Multi-period adaptive fleet planning problem

with Approximate Dynamic Programming

Laura Requeno García
Supervisor: Dr. ir. B.F. Santos

MSc THESIS

TU Delft Delft University of Technology

Kenya Airways The Pride of Africa

Multi-period adaptive fleet planning problem

with Approximate Dynamic Programming

by

L. Requeno García

Delft University of Technology

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on July 17, 2017 at 10:00 AM.

Student number:	4508831	
Project duration:	December 10, 2016 – July 17, 2017	
Thesis committee:	Prof. dr. ir. R. Curran,	TU Delft, ATO Chair
	Prof. dr. ir. B.F. Santos,	TU Delft, ATO Supervisor
	Prof. dr. ir. E. van Kampen,	TU Delft, C&S External member

This thesis is confidential and cannot be made public until July 17, 2017.

Cover frontpage: URL "https://travel.jumia.com/blog/wp-content/uploads/2016/04/", [cited June 21 2017]

TU Delft

Acknowledgments

The present MSc thesis has been carried out at the chair of Air Transport & Operations of the Aerospace Engineering faculty of Delft University of Technology. This work is the result of nine intense months where hard work, perseverance and passion have blended together to completely shape my skills as a MSc aerospace engineer.

Before starting this journey, it was clear to me that my personal and professional satisfaction depended on discovering a passionating field for research, being mentored by an inspiring supervisor and above all, surrounding myself of the best people.When it comes to motivation, not only is the research topic important, but also with whom you work.

My first thanks go to my supervisor, Dr.ir. Bruno F. Santos for introducing me to the amazing field of *the multi-period adaptive fleet planning problem with approximate dynamic programming*. His critical thinking and constant trust have always encouraged me to leave my comfort zone while pushing my limits. I will definitely miss all our update meetings which eventually led to debates and fun conversations. I would also like to express my gratitude to all professors from Air Transport & Operations for providing me with a great academic curriculum. A special thanks goes to Dr.ir. Dries Visser for offering me his valuable insights whenever I needed them. Despite not knowing him, I strongly believe that I must acknowledge the extraordinary work of Dr.ir. Warren B. Powell, who has become the major guiding reference of this MSc thesis.

Allow me to also thank Kenya Airways for giving me the opportunity to test my model in a real experimental setting. I feel very much lucky to having exchanged ideas with Thomas Omondi, Stephen Ngamau, Richard Mwikamba and all the colleagues of the network planning department. Likewise, I appreciate the time and interest shown by Rob Hogema, Jeroen Erdman and Wouter Stikvoort from Transavia.

When looking back at these two years in Delft, I honestly feel that I have earned so much more than a MSc diploma. I consider myself very lucky to have been surrounded by outstanding people from whom I have learnt and have been inspired every day. I would especially like to thank Sofia and Lucrezia for being the best flatmates I could have had. To Alejandra, Ilaria, Lotfi, Fede, Isma, Ari, Wouter, Timo and Anne-Nynke, who have provided me with an enriching and unforgettable international experience. I would like to especially mention Marta, who has become my second half and my best friend throughout this year. Without her daily energy, it would have been very difficult for me to come this far.

In expressing my acknowledgement, I cannot overlook the affection that many people have made me feel in spite of the distance. Thanks to Omar, Johan, Fer, Paula and Álvaro for truly caring about our friendship. Thanks to my lifelong friends Carla and Glòria for their lasting unconditional support.

Finally, my utmost sincere gratitude goes to the people who have always been by my side behind the scenes. I would especially like to thank Francesco for becoming the greatest companion in the best and most challenging moments. His joy, affection, patience and critical thinking have been fundamental in this journey. My warmest words are addressed to my sister and my parents, for always supporting me with their endless love. Thanks for giving me all the opportunities you could, and for trusting unconditionally all the decisions I took during the hardest and most uncertain moments of my life. Everything I have become and achieved is thanks to you.

Laura Requeno García
July 2, 2017
Delft, The Netherlands

Abstract

The inherent uncertainty of the fleet planning problem has hindered the emergence of sophisticated models to support airlines with strategic decision-making. For many years, airlines have been applying similar top-down deterministic approaches when planning their fleet development. In fact, spreadsheets are still commonly used tools by many airlines. This lack of sophistication is somewhat striking considering the important contributions that operations research has made to other domains in the airline planning process. In particular, the operations research community has shown a clear preference for multistage stochastic models when it comes to drawing adaptive fleet policies under demand uncertainty. Because of their recursive structure, one of the most popular methods to solve multistage stochastic models has been dynamic programming, based on the backward induction of their equivalent scenario trees. Nevertheless, this has often led to complex models whose computational and memory requirements go beyond the capabilities of current commercial optimisers, thereby hampering the development of fleet planning models. In this context, approximate dynamic programming (ADP) has emerged as an optimising simulator since it proposes a solving strategy based on the flexibility of Monte Carlo simulations and the power of operations research, both combined with machine learning. This blend of disciplines makes ADP a novel and promising method to solve multistage stochastic problems, whose computation have appeared to be so far intractable with backwards exact methods. Indeed, ADP allows to decompose large-scale problems by approximating unknown value functions, thus reducing the computational times required by dynamic programming.

Therefore, the objective of this MSc thesis is to contribute to the development of adaptive policies in the context of airline fleet planning under demand uncertainty by (a) modelling and solving with Approximate Dynamic Programming a multi-period adaptive fleet planning problem that integrates stochastic demand and by (b) detecting useful signposts for fleet planners.

With the aim of meeting this research objective, the developed methodology is split into two parts. Firstly, the problem is modelled as a dynamic program thanks to the suitable application of a state-space modelling framework. Next, an ADP algorithm based on value function iterations is implemented. The proposed ADP algorithm applies local value function approximations resulting from Gaussian kernel regressions to estimate future airline operating profits. Likewise, kernel regressions need to be trained with an initial set of structured observations, which provide meaningful information of the problem. The full multi-period adaptive fleet planning problem is then optimised by decomposing it into subproblems. To take full advantage of the computational power of commercial optimisers as well as the two-stage overlapping structure existing between subsequent subproblems, each maximisation subproblem is simultaneously decomposed into two independent parts A and B. Part A is solved directly with Gurobi and refers to the calculation of operations-related actions once previous fleet decisions have been made. Likewise, part B corresponds to the problem of selecting the fleet-related decisions that will impact future operations. For this part, an epsilon-greedy subroutine based on simulated annealing is developed to let the algorithm explore the state space and discover other states with high potential.

By applying this methodology, a 4-period fleet planning problem of 20 routes and 3 aircraft types is solved in both its deterministic and stochastic versions. Optimal and near-optimal solutions are achieved for the deterministic case with an average optimality gap of 0.2%: a magnitude which is highly comparable to the optimality gaps currently seen in ADP literature. Nevertheless, the ADP algorithm does not offset the computational performance of Gurobi, which can reach the same optimality gaps in 35s compared to the 120s of the ADP algorithm. Despite this, the potential benefits of ADP are expected to be more visible for the stochastic version of the problem. Indeed, it is in this context that the ADP adaptive policy proves to be the most robust fleet planning method: across the majority of scenarios it clearly excels the profits resulting from the optimal fleet plan solved for the most-likely scenario. Evidently, the most-likely solution performs better in those scenarios similar to the most-likely scenario. However, its performance starts to stall towards differing scenarios. Indeed, its worst scores are found in the neighborhood of the extreme scenarios: for the most pessimistic scenarios the most-likely solution could imply weekly operating losses ranging between 3% and

6%, whereas in the most optimistic scenarios losses remained between 2% and 1%. In contrast, the adaptive fleet policy mitigates losses in all extreme scenarios, which never surpass the 1.1%. By analyzing the general expected performance, it is concluded that the adaptive policy could reduce by 50 % the losses entailed by the most-likely solution.

Apart from this, a verification analysis has proven the stable behaviour of the ADP algorithm. By modifying the parameters of the baseline case within different intervals, it is confirmed that the kernel approximation strategy can still be trained successfully using the same dataset of initial observations. Furthermore, a sensitivity analysis is carried out to understand better the influence of the most relevant parameters within the ADP algorithm. On the one hand, the learning rules with the fastest value function updates provided the best results for the value iteration algorithm. On the other hand, the correct calibration of the control parameter regulating the impact of the initial value function approximation becomes essential to guarantee the convergence of the ADP algorithm as well as the coherence of the adaptive policy.

The conclusions obtained with the proof of concept unveil the advantage of applying adaptive fleet policies to hedge against demand uncertainty, apart from their capability to successfully reproduce the fleet planning decision-making process. Indeed, the obtained adaptive policies can capture the existent path dependency between the recommended fleet plans and the historical demand evolution.

Finally, an extended case study and expert survey carried out at Kenya Airways (KQ) have provided valuable insights to validate the performance of the developed ADP support tool in the air transport industry. The ADP tool has been capable of tackling successfully a 5-period fleet planning problem with 5 aircraft types and 64 routes aggregated per market growth regions. The proposed case study tackles 64 routes and 5 aircraft types, and consists in moving backwards in time to year 2015 with the objective of reproducing a 5-year expansion plan formerly defined by Kenya Airways. In this way, ADP recommendations are compared to the past reality. The results obtained from this validation analysis highlight the capability of the ADP algorithm to capture back in 2015 realistic and meaningful trends applicable to the present and upcoming years. Despite not considering the interactions between captured demand and frequency nor fuel or competition uncertainty, the ADP support tool is able to provide a future fleet policy, whose profitability performance proves to be notably better than the actual fleet plan followed by Kenya Airways from 2015 onwards. Even better, the ADP tool forecasts successfully the operating profits that KQ actually earned in 2016. Likewise, a network-frequency analysis presents a satisfactory correlation between the ADP recommended frequencies and the real frequency levels currently operated by KQ. Apart from this, the introduction of fleet maps and operational maps has allowed KQ fleet planners to quickly identify valuable signposts and general trends. Lastly, it is concluded that many of the recommendations provided by the ADP support tool back in 2015 strongly agree with the current strategy of Kenya Airways. In other words, this case study proves that the ADP support tool can adjust well to reality while providing meaningful adaptive policies to hedge against uncertainty, which indeed could make a difference in airline fleet planning.

Contents

List of Figures

List of Tables

1

An introduction to airline fleet planning

This chapter aims at introducing the reader to the subject of this Master Thesis: airline fleet planning. In Section 1.1, the fleet planning problem and its implications will be generally defined within the framework of transportation industry. Then, Section 1.2 will go through the characteristics of fleet planning in the context of the airline industry and its current modeling techniques.

1.1. The fleet planning problem

Fleet planning is a long-term strategic problem faced by all transportation companies, which mainly consists in deciding the quantity and composition of vehicles to be acquired along the future years in order to meet market demand and maximise profits.

Fleet planning is of paramount importance to transportation companies since it is the stage at which high capital investment decisions are made in order to materialize a previously defined strategy into future operations. Indeed, fleet composition and size can determine operations performance and thus, the accomplishment or failure of a strategy. For a long time, this issue has raised the interest of many researchers and companies, who have developed several different techniques to tackle this problem. While operations-related problems have undergone a dramatic improvement owing to progressively sophisticated computational techniques, the fleet planning problem has not benefited in the same way given its inherent uncertainty (Barnhart et al.; 2003). In fact, uncertainty might be considered in different shapes such as demand, operating costs, market competition and disruptions, as well as in different intensity levels, from clear-enough futures to true ambiguity passing through alternate scenarios (Courtney et al., 1997). In light of this, the way of embracing uncertainty into fleet planning decision-making models has been for many years a primary focal point.

By definition, fleet planning can be considered a supply-demand matching problem and thus, it shares many analogies with other general strategic problems faced by manufacturing and energy companies such as technology investments and capacity expansion. In terms of carrying out research, this fact is very relevant to consider since it may broaden the scope of study and obtain richer literature.

1.2. Fleet planning within the airline industry

1.2.1. Context

It goes without saying that the deregulation of the airline industry meant a turning point for airline management. The introduction of competition into market led to dramatic pricing reductions, thereby forcing airlines to reduce costs by focusing on operations optimisation. While in a regulated environment, profitability, technology advancements and operations efficiency were not a matter of concern, deregulation shaped a new landscape in which the airline planning process had to be redefined.

Nowadays, the core steps of airline planning are fleet planning, schedule planning, revenue management and crew scheduling. Belobaba et al. (2015) explained thoroughly the main characteristics of each of these activities. In his PhD. thesis, Lohatepanont (2002) drew a clear scheme of all the sequential steps integrating the airline planning process as it is performed by the major airlines. This scheme is shown in Figure 1.1, where every stage is positioned according to the type of decision made - strategic to tactic- and the time horizon - from several years to days -.

As in other industries, fleet planning is considered to be a long-term strategic problem that is the starting point of the entire company's planning process. Indeed, the definition of a fleet will affect future route and schedule development and eventually, the efficiency and profitability of its operations. This means that the airline's fleet will also have a long lasting financial impact that will become visible during a 10-year horizon at the very least by means of operating and depreciation costs, long-term debt and corresponding interest expenses (Belobaba et al., 2015).

