designing algorithms able to take advantage of the symmetries that arise in multi-item environments. Future research aiming to build upon the results presented in Chapter 3 will also need to deal with scalability. One main limitation of the study is that uncertainty in the future, an important driver in policy decision making, is not directly modeled. This limitation arises from the high complexity introduced by uncertainty in mathematical programming, both in terms of modeling and computational effort.

The reproducibility of published studies has become an increasingly important concern in the OR and scientific communities. While the seminal articles that raised the global issue of reproducibility in science are mainly discussing unreproducible results in medical and psychological sciences (Ioannidis, 2005; Open Science Collaboration, 2015), concerns were also raised in OR and management sciences (Fišar et al., 2023). We paid particular attention to providing fully reproducible results in each of our studies. To that end, all experiments were carried out with the Julia programming language (Bezanson et al., 2017), which natively provides working environments that a researcher interested in reproducing the study can instantiate to automatically download libraries in the same versions as those used for the original experiments.

However, perfectly reproducible experiments do not guarantee perfect transparency of the results. It is also necessary to also design the experiments and the analysis of the results such that they are interpretable. Once again this took a different form in each study contained in this thesis. In Chapter 2, interpretability is one of the main concerns surrounding the use of DRL in operations management. Indeed, because the most common algorithms in DRL are model-free, as is the case for the Proximal Policy Optimization (PPO) algorithm that we employed here, the trained neural network is essentially a black-box approximation of the policy whose internal complexity is hard to decipher. We took special care to analyze the behavior of the network in Section 2.5.5; this notably ensured that the learned policy indeed converged to a (s, S)-type of policy, rather than a simpler policy performs adequately, thereby showing the ability of the approach to truly tackle complex problems. In Chapter 3, interpretability concerns revolved mainly around the interpretation of the results. We had to carefully answer the question "what can or cannot we conclude from the results?" Correctly answering it is a fundamental aspect of the scientific method. Here, for example, we drew the conclusion that Germany's short-term LNG capacity expansion plan is not an overreaction in terms of supply security, even when assuming a steady decarbonation of the economy. Conversely, as pointed out in Section 3.7, due to limitations in the scope of the model, several aspects, such as a potential conversion of the distribution network to support hydrogen, or the risk analysis driver in decision-making, were left out and thus limit the conclusions that can be drawn from the study. Interpretability is also a concern on the algorithm design side. A few years ago, the field saw an increasing number of new metaheuristics mimicking natural behaviors such as those of animals, social groups or physical and biological systems, being published in the literature. As exposed by Sörensen (2015), while many were perfectly valid contributions, plenty also turned out to be merely renamings of existing methods. This work highlighted the importance of interpretable algorithm design through formal analysis. We followed that recommendation by focusing on designing algorithms that provably behave in an understandable way by proving that they directly optimize the desired objective. While metaheuristics are absolutely valuable, here, we took a particular care in proposing algorithms whose search of the solution space is understandable.

Ultimately, the studies presented in this thesis illustrate that effective OR, beyond offering a vast set of tools suited for a plethora of applications, also requires not only a rigorous scientific methodology but also a careful consideration of modeling assumptions, computational constraints, and interpretability of both algorithms and results. These important aspects of the research can take drastically different forms depending on the methods used and the application of interest. Correctly identifying them represents one of the core competencies an OR practitioner can develop.

BIBLIOGRAPHY

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X., 2015. TensorFlow: Large-scale machine learning on heterogeneous systems. URL: `https://www.tensorflow.org/`. software available from tensorflow.org.

Abrell, J., Chavaz, L., Weigt, H., 2016. Pathways for the european natural gas market, in: 2016 13th International Conference on the European Energy Market (EEM), pp. 1–6. doi:`10.1109/EEM.2016.7521360`.

ACER, 2023. Acer price assessments. URL: `https://aegis.acer.europa.eu/terminal/price_assessments`.

Agrell, P.J., Bogetoft, P., Gautier, A., 2024. Model Specification GAS, CEER Project TCB21. Technical Report. Sumicsid. URL: `www.sumicsid.com`.

Agrell, P.J., Bogetoft, P., Gautier, A., Petrov, K., 2023. Note on Environmental Modelling, CEER Project TCB21. Technical Report. Sumicsid. URL: `www.sumicsid.com`.

Ah-Voun, D., Chyong, C.K., Li, C., 2024. Europe's energy security: From russian dependence to renewable reliance. Energy Policy 184. doi:`10.1016/j.enpol.2023.113856`.

Akkaya, I., Andrychowicz, M., Chociej, M., Litwin, M., McGrew, B., Petron, A., Paino, A., Plappert, M., Powell, G., Ribas, R., et al., 2019. Solving Rubik's cube with a robot hand. arXiv preprint arXiv:1910.07113 .