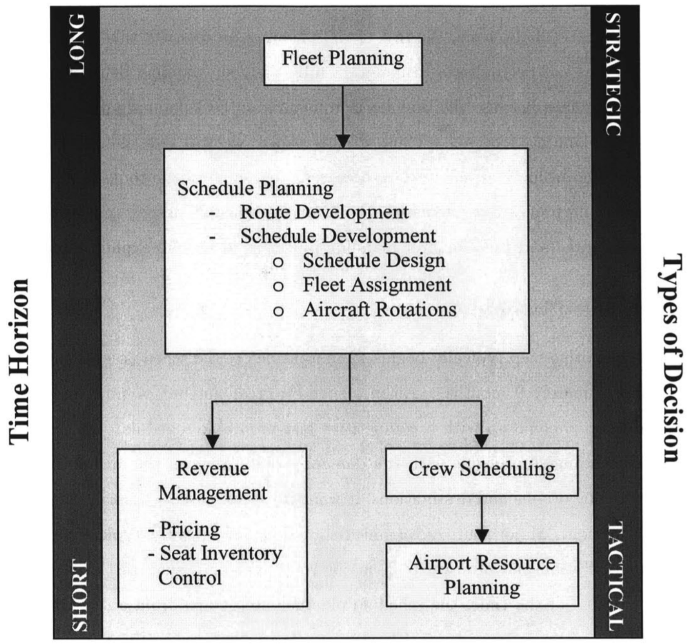


Figure 1.1: The airline fleet planning process (Lohatepanont, 2002)

Apart from that, an airline takes into account many aspects when choosing amongst aircraft types. On the one hand, aircraft performance characteristics play an important role in the deliberation. In fact, the range of an aircraft will constrain the selection of routes in which it is operated, while its capacity will define the number of seats available in each flight. Besides, and not less important, operating costs in terms of fuel, maintenance and airport taxes will also have a great impact on fleet composition and efficiency. On the other hand, an airline may be forced to select an aircraft type due to current constraints set by its previous fleet composition as well as its current maintenance expertise and infrastructure. Nevertheless, there might be other non-technical reasons such as the current financial health of the airline, political issues, international trades or airport restrictions.

1.2.2. Fleet planning models

As previously commented, fleet planning decisions result from a complex trade-off in which many technical and non-technical aspects must be evaluated. Strikingly, airlines do not make use of really sophisticated mathematical models to support their decision-making. In contrast, spreadsheets are still commonly used tools to analyse the impact of a fleet on a system of routes and develop their fleet plans according to it.
Regardless of the analytical detail, Belobaba et al. (2015) summarises the fleet planning evaluation process as depicted by Figure 1.2. From this scheme, it can be inferred that the overall process is carried out iteratively,

following three main steps. Firstly, the model is fed with traffic demand and yield forecasts to estimate load factors and available seats per mile (ASM) required. Secondly, the fleet composition and its size is defined taking into account the available seats per mile required as well as corresponding operating costs. Finally, financial sustainability is assessed once operating profit margins and expense forecasts are performed.

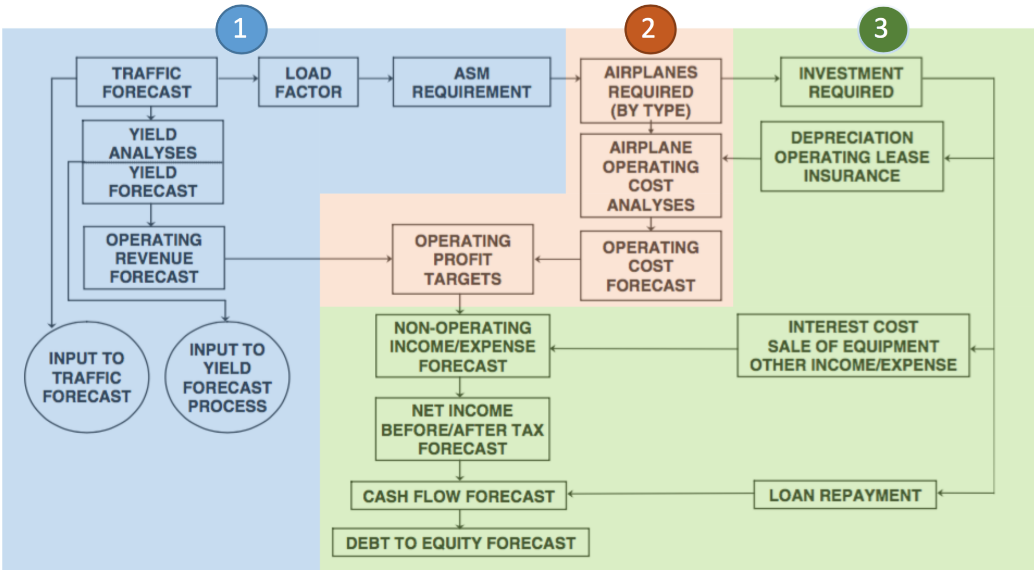


Figure 1.2: Fleet planning economic evaluation process (elaborated from Belobaba, 2009)

Depending on the level of detail applied to the above fleet planning evaluation process, literature classifies fleet planning models into two types (Belobaba et al., 2015):

- **Top-down or macro approach**, in which the fleet planning problem is modelled at a high aggregate level. General forecasts, aircraft data and many assumptions are taken as inputs for this type of analysis. They usually consist in more or less sophisticated spreadsheets that assess the hypothetical financial impact of a certain fleet on a set of routes. It provides rough but quick estimations that tend to be generally accepted by the airline industry due to the high level of uncertainty faced.

- **Bottom-up or micro approach**, in which the fleet planning problem is modelled in detail taking into account many system complexities. Instead of accepting general data, these models aim at being as much realistic as possible by means of taking into account all characteristics of the subsystem studied (e.g. origin-destination market demand forecasts, specific aircraft constraints, demand and competition uncertainty, fuel costs variability and alliance strategies). Nevertheless, increasing the level of detail adds complexity to the model, thereby requiring higher computational times to obtain more detailed results. The key issue is obtaining fair enough detailed inputs, which can be problematic due to uncertainties associated to long-term horizons.

Despite the improvement of computational power, the top-down approach is still in many cases the preferred technique by many airlines. This is due to the fact that it provides quick but sufficient estimations to make strategic decisions under uncertainty. Even though a bottom-up approach might provide more detailed and optimal solutions under a certain scenario, the question is how beneficial this approach might be when a long-term horizon implies inherent uncertainty, thereby holding a wide range of possible scenarios. As it will be reported in Chapter 2, the operations research community has dedicated high effort to move from deterministic to stochastic environments with the objective of obtaining robust enough-detailed solutions under different scenarios. Nevertheless, this has often led to complex models whose computation requirements go beyond the capabilities of current commercial optimisers. In fact, this dilemma is one of the reason why fleet planning models and corresponding solving techniques have not developed at the same pace as other airline planning-related problems. As reported by Barnhart et al. (2003), during the last decades, schedule development, revenue management and crew scheduling problems have undergone a dramatic improvement thanks to several contributions from the operations research discipline. Furthermore, differences in time-horizon from several years to some days in advance have also prevented fleet planning from being integrated into the rest of the airline planning process. Therefore, it can be seen that a common trend in literature is to take an airline's fleet as given and as a fixed input to the rest of the subsequent optimisation problems (Lohatepanont, 2002).

1.3. Summary

The inherent uncertainty of the fleet planning problem has hindered the emergence of sophisticated models to support airlines with strategic decision-making. For many decades, airlines have been applying similar top-down approaches when planning their fleet development. In fact, spreadsheets are still commonly used tools by many airlines. This lack of sophistication is somewhat striking considering the important contributions that operations research has made to other domains in the airline planning process such as schedule planning, revenue management and crew scheduling. In this type of problems, bottom-up models have been able to provide optimal solutions or at least, better than the ones obtained in a high-level aggregate way (Belobaba et al., 2015;Barnhart et al., 2003). In light of this, the question constantly raised in literature is how the fleet planning problem can take full advantage of bottom-up models while allowing for the uncertainty effects of a long-term horizon. As discussed in next chapter, fleet planning models have not experienced a significant evolution, thus remaining very similar along the years.

2
Literature review

For decades, fleet planning has risen the interest of many researchers and companies, who have been coming up with different models and solving techniques to tackle this problem. In its origins, fleet planning was tackled with deterministic models and linear programming. Nevertheless, in industries characterised by high capital investments with long-term impact, fleet planning is a problem in which long-term horizons must be considered so as to be useful and capture reality successfully. By definition, long-term horizons lead to inherent uncertainty, which can be present in different dimensions such as demand, competition, operating costs, disruptive events, etc. This is the particular case of the airline industry, a fast-paced and competitive market where uncertainty plays an essential role impossible to overlook in fleet planning models. In light of this, the fleet planning problem evolved from deterministic to stochastic modelling so as to include uncertainty in the decision-making process.

In this context, the aim of the present chapter is to review the existent literature related to fleet planning and any other analogue problems relevant to it, with a particular interest in the airline industry. Deterministic fleet planning models are discussed in Section 2.1, while Section 2.2 presents the different trends found in stochastic fleet planning models. Next, Section 2.3 goes beyond the performance of a single company and looks at the major contributions made in terms of market interactions modelling in the context of strategic planning. Finally, Section 2.4 will conclude this literature review by identifying the current knowledge gap in fleet planning.

2.1. Deterministic fleet planning models

From 1957 onwards, several researchers devised deterministic models to solve the fleet planning problem applied to the transport industry, mainly airlines and rail companies. Throughout the years, two main approaches were followed to model the fleet planning problem: a strategic perspective and a tactical perspective.

2.1.1. Strategic fleet planning models

On the one hand, there was a research stream (e.g. Shube and Stroup, 1975; Schick and Stroup, 1981; Bazargan and Hartman, 2012) adopting a strategic perspective which aimed at developing multistage models with long-term horizons at the expenses of simplifying the details in fleet management. Despite the existent time lapse between their publications, it can be concluded that multistage fleet planning problems did not experience much of an improvement in modelling when compared to the pioneering model of Shube and Stroup (1975). To the best of the writer's knowledge, Shube and Stroup were the first ones in designing a multistage problem to model fleet states in different time moments. This novel characteristic allowed to account for future demand evolution and aircraft lifespan. In other words, the model of Shube and Stroup became the basis of future contributions and reaffirmed the need to invest in computational techniques such as linear programming. In fact, the developed fleet planning model belonged to the problem category of linear programming (LP) since all its mathematical expressions were linear. Regarding its solving strategy, Shube and Stroup referred to the simplex method as a straight-forward technique to solve small dimensional problems. Nevertheless, the authors also looked ahead to the computational limitations that larger problems would entail.

Machine programming was already envisaged as a necessary investment to tackle fleet planning problems of a realistic size.

2.1.2. Tactical fleet sizing models

On the other hand, there was a more tactical research stream (e.g. Bartlett, 1957; Gertsbach and Gurevich, 1977; Sayarshad and Ghoseiri, 2009) that developed fleet sizing models by means of adopting a short-term perspective with a high-detailed representation of operations. Bartlett (1957) was one of the first authors in addressing the problem of fleet planning by means of designing an algorithm that calculates the minimum number of transport units to maintain a fixed schedule of a small network. Despite his specific focus on rail transportation, Bartlett explicitely adopted a general problem formulation using generic wording such as *transport units* for railcars or *runs* for trips. In this way, other industries could also benefit from the algorithm. In the same line as Bartlett (1957), Gertsbach and Gurevich (1977) designed a more sophisticated technique to come up with an optimal fleet for a transportation schedule. Again, fleet sizing was performed in a short-term basis and came as a result of a predetermined schedule. The proposed method was not applied to any specific industry unlike Bartlett's work. Indeed, the formulation of the problem was so abstract that it could be applied to either a transportation or production schedule. Furthermore, Sayarshad and Ghoseiri (2009) employed a deterministic model to study a multi-periodic rail-car fleet sizing problem and solved it by means of heuristics. Their objective was to integrate two problems in the rail industry that are still often solved independently: fleet sizing and fleet assignment. If extrapolated, their study could be valuable to the airline industry, where problems are also optimised subsequently rather than in an integrated way. Furthermore, it was claimed that one of the advantages of the proposed model was the optimal use of rail-cars throughout the different time stages of demand. This characteristic breaks with the past tendency of modelling static fleet sizing problems (see for example Bartlett, 1957; Gertsbach and Gurevich, 1977). The reader should take into account that the time scale considered by Sayarshad and Ghoseiri are operating days. Therefore, this model adopts a tactical perspective rather than a strategic perspective (operating years) seen in the models from Schick and Stroup (1981) and Shube and Stroup (1975).

2.1.3. Follow-up research

In conclusion, both research streams contributed in different ways to the fleet planning problem. While a multistage approach was more useful to support fleet planning decisions along the years, the adoption of a short-term perspective contributed more to the integration of fleet management and planning problems. In any case, both approaches led eventually to large-scale integer programs that required a high computational effort to be solved. It is for this reason that a great academic effort (e.g. Nemhauser and Wolsey, 1988; Barnhart et al.,1998, 2000, 2004) was concentrated in developing optimisation methods to solve huge integer programming problems significantly connected to fleet planning (e.g. fleet assignment and scheduling problems). Many other problems such as revenue management or crew scheduling greatly benefited from the achievements made in the field of optimisation methods (e.g. Branch & Bound , Branch & Cut). Nevertheless, the fleet planning problem did not experience the same improvement due to its inherent uncertainty. Nowadays, it is not longer common to find recent literature dedicated to deterministic planning models since several researchers started to question the usefulness of deterministic planning when high levels of uncertainty prevail within a long-term horizon. In other words, finding an exact solution to a specific future situation was not longer a research priority given that this situation is uncertain to happen eventually. Therefore, the current development of deterministic models revolves around reasons other than offering an exact solution. In many cases, modern deterministic models are used to simplify certain parts of a problem so as to allow a neater study of a specific characteristic such as the performance of an algorithm or the integration of some modelling aspects (Sayarshad and Ghoseiri, 2009; Bazargan and Hartman, 2012).

2.2. Stochastic fleet planning models

Until now, the reviewed fleet planning models have considered deterministic systems, where all future input values and parameters were assumed to be known. Nevertheless, the reality is that all industries face some kind of uncertainty and risk, which implies the existence of random processes affecting future states. This is the special case for transport industry, where uncertainty and risk can have a dangerous impact on operating profits if companies do not account for them appropriately in their strategic plans. Furthermore, uncertainty and risks can take several forms, affecting a specific or several parts of the profits structure of a company. Indeed, airlines are constantly dealing with uncertainty and risk elements: Figure 2.1 shows an overview of the different uncertainty dimensions that may affect either airline revenues, operating costs or both. In such a competitive and demanding market, the importance of embracing uncertainty is more alive than ever and it could definitely determine the success or bankruptcy of an airline. Therefore, it can be inferred that airlines operate in a system which differs greatly from a deterministic model. In this context, many researchers in the transport field (e.g. List et al., 2003; Listes and Dekker, 2005; Hsu et al., 2011) proved that deterministic models were not well suited for providing optimal fleet plans since their optimality was only valid for a certain scenario. This idea is what drove many researchers to eventually delve into stochastic systems.

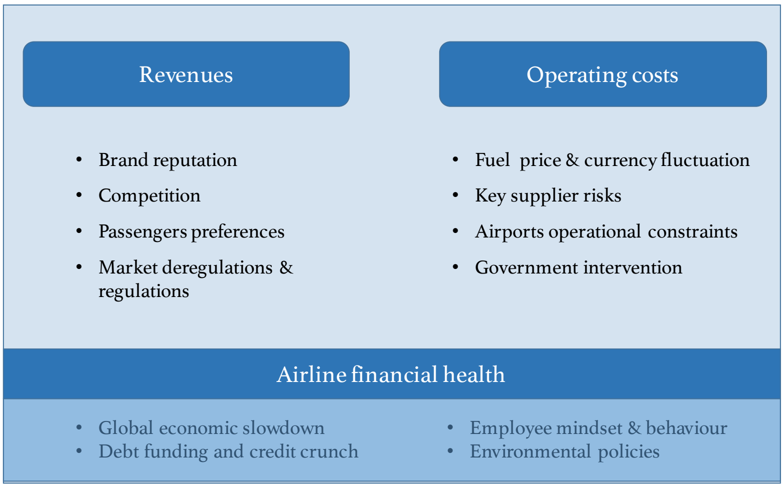


Figure 2.1: Potential risks and uncertainties in airline's operations (elaborated from British Airways, 2008-09)

For many years, there was an important research trend towards the development of sophisticated forecasts by means of logistic regressions (also known as logit models) and variations. Companies and research academies used to employ these forecasting methods in order to mitigate the faults of deterministic planning models (e.g. Brooke et al., 1994; Hansen and Weidner, 1995; Hess et al., 2013). In doing so, the available planning models were providing deterministic solutions for forecasted scenarios. In fact, main strategies consisted in trying to predict the future as a way to deal with uncertainty. Consequently, strategic plans were more prone to fail if future did not turn out as expected.

Considering that a wide range of scenarios could occur within a long-term horizon, by 1990's a parallel research stream started to gain momentum basing its foundations on stochastic programming. Stochastic programming (SP) became a mathematical optimisation discipline that allows to model and solve problems with uncertainty by means of introducing random parameters. Using the notation of Kali and Wallace (1994), the general formulation of a stochastic problem is:

$$\text{min (or max)} \quad f(x,\xi) \tag{2.1}$$

subject to the constraints:

$$g_i(x,\xi) \leq 0, \quad i = 1, \dots, m \tag{2.2}$$

$$x \in X \subset \mathbb{R}^n \tag{2.3}$$

where x is the vector of decision variables and ξ represents a random vector taking values from a set $\Xi \subset \mathbb{R}^k$ with a known or unknown probability distribution $\mathbb{P}$.

As synthesized by Shapiro et al. (2014), SP models were drawn from a combination of optimisation, probability and statistics as well as functional theories. Dantzig (1955) was one of the pioneers in laying down the mathematical principles which would define stochastic programming: recourse modelling, two-stage and multistage programming models. From that moment on, several articles proliferated during decades, driving the need of unifying the available literature in textbooks. Kali and Wallace (1994), Higle (2005), Shapiro and Philpott (2007) and Shapiro et al. (2014) contributed to this objective by strengthening the basis of stochastic programming for further studies. Indeed, stochastic programming has definitely been established as the primary tool for researchers when it comes to model uncertainty.

From literature, it can be inferred that stochastic fleet planning models and relevant problems have been expressed either in a two-stage or multistage structure. While two-stage structures have commonly been used to model problems with a short-term or tactical perspective (e.g. Oum et al., 2000; List et al., 2003; Listes and Dekker, 2005; Naumann and Suhl, 2013), multistage structures have finally been considered the best technique to draw optimal planning policies in the long-term strategic perspective (e.g. Hsu et al., 2011; Khoo and Teoh, 2014).

2.2.1. Major contributions based on two-stage stochastic structures

By means of modelling a two-stage stochastic problem, Oum et al. (2000) aimed at providing an optimal mix of leased and owned capacity. They focused their study on the operational effects of aircraft leasing in a context where demand is uncertain and cyclical. Backed up by many financial studies, Oum et al. stated that aircraft leasing offered airlines the possibility of sharing risks, lowering their debt and increasing their operational flexibility in exchange of assuming higher operating costs. Consequently, the authors highlighted the importance of carrying out a trade-off optimisation between capacity flexibility and higher costs. The problem was formulated in a very analytical way without considering any sort of discretization for programming. Instead, an empirical examination was performed based on 23 major airlines to determine their optimal demand for leased capacity. In this way, the different components of the airline profit function (e.g. variable costs, revenues, etc.) were estimated based on time-series data observed during the period 1986-1993. Furthermore, the probability distribution of demand for each carrier was considered to follow the shape of a normal distribution, whose mean and standard deviation were determined from available statistical data. Finally, Oum et al. concluded that the optimal proportion of leased aircraft would be between 40 to 60%. Concerning these results, one could note that the study of Oum et al. (2000) adopts an aggregate perspective. For instance, it just gives information about the general rate between leased-owned aircraft but it does not specify any fleet composition. On the other hand, the study discriminates regions according to their demand forecasts and the cost premium of lease.

In the context of robust optimisation, List et al. (2003) presented a fleet planning study under uncertainty with a particular focus on freight industry. In their paper, a formulation and a solution technique were proposed to tackle the problem of fleet sizing under uncertain demand and uncertain operating conditions. To the best of the writer's knowledge, List et al. were the first authors in incorporating two simultaneous sources of uncertainty in a transportation planning problem. In order to justify their contribution, List et al. criticised the fact that previous robust optimisation models applied a mean-variance trade off, giving equal weight to all deviations with respect to the mean. According to List et al., that solution robustness was not applicable to specific cases such as fleet sizing in the transportation industry, since that method could lead to inefficient results in real life. Therefore, List et al. aimed at strengthening that weakness by means of defining a new measure of robustness, which was more sensible to the impact of extreme scenarios.and could lead to reasonable results for the transportation industry.

Likewise, Listes and Dekker (2005) were pioneers in introducing stochastic demand to the airline fleet sizing problem. However, they took a fleet management perspective to come up with a strategic solution. That is to say, they used a fixed schedule as input to determine the optimal fleet size and composition with the highest level of flexibility to confront demand uncertainty. On the basis of their problem definition, it can be drawn that this model would be suitable for mid-term decisions, but not for designing a complete strategic fleet plan since different time periods are not considered.

The formulation built on a deterministic multicommodity flow problem applied to a time-space network, commonly used for fleet assignment problems. Then, Listes and Dekker (2005) included demand uncertainty by considering several demand scenarios, which resulted from sampling values of a normal probability distribution $F(\mu,\sigma)$ at equally spaced quantiles, as shown in Figure 2.2. Therefore, the demand uncertainty is defined by a set of scenarios $\{d_1,...,d_S\}$.

The main objective of Listes and Dekker was to find an optimal fleet composition that performed satisfactorily under different scenarios, thereby providing flexibility against uncertain demand. Consequently,

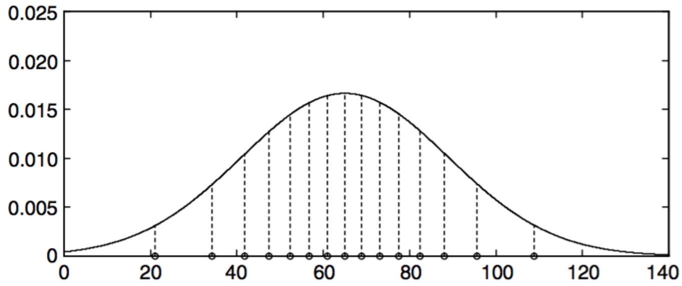


Figure 2.2: Technique for gathering sampling values (Listes and Dekker, 2005)

Listes and Dekker aimed at maximising the expected objective value across all sampled scenarios. Their corresponding stochastic problem was as shown below:

$$\text{SP} \quad \text{Max} \quad \sum_{s=1}^{S} p_s f(z, y_s, x_s, s) \tag{2.4}$$

$$\text{s.t.} \quad (z, y_s, x_s) \in C \quad \forall s = 1, \ldots, S \tag{2.5}$$

where Listes and Dekker saw a two-stage stochastic problem. The fleet composition vector z consisted of first-stage decision variables independent from any scenario parameter s with probability p_s, while y_s and x_s were second-stage decision variables whose value depended on each scenario.

Finally, Naumann and Suhl (2013) gave another perspective to strategic airline planning by assessing the impact of demand and jet fuel price uncertainty on frequency planning. Even though their objective was not directly related to fleet planning, their contribution gives interesting insights about how to approach fuel price uncertainty and develop a fleet planning model based on that. The objective of Naumann and Suhl was to come up with an optimal route frequency plan together with a fuel acquisition strategy to hedge against uncertainty, while maximising the expected airline profits. To achieve that, Naumann and Suhl introduced a two-stage stochastic linear model where jet fuel prices and demand were uncertain parameters. In order to include uncertainty into the model, a set of scenarios FS and DS were generated for fuel price and demand respectively.

2.2.2. Major contributions based on multistage stochastic structures

Multistage structures have proven to be well suited to adopt a strategic perspective and draw optimal fleet planning policies to match demand throughout the years. In the current decade, several researchers have developed multistage planning models to come up with optimal fleet renewal and replacement policies for different sectors, mainly the airline industry (Hsu et al., 2011; Khoo and Teoh, 2014) and the maritime industry (Pantuso et al., 2014, 2015).

Since recursion can be found in multistage stochastic models, dynamic formulations are generally preferred when modelling subsequent time stages and scenario trees. Indeed, it can be stated that there is a clear research motivation behind dynamic planning models: the combination of dynamic programming and scenario trees offers a very intuitive way to model multistage fleet planning problems under uncertainty. In fact, the recursive property of dynamic equations and the discretization of a stochastic process by means of a scenario tree have proven to capture effectively the main issues related to fleet planning problems (Hsu et al., 2011; Khoo and Teoh, 2014).

More particularly, Hsu et al. (2011) used dynamic programming to model a multistage stochastic fleet planning problem within the airline industry. As Oum et al. (2000) previously did with a two-stage stochastic model, Hsu et al. (2011) aimed at optimising the proportion between purchased and leased aircraft in order to mitigate the effects of demand uncertainty. Both models built on a given schedule of route flight frequencies. Nevertheless, the work of Hsu et al. went one step further by capturing better the reality:

- Their model was able to choose from several aircraft types throughout different time periods, whereas Oum et al. (2000) considered a homogeneous fleet along two time periods. From an airline's perspective, the model of Hsu et al. is more realistic and practical since it also assumed a given fleet as initial condition. In contrast, Oum et al. sized the fleet from the very starting point without considering the current state of the airline.