Andrychowicz, M., Raichuk, A., Stańczyk, P., Orsini, M., Girgin, S., Marinier, R., Hussenot, L., Geist, M., Pietquin, O., Michalski, M., Gelly, S., Bachem, O., 2021. What matters for on-policy deep actor-critic methods? A large-scale study, in: International Conference on Learning Representations.

Askin, R.G., 1981. A procedure for production lot sizing with probabilistic dynamic demand. AIIE Transactions 13, 132–137.

Axsäter, S., 2015. Inventory Control. volume 225. Springer.

Beerenwinkel, N., Schwarz, R.F., Gerstung, M., Markowetz, F., 2015. Cancer evolution: mathematical models and computational inference. Systematic biology 64, e1–e25.

Bellemare, M.G., Dabney, W., Rowland, M., 2023. Distributional Reinforcement Learning. MIT Press. `http://www.distributional-rl.org`.

Bezanson, J., Edelman, A., Karpinski, S., Shah, V.B., 2017. Julia: A fresh approach to numerical computing. SIAM Review 59, 65–98.

BMWK, 2020. The National Hydrogen Strategy. Technical Report. Bundesministerium für Wirtschaft und Klimaschutz. URL: `https://www.bmwk.de/Redaktion/EN/Publikationen/Energie/the-national-hydrogen-strategy.pdf`.

BMWK, 2023a. Bericht des Bundeswirtschafts- und Klimaschutzministeriums zu Planungen und Kapazitäten der schwimmenden und festen Flüssigerdgasterminals. Technical Report. Bundesministerium für Wirtschaft und Klimaschutz. URL: `https://www.bmwk.de/Redaktion/DE/Pressemitteilungen/2023/03/20230303-bmwk-legt-bericht-zu-planungen-und-kapazitaeten-der-schwimmenden-und-feste html`.

BMWK, 2023b. National Hydrogen Strategy: Update. Technical Report. Bundesministerium für Wirtschaft und Klimaschutz. URL: `https://www.bmwk.de/Redaktion/EN/Publikationen/Energie/national-hydrogen-strategy-update.pdf`.

BMWK, 2024a. Import Strategy for hydrogen and hydrogen derivatives. Technical Report. Bundesministerium für Wirtschaft und Klimaschutz. URL: `https://www.bmwk.de/Redaktion/EN/Publikationen/Energie/importstrategy-hydrogen.pdf`.

BMWK, 2024b. Wasserstoffbeschleunigungsgesetz. Technical Report. Bundesministerium für Wirtschaft und Klimaschutz. URL: `https://www.bmwk.de/Redaktion/DE/Downloads/W/wasserstoffbeschleunigungsgesetz.pdf`.

Bollapragada, S., Morton, T.E., 1999. A simple heuristic for computing nonstationary (s, S) policies. Operations Research 47, 576–584.

Bookbinder, J.H., Tan, J.Y., 1988. Strategies for the probabilistic lot-sizing problem with service-level constraints. Management Science 34, 1096–1108.

Boute, R.N., Gijsbrechts, J., van Jaarsveld, W., Vanvuchelen, N., 2021. Deep reinforcement learning for inventory control: A roadmap. European Journal of Operational Research In-press. doi:`https://doi.org/10.1016/j.ejor.2021.07.016`.

Brauers, H., Braunger, I., Jewell, J., 2021. Liquefied natural gas expansion plans in Germany: The risk of gas lock-in under energy transitions. Energy Research and Social Science 76. URL: `https://doi.org/10.1016/j.erss.2021.102059`, doi:`10.1016/j.erss.2021.102059`.

Bundesnetzagentur, 2023. Aktuelle Lage Gasversorgung - Gasimporte in GWh/Tag. URL: `https://www.bundesnetzagentur.de/DE/Gasversorgung/aktuelle_gasversorgung/_svg/Gasimporte/Gasimporte.html?nn=1077982`.

Buneman, P., 1974. A note on the metric properties of trees. Journal of combinatorial theory 17, 48–50.

Catanzaro, D., Frohn, M., Gascuel, O., Pesenti, R., 2022. A tutorial on the balanced minimum evolution problem. European Journal of Operational Research 300, 1–19. URL: `https://doi.org/10.1016/j.ejor.2021.08.004`, doi:`10.1016/j.ejor.2021.08.004`.

Catanzaro, D., Frohn, M., Pesenti, R., 2020a. An information theory perspective on the balanced minimum evolution problem. Operations Research Letters 48, 362–367.

Catanzaro, D., Pesenti, R., Milinkovitch, M., 2006. A non-linear optimization procedure to estimate distances and instantaneous substitution rate matrices under the GTR model. Bioinformatics 22, 708–715.