- As far as the modelling of demand uncertainty is concerned, Hsu et al. employed a Grey forecasting model to develop a Markov-chain process, which was represented by the scenario tree depicted in Figure 2.3. In fact, Hsu et al. acknowledged the Grey system theory (Julong, 1989) as a useful predictive method to draw a demand forecast from very poor information. As for the model of Oum et al., it used statistical data from major airlines and assumes a normal distribution of demand. Moreover, it is interesting to point out that Hsu et al. included the concept of market share for each route. That is to say, the airline forecast of captured demand for each route resulted from multiplying its market share to the overall forecasted route demand. Nevertheless, the model did not consider any interactions with other market players.

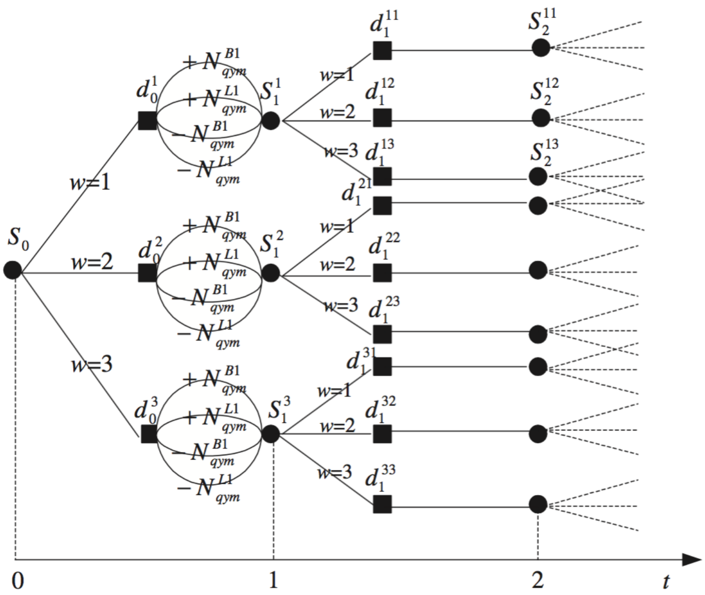


Figure 2.3: Proposed scenario tree for fleet planning under demand uncertainty by Hsu et al. (2011)

- Given the multistage stochastic structure of the problem, Hsu et al. leaned on dynamic programming equations to minimise the expected sum of costs from period t forward. Apart from considering operating and replacement costs, Hsu et al. included an innovative term into the cost function: a cost penalty whose weight depended on the inaccuracy level of the demand forecast.

- The vector d^t of replacement decisions had as components the number of aircraft purchased N^{Bt}_{qym} and the number of aircraft leased N^{Lt}_{qym} at time period t. As observed, aircraft were also associated with a detailed status (q, y, m) which tracked its type q, remaining available years y and mileage travelled m. By letting these variables take negative values, the model captured the selling and termination of leasing contracts. In this way, the model tried to reduce the number of decision variables. Yet, it can be inferred from looking at the detailed definition of N^{Bt}_{qym} and N^{Lt}_{qym}, that the dynamic program had to deal with a great number of decision variables.

Regarding the solving technique, Hsu et al. (2011) relied on backwards induction to determine the optimal replacement strategy for each period. Eventually, the optimal solution π would correspond to the set of replacement decisions for each period: $\pi = \{d^1, d^2, ..., d^n\}$. The obtained conclusion were consistent with the previous results of Oum et al. (2000): leasing is preferred for operating market with high fluctuations in demand. Furthermore, it was concluded that airlines should reduce fleet heterogeneity in order to achieve economies of scale in maintenance and operating costs. Another important contribution of Hsu et al. (2011) was the determination of threshold values which could be used as singposts in adaptive policies to make decisions. Figures 2.4 and 2.5 are examples of it.

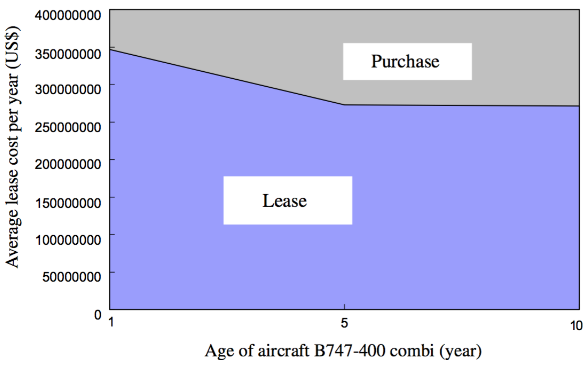


Figure 2.4: Purchase/lease decision threshold in function of lease cost and aircraft age (Hsu et al., 2011)

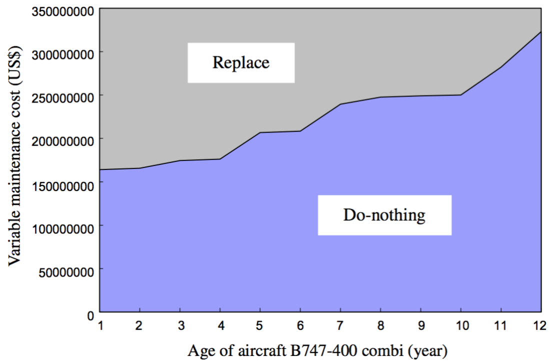


Figure 2.5: Replacement threshold in function of variable maintenace cost and aircraft age (Hsu et al., 2011)

Khoo and Teoh (2014) contributed to the research stream of Hsu et al. (2011) by presenting a new method for modelling demand uncertainty. In spite of just capturing stochastic demand trends, Khoo and Teoh put forward a Stochastic Demand Index (SDI) with the intention of capturing the potential occurrence of disruptive event with positive or negative effects. SDI was calculated by means of Monte Carlo simulations. Apart from that, Khoo and Teoh (2014) stated the need of integrating service frequency planning into fleet planning.

In two subsequent studies, Pantuso et al., (2014, 2015) explored the fleet renewal problem in the context of the maritime industry. Even though they used the notation of Hierarchical Stochastic Programs (HSP), the model that they formulated was still a multistage stochastic program whose objective consisted in minimising the expected total cost of operating and owning ships within a planning horizon. Nevertheless, they put higher emphasis on devising a solution technique which exploited the scenario tree structure in an efficient way. To that end, they decomposed the problem into a master problem and several independent linear programming subproblems. In this way, LP subproblems were solved to optimality, while a heuristic method was implemented for solving the master problem.

2.2.3. The curses of dimensionality and approximate dynamic programming

From the literature discussed, it can be inferred that dynamic programs suffer from great dimensionality problems due to its recursion nature. Powell (2007) referred to these dimensionality problems as *the curses of dimensionality*, which classified into three types: state-space, outcome space and action space. The idea behind this concept is that when the problem features higher dimensions, the volume of potential states, decisions and random processes increases at exponential rates. This fact entails the need for great computational efforts and storing memory to the extent that exceeds current computer capabilities. Due to this fact, many stochastic problems formulated as dynamic programs have been constrained in terms of dimensions (Hsu et al., 2011; Khoo and Teoh, 2014;Pantuso et al., 2014, 2015).

When the number of scenarios is very large or even infinite, a common method to reduce the set of scenarios to a reasonable size are Monte Carlo simulations. Relevant literature (Kleywegt et al., 2002; Schütz et al., 2009; Shapiro et al., 2014) referred to this method as sample average approximation (SAA). The idea consists in sampling the probability distribution of ξ_t so as to obtain N different values $\{\xi_t^1,...,\xi_t^N\}$. Then, if the values are distributed independently from each other, the probability of each sample p^j becomes $1/N$. In this way, the problem is approximated as:

$$\underset{x_t}{\text{Min (or Max)}} \quad f_t(x_t,\xi_t) + \frac{1}{N}\sum_{j=1}^{N} Q_{t+1}(x_t,\xi^j_{[t+1]}) \tag{2.6}$$

Moreover, Heitsch and Römisch (2009) discussed the issues of modelling a multivariate stochastic input process in the form of a scenario tree. They proposed a heuristic technique with backward and forward construction algorithms to create scenario trees out of a series of predefined scenarios. In this way, the obtained scenario tree could be an approximated representation of the stochastic process. To illustrate their idea, Heitsch and Römisch(2009) provided a case study in the context of electricity portfolio management.

In any case, the challenging dimensions associated to high dimensional dynamic systems have led to the stagnation of realistic fleet planning models. Many researchers have concluded that backwards induction is not convenient nor efficient for solving problems with multidimensional variables given the available computer power (Powell, 2007; Papageorgiou et al., 2014; Simão et al. (2009, 2010). Indeed, multistage planning models have undergone severe difficulties when it comes to broaden the range of possible uncertainties, decision variables, time stages as well as possible outcomes. This is due to the fact that current optimizers (e.g. CPLEX, Gurobi) and decomposition techniques are not able to reach or even to be close to an optimal solution within acceptable time limits (Cristobal et al., 2009; Pantuso et al.,2014, 2015). In consequence, researchers have been prevented from applying their models to realistic case studies (e.g. Hsu et al., 2011). Therefore, the main challenge to be addressed consists in finding an efficient solving technique against the well-known dimensionality problems faced by dynamic programming.

In light of this, current research streams have revolved around this concern. While some academic groups have focused on the development of powerful heuristic methods to solve planning problems (e.g. List et al., 2003), other communities have moved towards a new research trend, which has gained a lot of ground in different communities such as control theory, artificial intelligence and operations research. Even though the vocabulary and terms used were different for each community, the idea was still the same: *stepping forward* through time across the scenario tree, thereby avoiding backward induction and recursion. As for its notation, this method has been named in many different ways across communities: neuro-dynamic programming for control theory community, reinforcement learning for the artificial intelligence community and many other publications have refer to it as approximate dynamic programming, forward dynamic programming, adaptive dynamic programming, heuristic dynamic programming or iterative dynamic programming. According to Powell (2007), the term that has been increasingly accepted within the operations research community is approximate dynamic programming (ADP). Thus, the present MSc thesis will use this term hereinafter.

In order to *step forward* through time and avoid backward induction across the scenario tree, approximate dynamic programming consists in initially approximating the recursive value function V_t for all time stages and then, improve these estimations through subsequent Monte Carlo simulations until reaching a near-optimal policy. Analysing this general concept, ADP can be considered as an optimising simulator since it offers a solving strategy based on the flexibility of simulations and the power of optimisation together with feedback learning (Powell et al., 2014). This concept makes ADP a novel and promising method to solve multistage stochastic problems, whose computation have appeared to be so far intractable with backwards exact methods. Indeed, ADP allows to decompose large-scale problems and reduce the required computational times. In the same vein, Sutton and Barto (1998) proposed a similar algorithmic strategy to solve problems in the field of control. They referred to it as reinforcement learning since the algorithmic approach was based on learning-by-doing: at each iteration the approximations of the value function were adjusted and improved, thereby approaching progressively to the optimal solution.

In any case, there are still many challenges to tackle in terms of value function approximation methods, updating techniques and scenario tree exploration policies. Indeed, ADP is a flourishing discipline whose foundations have been studied from a mathematical perspective during the last two decades. This may explain the scarcity of ADP-based applications in current industry. Indeed, well-known problems in literature such as the newsvendor problem or the multicommodity-flow problem were applied at small-scale by Godfrey and Powell (2001, 2002a, 2002b) and Topaloglu and Powell (2006) so as to verify approximation techniques against exact methods. In these publications, several approximation methods were introduced and tested at different levels of complexity: a linear value-function approximation, a piecewise-linear value-function approximation as well as a hybrid value-function approximation. A restricting feature of their proposed methods was that they only could deal with concave functions. In any case, the obtained numerical results looked promising since they provided near-optimal solutions within acceptable computational times. Figure 2.6 shows the typical performance of three approximation methods applied to a time-staged integer multicommodity flow problem (Topaloglu and Powell, 2006). The hybrid (PL) and piece-wise linear (P) value-function approximation techniques achieved successful results within the 1%-2% of the objective upper bound, whereas the linear (L) approximation fluctuated more but still remained within the 10%.