Catanzaro, D., Pesenti, R., Sapucaia, A., Wolsey, L., 2024. Optimizing over path-length matrices of unrooted binary trees. Mathematical Programming, under review (available at `https://www.danielecatanzaro.com/wp-content/uploads/2024/06/CPSW.pdf`) .

Catanzaro, D., Pesenti, R., Wolsey, L.A., 2020b. On the Balanced Minimum Evolution polytope. Discrete Optimization 36, 1–33.

Cavalli-Sforza, L.L., Edwards, A.W., 1967. Phylogenetic analysis. models and estimation procedures. American journal of human genetics 19, 233.

Chyong, C.K., Henderson, J., 2024. Quantifying the economic value of russian gas in europe in the aftermath of the 2022 war in ukraine. Energy 292. doi:`10.1016/j.energy.2024.130604`.

Clark, A.J., Scarf, H., 1960. Optimal policies for a multi-echelon inventory problem. Management Science 6, 475–490.

Cobbe, K.W., Hilton, J., Klimov, O., Schulman, J., 2021. Phasic policy gradient, in: Proceedings of the 38th International Conference on Machine Learning, PMLR. pp. 2020–2027.

Cueto, M.A., Matsen, F.A., 2011. Polyhedral geometry of phylogenetic rogue taxa. Bulletin of Mathematical Biology 73, 1202–1226.

David, S., Max, G., Michael, M., Jan, K., Jakob, J., Hendrik, D., Meike, V., 2023. Analysis of the global gas markets until 2035. Technical Report. EWI, Institute of Energy Economics, University of Cologne. URL: `https://www.ewi.uni-koeln.de/en/publications/analysis-of-the-global-gas-markets-until-2035/`.

De Moor, B.J., Gijsbrechts, J., Boute, R.N., 2022. Reward shaping to improve the performance of deep reinforcement learning in perishable inventory management. European Journal of Operational Research 301, 535–545.

De Munck, T., Tancrez, J.S., Chevalier, P., 2025. Transfer Reinforcement Learning for Pricing, Driver Repositioning and Customer Admission in Ride-Hailing Networks. Technical Report.

Derse, O., Göçmen, E., Yılmaz, E., Erol, R., 2020. A mathematical programming model for facility location optimization of hydrogen production from renewable energy sources. Energy Sources, Part A: Recovery, Utilization and Environmental Effects 44, 6648–6659. URL: https://www.tandfonline.com/doi/abs/10.1080/15567036.2020.1812769, doi:10.1080/15567036.2020.1812769.

Desper, R., Gascuel, O., 2004. Theoretical foundations of the balanced minimum evolution method of phylogenetic inference and its relationship to the weighted least-squares tree fitting. Molecular Biology and Evolution 21, 587–598.

Diettrich, J., Pluta, A., Medjroubi, W., 2021. SciGRID gas IGGIELGN Data Model of the European Gas Transport Network. Technical Report. Zenodo. URL: https://zenodo.org/records/4767098, doi:10.5281/ZENODO.4767098.

Dong, L., Lee, H.L., 2003. Optimal policies and approximations for a serial multi-echelon inventory system with time-correlated demand. Operations Research 51, 969–980.

Dowson, O., Kapelevich, L., 2021. SDDP.jl: a Julia package for stochastic dual dynamic programming. INFORMS Journal on Computing 33, 27–33. doi:https://doi.org/10.1287/ijoc.2020.0987.

Duellman, W.E., Marion, A.B., Hedges, S.B., 2016. Phylogenetics, classification, and biogeography of the treefrogs (amphibia: Anura: Arboranae). Zootaxa 4104, 1–109.

Durakovic, G., del Granado, P.C., Tomasgard, A., 2023. Are green and blue hydrogen competitive or complementary? Insights from a decarbonized European power system analysis. Energy 282, 128282. doi:10.1016/J.ENERGY.2023.128282.

Dural-Selcuk, G., Rossi, R., Kilic, O.A., Tarim, S.A., 2020. The benefit of receding horizon control: Near-optimal policies for stochastic inventory control. Omega 97, 102091.

E-Control, 2024. Entwicklung der großhandelspreise. URL: https://www.e-control.at/industrie/gas/gaspreis/grosshandelspreise.

ECOnnect, 2024. FSRU Wilhelmshaven and ECOnnect have initiated installation of DET'S 2nd Terminal in Wilhelmshaven. URL: https://www.econnectenergy.com/articles/fsru-wilhelmshaven-and-econnect-have-initiated-installation-of-dets-2nd-terminal-in

EIA, 2023. Henry Hub Natural Gas Spot Price (Dollars per Million Btu). URL: https://www.eia.gov/dnav/ng/hist/rngwhhdm.htm.

Emiliozzi, S., Ferriani, F., Gazzani, A., 2023. The European Energy Crisis and the Consequences for the Global Natural Gas Market. Bank of Italy Occasional paper URL: https://www.bancaditalia.it/pubblicazioni/qef/2023-0824/QEF_824_23.pdf, doi:10.2139/ssrn.4849472.

Enerdata, 2023. Germany's gas consumption and imports declined in 2022 — Enerdata. URL: https://www.enerdata.net/publications/daily-energy-news/germanys-gas-consumption-and-imports-declined-2022.html.

Energy, O., 2024. Germany's Mukran LNG terminal starts trial operations - Offshore Energy. URL: https://www.offshore-energy.biz/germanys-mukran-lng-terminal-starts-trial-operations/.

Engstrom, L., Ilyas, A., Santurkar, S., Tsipras, D., Janoos, F., Rudolph, L., Madry, A., 2020. Implementation matters in deep rl: A case study on ppo and trpo, in: International Conference on Learning Representations.

ENTSOG, 2024. The european network of transmission system operators for gas, members. URL: https://www.entsog.eu/members.

Eser, P., Chokani, N., Abhari, R., 2019. Impact of Nord Stream 2 and LNG on gas trade and security of supply in the European gas network of 2030. Applied Energy 238, 816–830. doi:10.1016/J.APENERGY.2019.01.068.

European Commission, ., 2019. Communication From the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions. The European Green Deal. EUR-Lex , 1–43URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=COM%3A2019%3A640%3AFIN.

European Commission, ., 2022. Communication from the Commission to the European Parliament, the European Council, the Council, the European Economic and Social Committee and the Committee of the Regions. REPowerEU Plan. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=COM%3A2022%3A230%3AFIN&qid=1653033742483https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=COM%3A2022%3A230%3AFIN.

European Commission, 2023. Report from The Commission to The European Parliament, The Council, The European Economic and Social Committe and The Committe of the Regions. State of the Energy Union Report 2023. Technical Report. European Commission. URL: https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52023DC0650.

European Parliament, 2018. Regulation (EU) 2018/1999 of the European Parliament and of the Council of 11 December 2018 on the Governance of the Energy Union and Climate Action. URL: https://eur-lex.europa.eu/eli/reg/2018/1999/oj.

Eurostat, 2023a. Population structure indicators by nuts 2 region. URL: https://ec.europa.eu/eurostat/databrowser/view/demo_r_pjanind2/default/table?lang=en.

Eurostat, 2023b. Regional gross domestic product (pps per inhabitant) by nuts 2 regions. URL: https://ec.europa.eu/eurostat/databrowser/view/TGS00005/default/table?lang=en&category=na10.nama10.nama_10reg.nama_10r_gdp.

Federal Statistical Office Germany, 2024. Gas production, network injection, own consumption, storage level: Germany. URL: https://www-genesis.destatis.de/

genesis/online?operation=abruftabelleBearbeiten&levelindex=2&levelid=1727270050497&auswahloperation=abruftabelleAuspraegungAuswaehlen&auswahlverzeichnis=ordnungsstruktur&auswahlziel=werteabruf&code=43321-0001&auswahltext=&werteabruf=Value+retrieval#abreadcrumb.

Felsenstein, J., 1978. Cases in which parsimony or compatibility methods will be positively misleading. Systematic zoology 27, 401–410.

Felsenstein, J., 2004a. Inferring Phylogenies. Sinauer Associates, Sunderland, Mass.

Felsenstein, J., 2004b. Inferring phylogenies. Sinauer Associates, Sunderland, MA.

Fiorini, S., Joret, G., 2012. Approximating the balanced minimum evolution problem. Operations Research Letters 40, 31–35.

Fitch, W.M., 1971. Toward defining the course of evolution: minimum change for a specific tree topology. Systematic Biology 20, 406–416.

Fišar, M., Greiner, B., Huber, C., Katok, E., Ozkes, A.I., 2023. Reproducibility in management science. Management Science 70, 1343–1356. URL: https://doi.org/10.1287/mnsc.2023.03556, doi:10.1287/mnsc.2023.03556.

Floyd, R.W., 1962. Algorithm 97: Shortest path. Communications of the ACM 5, 345. URL: https://dl.acm.org/doi/10.1145/367766.368168, doi:10.1145/367766.368168.

Forcey, S., Keefe, L., Sands, W., 2016. Facets of the balanced minimal evolution polytope. Mathematical Biology 73, 447–468.

Forcey, S., Keefe, L., Sands, W., 2017. Split-facets for balanced minimal evolution polytopes and the permutoassociahedron. Bulletin of Mathematical Biology, in press .