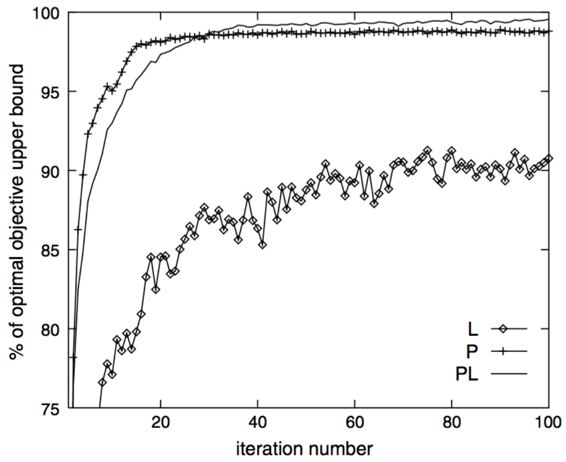


Figure 2.6: Typical performances for linear (L), piece-wise linear (P) and hybrid (PL) value-function approximations for a deterministic multicommodity flow problem (Topaloglu and Powell, 2006)

Given the clear potential of approximate dynamic programming, from 2000s onwards there has been a rising interest in the application of ADP to more realistic and larger operational problems under uncertainty. In the domain of operations research, some of the diverse problems already studied have been mainly tactical such as the vehicle routing problem under demand uncertainty (Secomandi, 2000), maritime inventory routing problems (Papageorgiou et al., 2014), the empty container allocation problem (Lam et al., 2007) or the resource allocation problem in hospitals (Hulshof et al., 2016). Moreover, Simão et al. (2009, 2010) proved that an ADP algorithm could capture the large-scale fleet management operations for Schneider National. Indeed, their model was capable of optimising over time a highly detailed system with an extremely high number of decision variables. Furthermore, the objective of Simão et al. (2009, 2010) was not to provide a better solution, but rather to closely match several operational statistics to validate the simulation. From a planning perspective, one of the most important contributions has been the implementation of an optimising simulator to support locomotive planning at an American rail transport company (Powell et al., 2014). In their work, Powell et al. (2014) used the ADP modelling and algorithmic framework to create a highly detailed model that supported fleet sizing decisions based on historical metrics. The obtained policies proved to be near-optimal, robust and adaptive to different scenarios. Thus, ADP appeared to be effective for the cases studied. To the writer's knowledge, this is the first time in which ADP has been employed taking a strategic planning perspective rather than a tactical one.

In conclusion, the recently obtained results with ADP have shed light on the computation of dynamic models and thus, the solution of multistage fleet planning models. Yet, there is still a wide range of possibilities for research.

2.3. Fleet planning and market interactions

The fleet planning and other relevant models introduced until now focused on the performance of a single company in a deterministic or stochastic environment. That is to say, the company performance depended only on their decisions and the environment conditions (demand, disruptions and operating costs), without being influenced by the possible presence of other companies competing for the same market. Yet, the possible presence of players in the market could bring a higher degree of uncertainty.

For many years, competition in the airline industry was investigated as a central problem by means of applying game theory in several forms such as simultaneous games and two-stage games. This allowed the comprehension of different competitive behaviors (e.g. Wei and Hansen, 2007; Adler, 2001). Furthermore, many researchers (Simpson, 1970; Gelerman and De Neufville, 1973; De Neufville and King, 1979; Powell, 1982; Wei and Hansen, 2005) contributed to the study of demand-supply interactions under competition. Nevertheless, it is evident that the integration of competition models into planning models has been enormously constrained by the limited computational power available. In fact, modelling any kind of interaction between independent entities involves a coupling expression, which is prone to entail nonlinear programs that need to be solved with heuristic method as shown by Teodorović and Krcmar-Nozić (1989), Wei and Hansen (2007) and Wang et al. (2015). What is more, computing complexity would increase dramatically if demand uncertainty was to be added. Consequently, this fact may explain the reason for which competition elements have not been included yet in stochastic planning models.

2.4. Conclusions: Knowledge gap

Once reviewed the existent literature related to fleet planning, the mind map technique is a useful way to structure all relevant information and understand where the current knowledge gap is. Figure 2.7 shows the current research status of fleet planning in different areas: core modelling elements, problem extensions and optimisation methods. Blue areas correspond to elements frequently included in publications, orange areas to elements partially seen in publications and yellow areas to issues barely or never studied in existent literature related to fleet planning.

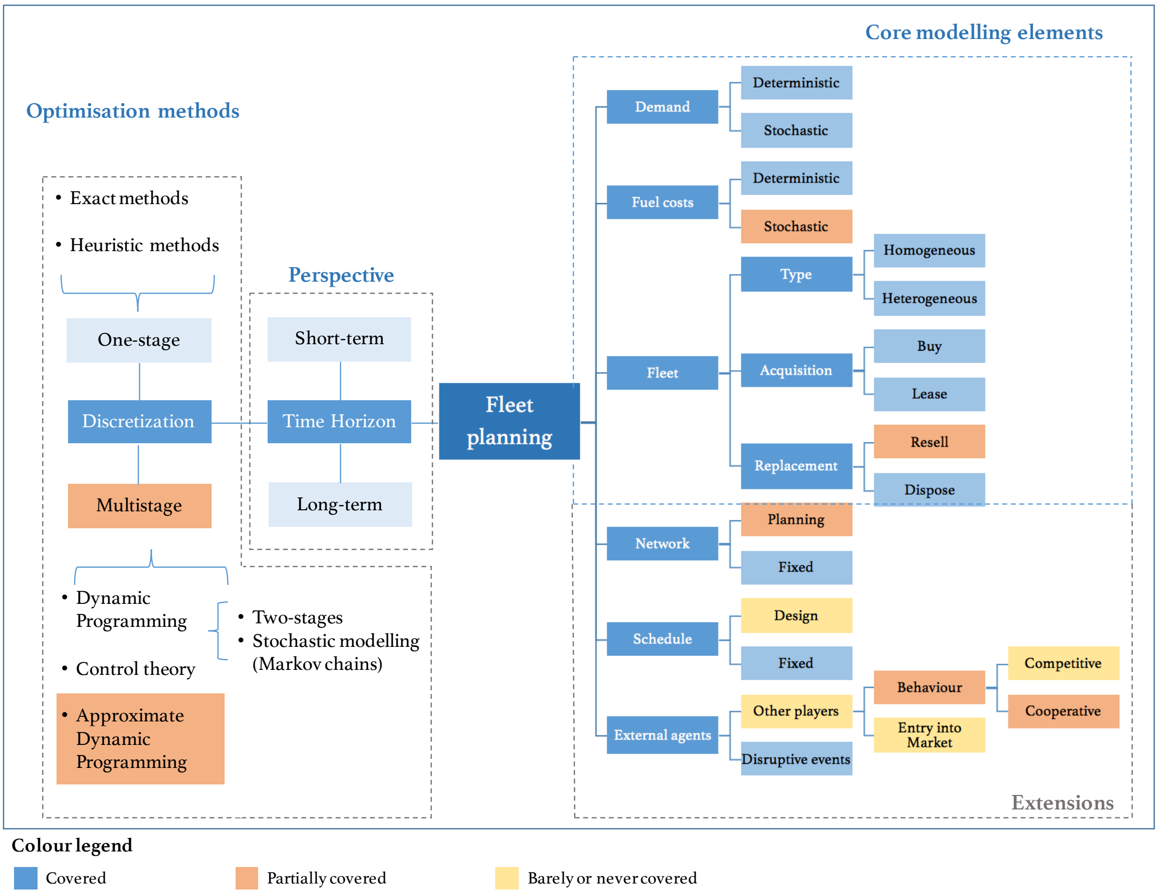


Figure 2.7: Knowledge gap mind map

Fleet planning models have not experienced great modelling changes with respect to the first contribution of Shube and Stroup (1975). Indeed, the major part of models do not go beyond the representation of the core defining elements of the problem: demand, operating costs, a portfolio of routes, fleet characteristics and replacement decisions. Yet, a medium-size multistage deterministic version of the problem may already lead to a large-scale integer program challenging to solve (e.g. Sayarshad and Ghoseiri; 2009).

Taking into account the importance of including uncertainty into fleet planning models, demand has frequently been tested as a random variable in both two-stage and multistage stochastic models (e.g. Hsu et al., 2011; Khoo and Teoh, 2014). The uncertainty of fuel costs has also been studied following a two-stage structure (e.g. Naumann and Suhl, 2013) but not in a multistage structure yet. Apart from that, the simultaneous analysis of demand and fuel uncertainties needs to be further developed. In fact, current studies have either focused on the effect of demand uncertainty or fuel uncertainty without analysing their interrelations.

While the transition from deterministic to stochastic models was an essential step to develop more robust fleet plans under uncertainty, the consideration of different scenarios has extremely complicated the computation of real-sized applications. Consequently, a lot of academic effort has been dedicated to develop efficient dynamic programs and heuristic methods to deal with this challenge. Nowadays, approximate dynamic programming seems a promising field of study, which has not yet been tested in the context of airline fleet planning. Its good results in the locomotive domain position ADP as a possible method to tackle the dimensional constraints of multistage stochastic fleet planning models. If implemented sucessfully, ADP could pave the way for developing more realistic fleet planning models. Indeed, the inclusion of extensions in fleet planning models may allow to capture better reality and help airlines to take better strategic decisions.

Until now, it is evident that the integration of any extensions into the fleet planning problem has been constantly limited by the current computational power available. In general, an increase in the level of detail leads to an unmanageable number of possible scenarios to explore. Nevertheless, this has not prevented researchers from testing in small-scale cases the inclusion of fleet management issues into fleet planning models (e.g. Hsu et al., 2011; Khoo and Teoh, 2014). Likewise, competition and demand-supply models have been studied as possible extensions to frequency planning (e.g. Teodorović and Krcmar-Nozić, 1989) or more recently, as extension to the fleet planning problem (e.g. Wang et al., 2015). In any case, modelling any kind of interaction between independent entities involves a coupling expression, which is prone to entail multi-objective nonlinear programs that need to be solved with heuristic methods as shown by Teodorović and Krcmar-Nozić (1989), Wei and Hansen (2007) and Wang et al. (2015). On top of that, adding uncertainty to this kind of problems would do nothing more than increase dramatically their computational complexity. All these reasons may explain why extensions have not been included yet in stochastic planning models.

As a result, the multistage fleet planning models proposed until now focused on the performance of a single company in a deterministic or stochastic environment. That is to say, the company performance depended only on its decisions and the environment conditions (demand, disruptions and operating costs), without being influenced by the possible presence of other companies competing for the same market. This is quite unrealistic since the possible presence of players in the market can bring a higher degree of uncertainty. Yet, competition uncertainty has not been explored from all perspectives. As it can be observed, many researchers have focused solely on modelling global competitive behavior with game theory and have come up with complex multiobjective programs difficult to couple with stochastic planning models.

Nevertheless, competition could be modelled from the perspective of a single airline in a more simple and useful way, which would allow the integration of competitive elements into fleet planning models. Instead of using game theory, the impact of competition on fleet plans could be modelled as the uncertainty entailed by the probability of other airline entering into a market. Indeed, an airline may invest in its fleet expansion so as to exploit a promising market with high growth rates. Nevertheless, any optimistic forecasted results may increase the probability of other airlines entering into the same market, thereby leading to a fleet surplus and its corresponding losses. This new idea would still capture the relevant effects of competition in fleet planning while keeping the model dimensions within acceptable limits.

In view of the above, it can be concluded that the development of multistage stochastic fleet planning models has been enormously constrained by the limitations of dynamic programming on solving wide scenario trees. Nevertheless, the implementation of ADP represents a promising knowledge gap in the field. In fact, no airline fleet planning model has been based on this discipline before and results in the locomotive industry have proven to be successful. In other words, its effective implementation would mean a step forward towards the development of multi-period adaptive fleet planning policies in the airline industry. Finally, it could also represent a viable solving technique for more complex planning models, which may eventually combine demand, fuel and competition uncertainty.

3 Research Scope

Given the current knowledge gap existent in fleet planning literature, the aim of this chapter is to establish the research boundaries of this MSc thesis. Firstly, the research objective is defined in Section 3.1. Next, Section 3.2 introduces a research framework, which will help formulate the corresponding research questions in Section 3.3. To answer these questions and achieve the research objective, a experimental set-up will be detailed in Section 3.4. Finally, Section 3.5 discusses the expected outcome and relevance of this MSc thesis.

3.1. Research objective

The objective of this MSc thesis is to contribute to the development of adaptive policies in the context of airline fleet planning under demand uncertainty by (a) modelling and solving with Approximate Dynamic Programming a multi-period adaptive fleet planning problem that integrates stochastic demand and by (b) detecting useful signposts for fleet planners.

3.2. Research framework

In order to achieve the research objective, it is convenient to draw a research framework that permits the identification of the essential pillars to follow during the project. Its scheme is outlined in Figure 3.1. Consequently, this technique facilitates the formulation of research questions as well as the definition of the project plan.