Frohn, M., 2021. On the approximability of the fixed-tree balanced minimum evolution problem. Optimization Letters 15, 2321–2329. URL: https://doi.org/10.1007/s11590-020-01677-x, doi:10.1007/s11590-020-01677-x.

Gascuel, O., 2005. Mathematics of evolution and phylogeny. Oxford University Press, New York.

Gascuel, O., Steel, M., 2006. Neighbor-joining revealed. Molecular Biology and Evolution 23, 1997–2000.

Gasparin, A., Verdù, F.J.C., Catanzaro, D., Castelli, L., 2023. An evolution strategy approach for the balanced minimum evolution problem. Bioinformatics 39, btad660.

Gijsbrechts, J., Boute, R.N., Van Mieghem, J.A., Zhang, D.J., 2022. Can deep reinforcement learning improve inventory management? performance on lost sales, dual-sourcing, and multi-echelon problems. Manufacturing & Service Operations Management 24, 1349–1368. doi:https://doi.org/10.1287/msom.2021.1064.

Global-LNG, 2024. Fid nears for german lng terminals. URL: https://globallnghub.com/fid-nears-for-german-lng-terminals.html.

Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y., 2016. Deep learning. volume 1. MIT press Cambridge.

Graves, S.C., 1999. A single-item inventory model for a nonstationary demand process. Manufacturing & Service Operations Management 1, 50–61.

Graves, S.C., Meal, H.C., Dasu, S., Qui, Y., 1986. Two-stage production planning in a dynamic environment, in: Multi-stage production planning and inventory control. Springer, pp. 9–43.

Gritz, A., Wolff, G., 2024. Gas and energy security in Germany and central and Eastern Europe. Energy Policy 184, 113885. URL: https://www.sciencedirect.com/science/article/pii/S0301421523004706?casa_token=EOXBvB8ygzOAAAAA:lb3FXeErk2ndIO2UCBemw7VemV2ymbJPiNXjPCWjC99Ni1S2lMfjf1OZl9B8UTUfB0p4sdA3cQhttps://linkinghub.elsevier.com/retrieve/pii/S0301421523004706, doi:10.1016/j.enpol.2023.113885.

Guindon, S., Dufayard, J.F., Lefort, V., Anisimova, M., Hordijk, W., Gascuel, O., 2010. New Algorithms and Methods to Estimate Maximum-Likelihood Phylogenies: Assessing the Performance of PhyML 3.0. Systematic Biology 59, 307–321. doi:10.1093/sysbio/syq010.

Gürsan, C., de Gooyert, V., 2021. The systemic impact of a transition fuel: Does natural gas help or hinder the energy transition? Renewable and Sustainable Energy Reviews 138. doi:10.1016/j.rser.2020.110552.

Haarnoja, T., Zhou, A., Abbeel, P., Levine, S., 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, in: Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden. pp. 1861–1870.

Hadfield, J., Megill, C., Bell, S.M., Huddleston, J., Potter, B., Callender, C., Sagulenko, P., Bedford, T., Neher, R.A., 2018. Nextstrain: real-time tracking of pathogen evolution. Bioinformatics 34, 4121–4123.

Halser, C., Paraschiv, F., 2022. Pathways to Overcoming Natural Gas Dependency on Russia—The German Case. Energies 15. URL: https://doi.org/10.3390/en15144939, doi:10.3390/en15144939.

Hamedi, M., Zanjirani Farahani, R., Husseini, M.M., Esmaeilian, G.R., 2009. A distribution planning model for natural gas supply chain: A case study. Energy Policy 37, 799–812. doi:10.1016/j.enpol.2008.10.030.

Haws, D.C., Hodge, T.L., Yoshida, R., 2011. Optimality of the neighbor joining algorithm and faces of the balanced minimum evolution polytope. Bulletin of Mathematical Biology 73, 2627–2648.

Heath, D.C., Jackson, P.L., 1994. Modeling the evolution of demand forecasts with application to safety stock analysis in production/distribution systems. IIE transactions 26, 17–30.

Heather, P., 2024. European Traded Gas Hubs: the markets have rebalanced. Technical Report. The Oxford Institute for Energy Studies. URL: https://www.oxfordenergy.org/publications/european-traded-gas-hubs-the-markets-have-rebalanced/.

Hendrycks, D., Gimpel, K., 2016. Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415 .

van Hezewijk, L., Dellaert, N., Woensel, T.V., Gademann, N., 2022. Using the proximal policy optimisation algorithm for solving the stochastic capacitated lot sizing problem. International Journal of Production Research 0, 1–24. doi:10.1080/00207543.2022.2056540.

Hill, R.M., Johansen, S.G., 2006. Optimal and near-optimal policies for lost sales inventory models with at most one replenishment order outstanding. European Journal of Operational Research 169, 111–132.

Hornik, K., Stinchcombe, M., White, H., 1989. Multilayer feedforward networks are universal approximators. Neural networks 2, 359–366.

Huangfu, Q., Hall, J.A.J., 2018. Parallelizing the dual revised simplex method. Math. Prog. Comp 10, 119–142. URL: https://doi.org/10.1007/s12532-017-0130-5, doi:10.1007/s12532-017-0130-5.

IEA, 2021. Gas market report Q1 2021. URL: https://www.iea.org/reports/gas-market-report-q1-2021.

IEA, 2023. International Energy Agency. URL: https://www.iea.org/data-and-statistics.

Iglehart, D.L., 1963. Optimality of (s, S) policies in the infinite horizon dynamic inventory problem. Management Science 9, 259–267.

Iida, T., Zipkin, P.H., 2006. Approximate solutions of a dynamic forecast-inventory model. Manufacturing & Service Operations Management 8, 407–425.

Innes, M., 2018. Flux: Elegant machine learning with julia. Journal of Open Source Software 3, 602. doi:10.21105/joss.00602.

Ioannidis, J.P.A., 2005. Why most published research findings are false. PLoS Medicine 2, e124. URL: https://doi.org/10.1371/journal.pmed.0020124, doi:10.1371/journal.pmed.0020124.

Jaller-Makarewicz, A.M., 2024. European LNG Tracker. URL: https://ieefa.org/european-lng-tracker-september-2024-update.

Jirušek, M., 2022. Step Up or Perish: The EU's Struggle With Russian Energy Imports and Its Role in the New Geopolitical Reality. Russian Analytical Digest 284, 5–8. URL: https://www.ssoar.info/ssoar/handle/document/86981www.css.ethz.ch/en/publications/rad.html, doi:10.3929/ethz-b-000550755.

Karlin, S., Scarf, H., 1958. Inventory models of the Arrow-Harris-Marschak type with time lag. Stanford University Press. pp. 155–178.

Kędzierski, M., 2023. Germany: how the gas sector changed in the crisis year of 2022. URL: www.osw.waw.plwww.osw.waw.plhttps://www.ceeol.com/search/gray-literature-detail?id=1163179.

Khatiwada, D., Vasudevan, R.A., Santos, B.H., 2022. Decarbonization of natural gas systems in the EU – Costs, barriers, and constraints of hydrogen production with a case study in Portugal. Renewable and Sustainable Energy Reviews 168, 112775. doi:10.1016/j.rser.2022.112775.

Khoiriyah, L., Buda Artana, K., Dinariyana, A.A.B., Turbaningsih, O., Nur, W., Nisa, H., Mustakim, A., 2023. Conceptual design of floating storage and regasification unit (FSRU) for Eastern Part Indonesia. IOP Conference Series: Earth and Environmental Science 1239, 012015. URL: https://iopscience.iop.org/article/10.1088/1755-1315/1239/1/012015, doi:10.1088/1755-1315/1239/1/012015.

Kidd, K.K., Sgaramella-Zonta, L.A., 1971. Phylogenetic analysis: concepts and methods. American journal of human genetics 23, 235.

Kingma, D.P., Ba, J., 2015. Adam: A method for stochastic optimization, in: Proceedings of the 3rd International Conference for Learning Representations, San Diego, California.

Kuzemko, C., Blondeel, M., Dupont, C., Brisbois, M.C., 2022. Russia's war on Ukraine, European energy policy responses & implications for sustainable transformations. doi:10.1016/j.erss.2022.102842.

Lefort, V., Desper, R., Gascuel, O., 2015a. FastME 2.0: A comprehensive, accurate, and fast distance-based phylogeny inference program. Molecular Biology and Evolution 32, 2798–2800.

Lefort, V., Desper, R., Gascuel, O., 2015b. FastME 2.0: A Comprehensive, Accurate, and Fast Distance-Based Phylogeny Inference Program: Table 1. Molecular Biology and Evolution 32, 2798–2800. doi:10.1093/molbev/msv150.

Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D., 2016. Continuous control with deep reinforcement learning, in: Proceedings of the 4th International Conference on Learning Representations, San Juan, Puerto Rico.

LNG Prime, 2024. German FSRU terminal operator offers regas capacity for first time - LNG Prime. URL: https://lngprime.com/europe/german-fsru-terminal-operator-offers-regas-capacity-for-first-time/92348/.

Lowerre, B.T., 1976. The Harpy Speech Recognition System. Phd dissertation. Carnegie Mellon University.