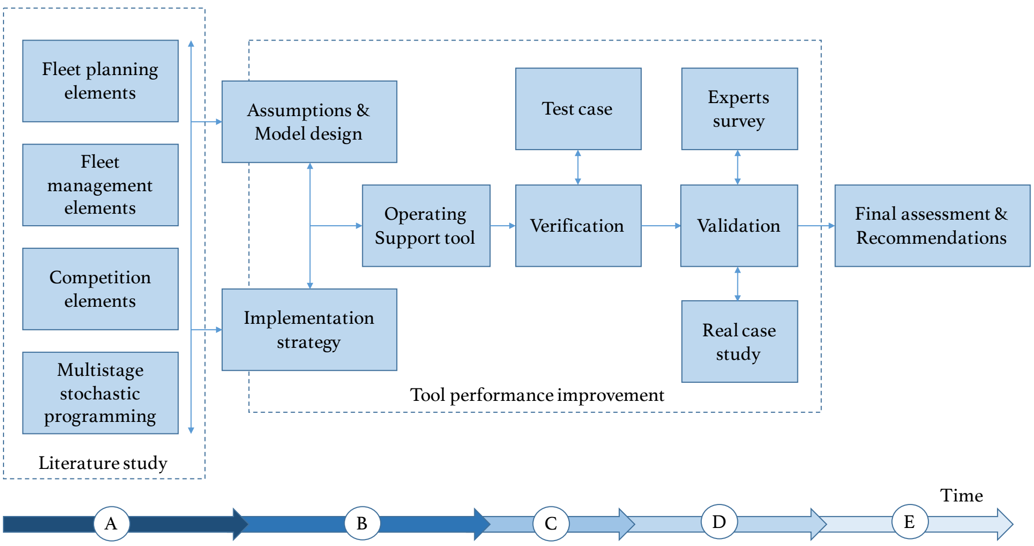


Figure 3.1: MSc thesis research framework

The research framework can be explained and structured as follows: (A) An initial literature study is carried out so as to identify the main elements of all relevant disciplines to the problem, define a set of assumptions and design the mathematical model. (B) A trade-off is performed between the complexity of the model and the solving implementation strategy so as to create an ADP-based decision support tool. (C) The verification criteria is defined and the developed tool is tested with a small case. (D) The tool validation is performed by means of a real case study and an experts survey. Furthermore, it must be noted that phases B, C and D are interrelated by the simultaneous improvement of the tool performance. (E) The final project stage consists in providing a final assessment of the obtained ADP-based tool and recommendations for future research.

3.3. Research questions

Research questions are formulated based on the structure of the research framework.

(A) What should be the main characteristics of a multistage stochastic model that provides an adaptive policy for fleet planning?

- (a.1) What are the essential elements that model a fleet planning problem?
- (a.2) Should fleet management be integrated to the fleet planning problem and if so, at which level of detail?
- (a.3) How can the model account for market interactions?
- (a.4) What are the most appropriate solving techniques to solve the problem?

(B) What is an efficient way to implement the model?

- (b.1) Has the designed model any programming constraints that should be taken into account?
- (b.2) Amongst all candidate solving techniques, which one should be chosen?
- (b.3) What are the main advantages of the programming language used?

(C) Can the already operating tool be verified?

- (c.1) What is the most appropriate small test to verify the behaviour of the operating tool?
- (c.2) What are the relevant criteria to assess the quality of the obtained results from the small test case?
- (c.3) Which model and algorithm parameters should be analysed with a sensitivity analysis?
- (c.4) Does the support tool feature any weakness that can affect the quality of the results?

(D) Can the already verified model be validated?

- (d.1) Is there a real case study that could validate the model and if so, to what extent can the model capture reality?
- (d.2) What is the evaluation of fleet planning experts concerning the model?

(E) Has the support tool contributed to the development of adaptive fleet plan policies?

- (e.1) Does the model feature any advantage with respect to the current available ones?
- (e.2) Which aspects need to be improved?
- (e.3) Is the support tool practical to support decision-making?

3.4. Experimental Set-up

The proposed MSc thesis implies experiments which will mainly consist in carrying out computer simulations and expert surveys within the air transport industry.

As far as computer simulations are concerned, the experimental setting can be defined by the hardware and software used. In terms of hardware, it will be used a MacBook Pro consisting of one processor Intel Core i5 with 2.7 GHz speed and 16 GB 1867MHz RAM memory. Although the implementation of approximate dynamic programming requires a lot of computational memory, this computer is expected to have enough power to meet the MSc thesis objective. When it comes to software, Python 2.7 is chosen as programming language with the aim of facilitating a collaboration with industry experts in the validation phase (D). Given the fact that Python is open-source but still very powerful, the developed decision support tool can easily be tested within a company. Moreover, Python will be installed within Anaconda, which is a Python open-source distribution and environment manager including a vast collection of Python extensions (SciPy, PySP, Pandas, Matplotlib,etc.) as well as useful interfaces such as Spyder and IPython. Apart from this, the commercial optimiser Gurobi will be used due to its efficient interface with Anaconda and well-documented Python API. Despite its commercial purposes, Gurobi can provide free licenses to students and young graduates. Apart from this, other open-source Python packages such as ete3 will be employed for specific modelling purposes.

In the verification stage, the input data for the test case will be taken from previous work of the Air Transport Operations (ATO) Department of TU Delft Aerospace Faculty. This will allow a reliable evaluation of the algorithm performance when solving the test case. After having verified the model, the validation stage will require the use of real input data from a reference airline. This will be possible thanks to the collaboration of the ATO department with Kenya Airways. Furthermore, validation experiments will be carried out right in the field so as to interview industry experts and gather realistic feedback on the tool.

3.5. Results, Outcome and Relevance

Since fleet planning is an optimisation problem, the data will be structured in an objective function to be maximised upon a set of different constraints. In this way, the data will consist of variables modelling the system: an airline deciding on when and how many aircraft of each type to buy in order to maximise profits while meeting as best as possible a stochastic demand. Depending on its main function, variables will be classified as parameters and decision variables:

- **Parameters:** will define the main characteristics and constraints of the system. For instance, they will represent necessary information such as airline route fares, fixed and operating costs aircraft characteristics or the stochastic demand along different time periods.
- **Decision variables:** represent the set of strategic decisions to make in order to maximise the objective function. In this case, the decision variables will consist of the number of aircraft of each type to buy or discard for different time periods and demand scenarios. When optimality is reached, decision variables become the results for the problem given a set of predefined parameters. Decision variables are expected to be in an integer form. Nevertheless, they could be considered continuous in order to relax the problem and reduce its computational requirements.

As it can be concluded, the solution to an approximate dynamic programming problem is a random variable $\bar{x}_t(\xi_{[t]})$ since its value depends on a random process $\xi_{[t]}$ defined by a stochastic demand. Therefore, instead of providing a single best solution, approximate dynamic programming offers the best rule for making decisions in function of the available information at that time. This rule is what is known as policy (Powell, 2007). Therefore, the main outcome is supposed to be an adaptive fleet plan policy capable of giving recommendations that adapt to every possible demand scenario. The impact of obtaining such an adaptive policy is very relevant since it would mean a strong basis for eventually building a reinforced learning system to support fleet planning decisions. Most importantly, the relevance of this MSc thesis lays on proving approximate dynamic programming as an effective method to solve the multi-period adaptive fleet planning problem under demand uncertainty. In fact, its successful implementation could boost the development of currently stagnated fleet planning models, thus paving the way for more sophisticated tools combining demand, fuel and competition uncertainty.

4
Methodology

Chapter 4 discusses the methodology followed to achieve the proposed research objective. To this end, Section 4.1 introduces the required mathematical background to understand the concept of approximate dynamic programming within the context of the multi-period adaptive fleet planning problem. Next, in Section 4.2 the problem will be formulated as a dynamic program, which will enable the effective implementation of approximate dynamic programming. Finally, Section 4.4 will present the ADP algorithm employed to solve the multi-period adaptive fleet planning problem as well as the strategies created for its implementation.

4.1. Mathematical background for approximate dynamic programming

Before modelling any problem, it is very important to have a deep understanding of its main mathematical essence as well as the fundamentals of its possible optimisation methods. Carrying out this exercise facilitates the identification of problem distinctive characteristics and consequently, their implication in terms of mathematical modelling and optimisation. Apart from that, a single problem can be modelled and solved in multiple ways.

In light of this, the present section will begin with a brief analysis of fleet planning in the context of sequential decision problems and dynamic programming. This will be followed by a description of a state-space framework suitable for modelling dynamic programs. Lastly, the concept of approximate dynamic programming will be described together with its expected contribution to the multi-period adaptive fleet planning problem.

4.1.1. Multi-period adaptive fleet planning as a sequential decision problem

The multi-period adaptive fleet planning problem is a long-term strategic problem faced by all transportation companies, which mainly consists in deciding the quantity and composition of vehicles to be acquired and disposed along the future years in order to maximize profits by meeting future uncertain demand. Therefore, this problem can be seen as a sequential decision process over time periods and under uncertainty.

Throughout the years and in different industries (e.g. finance, production, energy, transportation and logistics), sequential decision problems under uncertainty have been modelled by means of multistage stochastic models. These models can be seen as the natural extension of two-stage stochastic models over $T+1$ time periods $t = 0, 1, ..., T$. Thus, the inputs of a multistage stochastic model are multivariate random processes $\xi_{[t]} := (\xi_1, \xi_2 ..., \xi_T)$, whose values over time t, ξ_t, are given by a certain probability function $\mathbb{P}$. If the values taken at each stage ξ_t are stochastically independent from the previous values ξ_{t-1}, the process is considered stage-wise independent and is commonly called Markov Chain. In general, multistage models cover decision processes in the form of Figure 4.1 (Shapiro et al., 2014). Decisions chosen at a time period t, x_t, are only made upon observations ξ_t available up to time t and not on future observations (nonanticipativity). The key point is that current decisions will have an impact on future problem conditions, which at the same time are still uncertain.

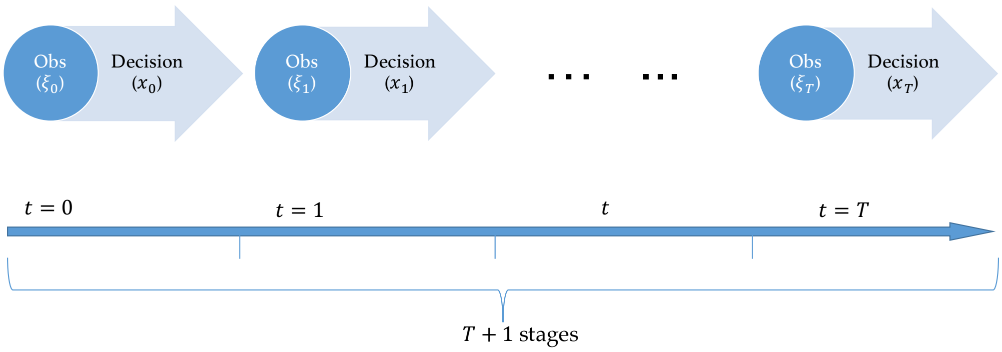


Figure 4.1: Decision process scheme elaborated from Shapiro et al., (2014)

As far as airline fleet planning is concerned, it can be stated that a Markov chain model would grasp the full essence of the strategic problem. Decisions to acquire or dispose aircraft at a particular time t are made upon current airline performance and available demand forecasts, which can be treated as Markov chain observations at time t. Apart from this, current decisions to acquire or dispose certain aircraft will not only impact current but also future operational performance of the airline.

By extending the general formulation of a two-stage stochastic problem and using the notation of Shapiro et al. (2014), a T-stage model can be expressed in a nested general form:

$$\underset{x_0 \in \chi_0}{\text{Min}} \quad f_0(x_0, \xi_0) + \mathbb{E}\left[\inf_{x_1 \in \chi_1(x_0,\xi_1)} f_1(x_1, \xi_1) + \mathbb{E}\left[... \quad + \quad \mathbb{E}[\inf_{x_T \in \chi_T(x_{T-1},\xi_T)} f_T(x_T, \xi_T)]\right]\right] \tag{4.1}$$

where χ_t represents the region of feasible solutions for x_t defined by the set of constraints of the t-stage subproblem. Likewise, these constraints are determined by a set of parameters which are function of previous decisions x_{t-1} and the most recent observations of the stochastic variable ξ_t. However, the idea remains the same as in the two-stage stochastic problem: find the optimal actions x_t which will not only optimize (in this case minimize) the current function f_t, but also the expected function value for the next stage $t+1$. Similarly, in the airline fleet planning problem the objective is to maximize current airline profits as well as the expected profits for the next uncertain years:

$$\sup_{x_0 \in \chi_0} \quad \text{Profit}_0(x_0) + \mathbb{E}\left[\sup_{x_1 \in \chi_1(x_0,\xi_1)} \text{Profit}_1(x_1, \xi_1) + \mathbb{E}\left[... \quad + \quad \mathbb{E}[\sup_{x_T \in \chi_T(x_{T-1},\xi_T)} \text{Profit}_T(x_T, \xi_T)]\right]\right] \tag{4.2}$$

Nevertheless, these cumbersome nested formulations can be simplified by taking advantage of their recursive mathematical structure. This is depicted by Equation 4.3, which is commonly known as Bellman equation or dynamic programming equation:

$$V_t = \underset{x_t \in \chi_t(x_{t-1},\xi_t)}{\text{Min (or Max)}} \quad \left(\quad f_t(x_t, \xi_t) + \mathbb{E}\{V_{t+1}\} \quad \right) \tag{4.3}$$

In this expression, V_t is called value function and denotes the value of taking a series of optimal decisions $x_{[t]}$ from period t onwards. As inferred from Equation 4.3, the value function V_t is defined mathematically by its own definition. Following this logic, it can be drawn that the value function in the initial period V_0 is equal to the objective function of the entire problem.