Masudin, I., 2013. Facility Location Modeling in Multi-Echelon Distribution System: A Case Study of Indonesian Liquefied Petroleum Gas Supply Chain. Aceh International Journal of Science and Technology 2, 37–43. URL: https://jurnal.usk.ac.id/AIJST/article/view/595, doi:10.13170/aijst.2.1.595.

Melkote, S., Daskin, M.S., 2001. Capacitated facility location/network design problems. European Journal of Operational Research 129, 481–495. doi:10.1016/S0377-2217(99)00464-6.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al., 2015. Human-level control through deep reinforcement learning. Nature 518, 529.

Molnar, G., 2022. Economics of gas transportation by pipeline and LNG, in: The Palgrave handbook of international energy economics. Springer International Publishing Cham, pp. 23–57.

Nahmias, S., 1979. Simple approximations for a variety of dynamic leadtime lost-sales inventory models. Operations Research 27, 904–924.

Nextstrain, 2025. Genomic epidemiology of sars-cov-2 with subsampling focused globally over the past 6 months. https://nextstrain.org/ncov/gisaid/global/6m?l=unrooted. Accessed: 2025-05-12.

Norouzi, A., Uzsoy, R., 2014. Modeling the evolution of dependency between demands, with application to inventory planning. IIE Transactions 46, 55–66.

Open Science Collaboration, 2015. Estimating the reproducibility of psychological science. Science 349. doi:10.1126/science.aac4716.

OpenAI, 2022. Introducing chatgpt. https://openai.com/index/chatgpt/. Published November 30, 2022; accessed April 30, 2025.

Oroojlooyjadid, A., Nazari, M., Snyder, L., Takáč, M., 2021. A deep q-network for the beer game: Deep reinforcement learning for inventory optimization. Manufacturing and Service Operations Management .

Overland, I., 2022. Russia's invasion of Ukraine: Consequences for global decarbonization. Russian Analytical Digest 284, 2–5. URL: https://www.ssoar.info/ssoar/handle/document/86980https://doi.org/10.3929/ethz-b-000550755.

Pardi, F., 2009. Algorithms on Phylogenetic Trees. Ph.D. thesis. University of Cambridge, UK.

Pardi, F., Guillemot, S., Gascuel, O., 2010. Robustness of phylogenetic inference based on minimum evolution. Bulletin of Mathematical Biology 72, 1820–1839.

Paszke, A., 2019. Pytorch: An imperative style, high-performance deep learning library. arXiv preprint arXiv:1912.01703 .

Pauplin, Y., 2000. Direct calculation of a tree length using a distance matrix. Journal of Molecular Evolution 51, 41–47.

Pisinger, D., Ropke, S., 2018. Large neighborhood search, in: Handbook of metaheuristics. Springer, pp. 99–127.

Pluta, A., Medjroubi, W., Diettrich, J.C., Dasenbrock, J., Tetens, H.P., Sandoval, J.E., Lunsdorf, O., 2022. Data Model of the European Gas Transport Network, in: 2022 Open Source Modelling and Simulation of Energy Systems (OSMSES), Institute of Electrical and Electronics Engineers Inc.. pp. 1–7. doi:10.1109/OSMSES54027.2022.9769122.

Pochet, Y., Wolsey, L.A., 2006. Production planning by mixed integer programming. Springer Science & Business Media.

Popper, K., 2005. The logic of scientific discovery. Routledge.

Porteus, E.L., 1971. On the optimality of generalized (s, S) policies. Management Science 17, 411–426.

Porteus, E.L., 2002. Foundations of Stochastic Inventory Theory. Stanford University Press, CA.

Reuters, 2022. Finland leases floating lng terminal for 10 years to break free from russian gas. URL: https://www.reuters.com/business/finland-agrees-10-year-charter-fsru-us-firm-excelerate-2022-05-20/.

Riepin, I., Schmidt, M., Baringo, L., Müsgens, F., 2022. Adaptive robust optimization for European strategic gas infrastructure planning. Applied Energy 324, 119686. doi:10.1016/j.apenergy.2022.119686.

Rossi, R., Kilic, O.A., Tarim, S.A., 2015. Piecewise linear approximations for the static–dynamic uncertainty strategy in stochastic lot-sizing. Omega 50, 126–140.

Rzhetsky, A., Nei, M., 1993a. Theoretical foundation of the minimum-evolution method of phylogenetic inference. Molecular biology and evolution 10, 1073–1095.

Rzhetsky, A., Nei, M., 1993b. Theoretical foundations of the minimum evolution method of phylogenetic inference. Molecular Biology and Evolution 10, 1073–1095.

Ríos-Mercado, R.Z., Borraz-Sánchez, C., 2015. Optimization problems in natural gas transportation systems: A state-of-the-art review. Applied Energy 147, 536–555. doi:10.1016/J.APENERGY.2015.03.017.