This recursive relationship between value functions is very distinctive of any multi-period planning problem and represents the basis of dynamic programming. Indeed, dynamic programming is a well-known mathematical optimisation method which consists in tackling the entire recursive problem V_0 by breaking it into smaller nested subproblems V_t which overlap throughout the time periods and need to be solved at least once. Therefore, it can be concluded that the multi-period fleet planning problem can be modelled and solved intuitively as a dynamic program.

4.1.2. Framework for modelling dynamic programs

When it comes to solving dynamic programs such as the multi-period fleet planning problem, it is very meaningful to firstly adopt a sound modelling framework so as to take full advantage of the problem structure and solving strategies. Indeed, the way in which a problem is modelled plays an essential role during the algorithm implementation of the optimisation method. Whithin the operations research community, Bertsekas and Tsitsiklis (1995a, 1995b) and Powell (2007) led the way in the use of a simple but very powerful framework to represent dynamic programs efficiently. Based on control theory, this framework consists in a state-space representation of the problem and is composed by six essential elements:

- **State vector (S_t),** which contains all necessary information to take decisions and describe the system evolution over time.
- **Decision vector (a_t),** which can be understood as the controller actions of the process.
- **Exogenous information ($\omega_{[t]}$),** which represents a stochastic process whose random variables ω_t are revealed gradually at the start of each time stage.
- **Transition function** ($S_{t+1} = S^{Model}(S_t, a_t, \omega_t)$), which indicates the manner in which the system evolves from the state S_t to the state S_{t+1} in function of the previous decisions made a_t and the most recent realisation of exogenous information ω_t.
- **Contribution function ($C_t(S_t, a_t)$),** which indicates the costs incurred or rewards received based on the decisions made during each time stage.
- **Objective function,** where Powell explicitly states the maximisation (or minimisation) of all aggregate contributions C_t along a certain time horizon T.

Using Powell's framework, the Bellman equation can be rewritten as follows:

$$V_t(S_t) = \underset{a_t \in \mathscr{A}_t}{\text{Max}} \quad \left(C_t(S_t, a_t) + \gamma \mathbb{E}\{V_{t+1}(S_{t+1}(S_t, a_t)) \mid S_t\} \right) \tag{4.4}$$

where the state variable S_t contains the exogenous information ω_t and γ denotes a time discount factor. Furthermore, the conditional expectation is defined as:

$$\mathbb{E}\{V_{t+1}(S_{t+1}(S_t, a_t)) \mid S_t\} = \sum_{s' \in \mathscr{S}} \mathbb{P}(S_{t+1} = s' \mid S_t, a_t) V_{t+1}(s')) \tag{4.5}$$

where $\mathbb{P}$ corresponds to the probability of next state S_{t+1} becoming s' once actions a_t are taken under current state S_t. Finally, the objective function of the problem becomes the maximisation of the initial contribution function together with the expected value of future contributions functions throughout next uncertain periods:

$$V_0(S_0) = \underset{a_0 \in \mathscr{A}_0}{\text{Max}} \quad \left(C_0(S_0, a_0) + \gamma \mathbb{E}\{V_1(S_1(S_0, a_0)) \mid S_0\} \right) \tag{4.6}$$

Given this framework for modelling dynamic programs, the multistage stochastic problem can also be schematized with a scenario tree representation, as depicted in Figure 4.2. The circles represent the different possible states describing the system at time stage t. In particular, the purple circle denotes the initial state of the problem, whereas the green circles its future possible states. As inferred from the definition of the transition function, these future states result from previous random realizations of exogenous information ω_t, which are represented by the discontinuous branches. Finally, squares represent the moments in which decisions must be taken depending on future forecasts of exogenous information as well as the current performance indicated by the contribution function. As highlighted with a yellow shadow, every sequence of events (here $\{\omega_1^0, \omega_2^0, \omega_3^0\}$) linking the initial with the final states (here S_0 and S_3) will correspond to each specific scenario that can take place in the problem. In any case, the reader should note that Figure 4.2 represents a discrete representation of the problem; however, the number of possible scenarios could also be infinite.

All in all, this state-space representation of dynamic problems is commonly seen in operations research and as will be discussed shortly, it represents the modelling basis for other emerging disciplines such as approximate dynamic programming.

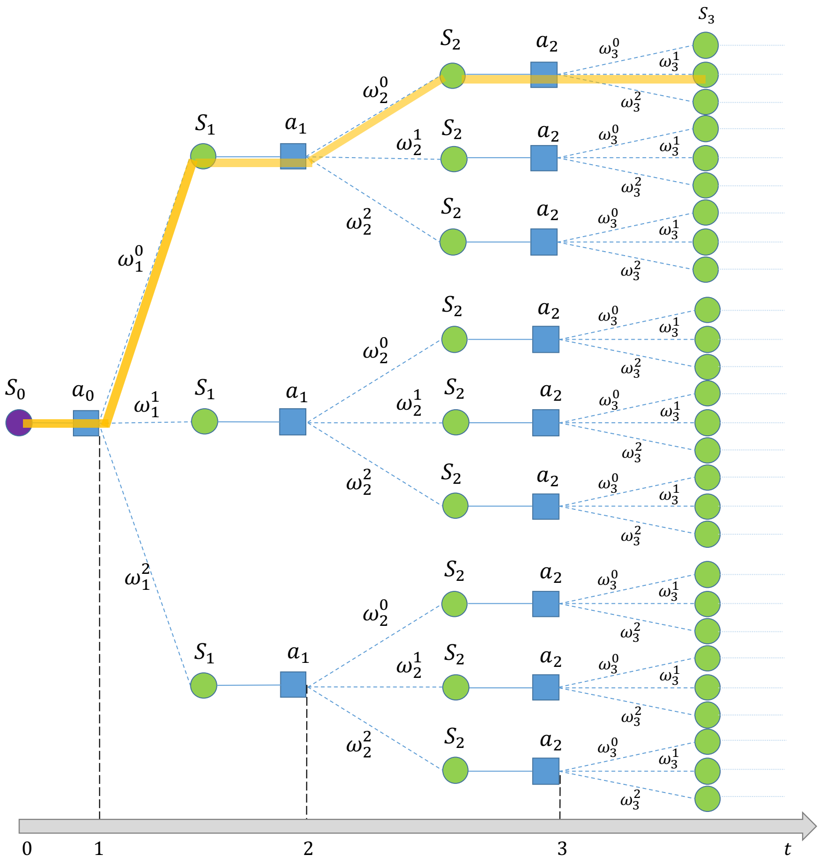


Figure 4.2: Scenario tree representation for multi-period fleet planning problems

4.1.3. Foundations of approximate dynamic programming

Even though there are many computer algorithms based on dynamic programming , one of the most common and intuitive techniques to solve the Bellman equation has been backward induction. As its name implies, backward induction consists in optimising the problem backward in time. That is to say, the optimisation starts by solving the value function at the final stage to then determine the series of previous optimal actions. In this way, all possible value functions need to be exactly computed at least once in order to make decisions. Hence, this method is specially intended for solving finite-horizon discrete-time dynamic programs.

Following the logic of backward induction, one can start considering the last-stage subproblem ($t = T$):

$$\underset{a_T \in \mathscr{A}_T(a_{T-1}, \omega_T)}{\text{Max}} \quad C_T(S_T, a_t) \tag{4.7}$$

whose optimal solution is denoted by $V_T(S_T)$. Then, for the earlier stages $t = 0, 1, ..., T-1$ the problem is written as:

$$\begin{aligned} V_t(S_t) &= \max_{a_t \in \mathscr{A}_t} \left(C_t(S_t, a_t) + \gamma_{t+1} \mathbb{E}\{V_{t+1}(S_{t+1}) | S_t\} \right) \\ &= \max_{a_t \in \mathscr{A}_t} \left(C_t(S_t, a_t) + \gamma_{t+1} \sum_{s' \in \mathscr{S}} \mathbb{P}\left(s' | S_t, a_t\right) V_{t+1}(s') \right) \end{aligned} \tag{4.8}$$

Moving backwards across the scenario tree as shown in Figure 4.3, the problem for the initial stage can be eventually solved:

$$\begin{aligned} V_0(S_0) &= \max_{a_0 \in \mathscr{A}_0} \left(C_0(S_0, a_0) + \gamma_1 \mathbb{E}\{V_1(S_1) | S_0\} \right) \\ &= \max_{a_0 \in \mathscr{A}_0} \left(C_0(S_0, a_0) + \gamma_1 \sum_{s' \in \mathscr{S}} \mathbb{P}\left(s' | S_0, a_0\right) V_1(s') \right) \end{aligned} \tag{4.9}$$

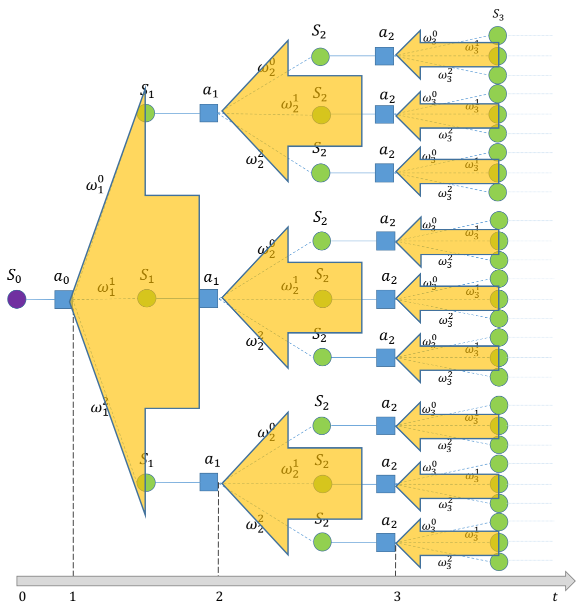


Figure 4.3: Scheme of backward induction across scenario tree.

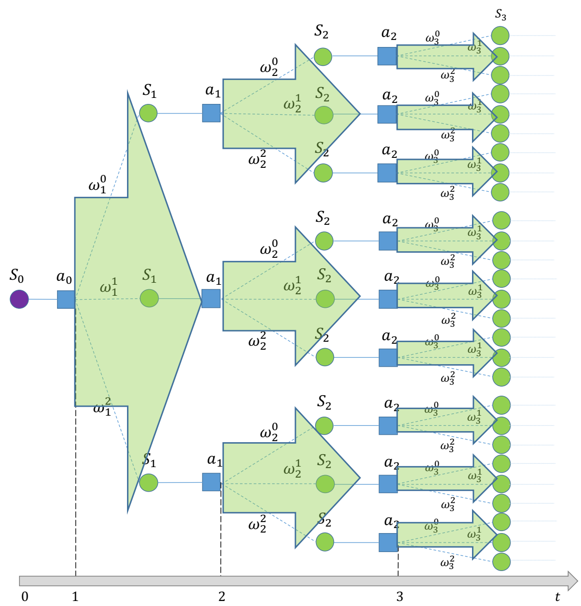


Figure 4.4: Scheme of forward induction across scenario tree.

Nevertheless, it was concluded from the literature review that this technique only remains effective for small scenario trees and few stages (Hsu et al., 2011; Khoo and Teoh, 2014; Pantuso et al., 2014, 2015)). More particularly, it was discussed that dynamic programs suffer from *the curses of dimensionality* due to their recursive nature and thus, large multistage stochastic problems become intractable with backward induction (Powell; 2007). This is due to the fact that backward induction requires so high computational efforts and storing memory that current computer capabilities are frequently exceeded. Consequently, many stochastic problems formulated as dynamic programs are constrained in terms of dimensions. It is precisely under these conditions that approximate dynamic programming plays an important role. Using the notation of Powell (2007), the foundation of ADP is explained hereunder.

As previously introduced in Chapter 2, the idea of approximate dynamic programming lays on an algorithmic strategy which allows to *step forward* throughout the time periods. This novel concept breaks with the common tendency of solving dynamic programs by means of backward induction and recursion. However, its modelling framework remains the same state-space representation used in dynamic programming, whose suitability has been proven by many papers implementing ADP algorithms (Godfrey and Powell, 2001, 2002a, 2002b; Topaloglu and Powell, 2006; Secomandi, 2000;Papageorgiou et al., 2014; Lam et al., 2007; Hulshof et al., 2016; Simão et al., 2009, 2010; Powell et al., 2014).