Saitou, N., Nei, M., 1987. The neighbor-joining method: A new method for reconstructing phylogenetic trees. Molecular Biology and Evolution doi:10.1093/oxfordjournals.molbev.a040454.

Sarker, B.R., Wu, B., Paudel, K.P., 2022. Optimal Location for Renewable Gas Production and Distribution Facilities: Resource Planning and Management. Bioenergy Research 15, 650–666. doi:10.1007/s12155-021-10278-0.

Saxe, A.M., McClelland, J.L., Ganguli, S., 2014. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks, in: International Conference on Learning Representations.

Scarf, H.E., 1960. The optimality of (S, s) policies in the dynamic inventory problem .

Schäl, M., 1976. On the optimality of (s, S)-policies in dynamic inventory models with finite horizon. SIAM Journal on Applied Mathematics 30, 528–537.

Schulman, J., Levine, S., Abbeel, P., Jordan, M., Moritz, P., 2015. Trust region policy optimization, in: Bach, F., Blei, D. (Eds.), Proceedings of the 32nd International Conference on Machine Learning, PMLR, Lille, France. pp. 1889–1897. URL: https://proceedings.mlr.press/v37/schulman15.html.

Schulman, J., Moritz, P., Levine, S., Jordan, M., Abbeel, P., 2016. High-dimensional continuous control using generalized advantage estimation, in: Proceedings of the 4th International Conference on Learning Representations.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O., 2017. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 .

Schwartz, R., 2019. Computational models for cancer phylogenetics, in: Warnow, T. (Ed.), Bioinformatics and Phylogenetics. Computational Biology. Springer, Cham. volume 29, pp. 243–275.

Semple, C., Steel, M., 2004. Cyclic permutations and evolutionary trees. Advances in Applied Mathematics 32, 669–680.

Sesini, M., Giarola, S., Hawkes, A.D., 2020. The impact of liquefied natural gas and storage on the EU natural gas infrastructure resilience. Energy 209. URL: https://doi.org/10.1016/j.energy.2020.118367, doi:10.1016/j.energy.2020.118367.

Silver, D., Huang, A., Maddison, C.J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al., 2016. Mastering the game of go with deep neural networks and tree search. Nature 529, 484.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al., 2017. Mastering the game of go without human knowledge. Nature 550, 354.

Sinkhorn, R., Knopp, P., 1967. Concerning nonnegative matrices and doubly stochastic matrices. Pacific Journal of Mathematics 21, 343–348.

Songhurst, B., 2017. The outlook for floating storage and regasification units (FSRUs). Oxford Institute for Energy Studies 12, 1–54. URL: https://www.oxfordenergy.org/publications/outlook-floating-storage-regasification-units-fsrus/.

Sörensen, K., 2015. Metaheuristics—the metaphor exposed. International Transactions in Operational Research 22, 3–18. URL: https://doi.org/10.1111/itor.12001, doi:10.1111/itor.12001.

Stamatakis, A., 2005. An Efficient Program for Phylogenetic Inference Using Simulated Annealing, in: 19th IEEE International Parallel and Distributed Processing Symposium, IEEE, Denver, CO, USA. pp. 198b–198b. doi:10.1109/IPDPS.2005.90.

Sumicsid, 2019. Norm Grid Development, revision V1.3, CEER Project TCB18. Technical Report. Sumicsid. doi:10.13140/RG.2.2.19134.09289.

Sutton, R.S., Barto, A.G., 2018. Reinforcement learning: An introduction. MIT press.

Vanvuchelen, N., Gijsbrechts, J., Boute, R., 2020. Use of proximal policy optimization for the joint replenishment problem. Computers in Industry 119, 103239.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. Advances in neural information processing systems 30.

Veinott, Jr, A.F., 1966. On the opimality of (s, S) inventory policies: New conditions and a new proof. SIAM Journal on Applied Mathematics 14, 1067–1083.

Waterman, M.S., Smith, T.F., Singh, M., Beyer, W.A., 1977. Additive evolutionary trees. Journal of Theoretical Biology 64, 199–213.

Xiang, M., Rossi, R., Martin-Barragan, B., Tarim, S.A., 2018. Computing non-stationary (s, s) policies using mixed integer linear programming. European Journal of Operational Research 271, 490–500.

Zwickl-Bernhard, S., Rodgarkia-Dara, A., Gatzen, C., Sonnen, L., Lane, A., Otti, M., Golab, A., Auer, H., 2024. Modeling insights from the Austrian national gas grid under declining natural gas demand and increasing domestic renewable gas generation by 2040. Energy Reports 11, 1302–1317. doi:10.1016/J.EGYR.2023.12.064.