Figure 4.4 outlines the idea of approximate dynamic programming: *stepping forward* across the scenario tree. This is achieved by initially approximating the next value function V_{t+1} so as to make decisions a_t:

$$\begin{aligned} V_t(S_t) &= \max_{a_t \in \mathscr{A}_t} \left(C_t(S_t, a_t) + \gamma_{t+1}\mathbb{E}\{V_{t+1}(S_{t+1})|S_t\}\right) \\ &\approx \max_{a_t \in \mathscr{A}_t} \left(C_t(S_t, a_t) + \gamma_{t+1}\mathbb{E}\left\{\overline{V}_{t+1}(S_{t+1})|S_t\right\}\right) \end{aligned} \tag{4.10}$$

where $\overline{V}_{t+1}$ denotes an approximation of the next value function V_{t+1}. Once obtained the initial approximated values $\overline{V_t^0}$ at an initial iteration $n = 0$, one could start solving the Bellman equation for the next iteration $n = 1$:

$$\hat{v}_t^1 = \underset{a_t^1 \in \mathscr{A}_t^1}{\text{Max}} \left(C_t(S_t^1, a_t^1) + \gamma_{t+1}\mathbb{E}\left\{\overline{V}_{t+1}^0(S_{t+1}^0) \mid S_t^0\right\}\right) \tag{4.11}$$

In this case, a_t^1 would be the value of a_t that solves the approximated maximisation problem for period t in iteration $n = 1$ and $\hat{v}_t^1$ its corresponding optimal value observed. When stepping forward through time stages, a random scenario is chosen. That is to say, a Monte Carlo simulation will dictate a sample path $\omega^1_{[t]}$, which will indicate the realisation of the next random variables. Likewise, the system evolution is captured by the transition function:

$$S_{t+1} = S^{Model}(S_t, a_t, \omega_{t+1}) \tag{??}$$

In this way, a new set of values $\hat{v}^1_t(S_t)$ are generated throughout the time periods based on the initial approximation, and will be used to update the former approximated values of the scenario studied. If the scenario chosen by the Monte Carlo simulation does not pass through certain nodes, then the value at these nodes will remain with their previous approximated values:

$$\overline{V}^n_t(S_t) = \begin{cases} F^{\text{Update}}(\hat{v}^n_t), & \text{if} \quad S_t = S^n_t \\ \overline{V}^{n-1}_t(S_t), & \text{otherwise} \end{cases} \tag{4.12}$$

Value function observations are generally updated taking the form of a weighted function between prior value function observations and the most recent observation:

$$\overline{V}^n_t(S^n_t) = (1-\alpha_{n-1})\overline{V}^{n-1}_t(S^n_t) + \alpha_{n-1}\hat{v}^n_t \tag{4.13}$$

where α_{n-1} is called stepsize and defines the rate at which the value function approximation is updated throughout the iterations. Its value ranges between [0,1], being 1 the fastest updating rate and 0 the cancellation of the value function update.

Once all approximated values are updated, a new iteration n starts. A new Monte Carlo simulation is performed to determine the new scenario path $\omega^n_{[t]}$ to analyse and the previously explained process starts again by solving Equation 4.14:

$$\hat{v}^n_t = \underset{a_t \in \mathscr{A}^n_t}{\text{Max}} \left(C_t(S^n_t, a^n_t) + \gamma_{t+1}\mathbb{E}\left\{ \overline{V}^{n-1}_{t+1}(S^n_{t+1}) \mid S^n_t \right\} \right) \tag{4.14}$$

Furthermore, it should be noted that the number of iterations required (N) depends on problem dimensions and the value function approximation. Generally speaking, the closer the approximation technique is to the real function, the less iterations the algorithmic strategy needs to reach convergence (Topaloglu and Powell, 2006).

In conclusion, approximate dynamic programming can be considered as an optimising simulator since it offers a solving strategy based on the flexibility of Monte Carlo simulations and the power of OR optimisation, both combined with machine learning (Powell et al., 2014). This blend of disciplines makes ADP a novel and promising method to solve multistage stochastic problems, whose computation have appeared to be so far intractable with backwards exact methods. Indeed, ADP allows to decompose large-scale problems by approximating unknown value functions, thus reducing the required computational times needed in dynamic programming.

4.2. The multi-period adaptive fleet planning model

Having discussed the dynamic essence of strategic fleet planning and the suitability of the state-space framework to capture its mathematical characteristics, this section focuses on modelling the multi-period adaptive fleet planning problem in an efficient way to implement an ADP-based algorithm. To begin with, a set of assumptions are defined so as to draw the boundaries of the model. Then, the multi-period adaptive fleet planning problem is formulated as a state-space system.

4.2.1. Problem assumptions

To formulate the multi-period adaptive fleet planning problem, it is highly convenient to employ an already existing model as a baseline to assess the effectiveness of a new ADP-based algorithm. Indeed, it should be recalled that the main contribution of this MSc thesis revolves around developing a solution methodology rather than a modelling method. Therefore, the same assumptions from the model of Repko and Santos (2017) are taken. Even if their model is simplified in certain aspects, the assumptions made can still provide a meaningful strategic perspective without a high relevant loss of accuracy. The set of assumptions is listed hereafter:

- **Inelastic demand:** The model takes the demand already captured by the airline and considers it as fixed and independent of the offered route frequency. While overall route demand is function of socio-economical variables, in reality the demand captured by each airline depends on the level of frequency offered in that route amongst other factors such as pricing and schedules. Nevertheless, the model makes this assumption to avoid non-linear expressions, thereby reducing significantly computational power. In any case, results should be examined carefully since the model will tend to reduce frequency and increase load factors, especially in low-demand scenarios. Nevertheless, the establishment of a minimum frequency could avoid a significant reduction in route market shares.
- **Star network and leg-based demand:** The airline is assumed to operate a hub network, in which passengers are transported between nodes and the hub. Consequently, direct flights between nodes are not considered and demand is leg-based. On the one hand, the assumption of a star network avoids the need of including continuity constraints for aircraft in the network, since both inbound and outbound legs of a route will be operated by the same aircraft. If any triangular route was to be considered, then it would need to be modelled as a single flight. On the other hand, the assumption of a leg-based demand allows to only model the routes rather than all possible itineraries. While this assumption reduces the problem dimensions, it also prevents the model from considering network effects regarding transported and spilled passengers.
- **Average week demand:** The demand of a representative week is taken as a reference for all the weekly demands occurring throughout a year. Therefore, seasonal behaviours and weekly fluctuations are not considered in the model. Furthermore, the optimisation is carried out on a weekly basis and annual profits result from aggregating the weekly profits of 52 weeks.
- **Frequency schedule:** Following the pattern of other strategic planning models, frequency schedule is preferred over a time-based schedule. The reason for this choice is that frequency planning only needs assigning aircraft types to routes rather than a specific aircraft tail as the case for time-based schedules. Due to this fact, frequency schedule is more efficient from a strategic perspective.
- **No revenue management and single cabin:** By overlooking revenue management, a single average fare is considered for each route. Indeed, it is very hard to meaningfully and efficiently model the impact of revenue management with a long term perspective. Since an average fare is already assumed, the adoption of a single cabin configuration does not entail a high loss of accuracy and is considered necessary to reduce problem dimensions.
- **No cargo:** Even though revenues from transporting cargo can be very relevant for certain airlines, the model relaxes this assumption. Therefore, the impact of cargo is not considered within the airline operating performance.
- **Demand uncertainty:** As previously commented, the airline industry must deal with different sources of uncertainty: demand, competition and fuel price are the variables that are most uncertain. While the impact of competitors and fuel price changes shall never be dismissed from a strategic perspective, demand is often considered the only stochastic variable when modelling dynamic problems. In fact, this restriction in modelling is due to the computational limitations entailed by solving dynamic programs with backward induction. Since the objective of this MSc thesis is to prove Approximate Dynamic Programming as an efficient way to solve the fleet planning problem, it is considered convenient to limit the study to just one stochastic process representing demand uncertainty. In this way, a potential success of this proof-of-concept may pave the way for including other sources of uncertainty.

4.2.2. Problem formulation

Taking into account the set of assumptions recently discussed, the nomenclature as well as the formulation of the multi-period adaptive fleet planning problem follows up the work of Repko and Santos (2017). Even though all the essence of their model is fully respected, the problem formulation has been slightly adjusted to define the six essential elements of the state-space framework.

Sets

Within the problem, three sets are defined as follows:

$\mathcal{T}$:= set of discrete time periods considered within a time horizon T with $t = 0$ denoting the initial conditions, $\{0, 1, ..., T\}$;

$\mathcal{I}$:= set of aircraft types, being I the total number of aircraft types, $\{1, 2, ..., I\}$;

$\mathcal{R}$:= set of routes, with R as total number of routes considered, $\{1, 2, ..., R\}$

State vector

With the aim of defining the state vector of the fleet planning problem, the following elements are denoted.

R_{ti} := number of aircraft of type i available at time stage t;

D_{tr} := average weekly demand for each of both legs of route r at time stage t.

Let $[R_{ti}]_{i\in\mathcal{I}}$ and $[D_{tr}]_{r\in\mathcal{R}}$ be the resource vector R_t and demand vector D_t respectively.

R_t := $[R_{ti}]_{i\in\mathcal{I}}$

D_t := $[D_{tr}]_{r\in\mathcal{R}}$

Then, the system state vector can be defined as the concatenation of both resource and demand vectors, which will provide the necessary information to describe the system evolution throughout a time period t:

$$S_t = (R_t \quad D_t) \tag{4.15}$$

Decision vector

As for the decision vector, the actions to control the system are assumed to be:

x_{ti}^{buy} := number of aircraft of type i acquired at time stage t and to be delivered in next time period $t+1$;

x_{ti}^{disp} := number of aircraft of type i assigned at time stage t to be disposed in next time period $t+1$;

y_{tir} := assigned weekly frequency of aircraft type i on route r in period t ;

q_{tr} := number of passengers transported weekly on route r per leg in time period t.

Following the same procedure as with the state vector, the decision vector is denoted as:

$$a_t = \left(x_t^{buy} \quad x_t^{disp} \quad y_t \quad q_t\right) \tag{4.16}$$

where:

x_t^{buy} := $\left[x_{ti}^{buy}\right]_{i\in\mathcal{I}}$

x_t^{disp} := $\left[x_{ti}^{disp}\right]_{i\in\mathcal{I}}$

y_t := $\left[y_{tir}\right]_{i\in\mathcal{I}, r\in\mathcal{R}}$

q_t := $\left[q_{tr}\right]_{r\in\mathcal{R}}$

Exogenous information

Passenger demand evolution can be simulated as a stochastic process of subsequent random events, thereby being revealed gradually at the start of each time stage. Nevertheless, it is very convenient to perform a transformation of random variables: instead of directly considering passenger demand a stochastic variable, we take demand growth as an equivalent stochastic variable. This transformation of random variables allows to reduce the dimensions of the model without losing the problem essence.

Let Ω be a set of all possible demand outcomes with probability function $\mathbb{P}$ and let ω be a real random variable that maps the set of outcomes with a measurable real number indicating the percentage of global demand growth. This is expressed mathematically like:

$$\omega_t : \quad \Omega \rightarrow \mathbb{R} \tag{4.17}$$

where $\mathbb{R}$ represents the set of real numbers. Furthermore, the random variable can either be a scalar or a vector. Next, demand uncertainty is modelled by means of the following stochastic process:

$$\omega_{[t]} := (\omega_0, \omega_1, ..., \omega_{\mathbb{T}-1}) \tag{4.18}$$

which represents the history of demand growth. In this way, each possible combination of $\omega_{[t]}$ defines each possible scenario. If Ω is a finite space of possible events, then the model will feature a finite number of scenarios and the probability function $\mathbb{P}$ will be discrete. Likewise, if Ω is continuos, then the number of possible scenarios is infinite and probability function $\mathbb{P}$ will be continuos.

Finally, the absolute growth of passenger demand $\widehat{D}_t$ can be obtained by applying the transformation below:

$$\widehat{D}_{t+1} = \omega_{t+1} D_t \tag{4.19}$$

Transition function

As commented previously, the transition function dictates the dynamics of the system. It indicates the manner in which the system evolves from the state S_t to the state S_{t+1} in function of the previous decision made a_t and the most recent realisation of exogenous information ω_{t+1}.

$$\begin{gathered} S_{t+1} = S^{Model}(S_t, a_t, \omega_{t+1}) \\ S_{t+1} = \begin{bmatrix} R_{t+1} \\ D_{t+1} \end{bmatrix} = \begin{bmatrix} R_t + x_t^{buy} - x_t^{disp} \\ D_t + \widehat{D}_{t+1} \end{bmatrix} = \begin{bmatrix} R_t + x_t^{buy} - x_t^{disp} \\ (1+\omega_{t+1}) D_t \end{bmatrix} \end{gathered} \tag{4.20}$$

On the one hand, the transition function imposes the condition of continuity in the fleet throughout time. In other words, the impact of all fleet-related decisions taken at a current time stage t will be visible in next time period. Indeed, this component of the transition function is deterministic. On the other hand, the value taken by demand vector D_{t+1} is uncertain and will only be known once the random variable $\widehat{D}_{t+1}$ is revealed.

Contribution function

In the context of strategic fleet planning, the contribution function can be correlated to the annual operating profit of an airline. As a reminder to the reader, the airline operating profit is defined as the amount of money earned from its core business operations:

$$\text{Operating Profit} \quad = \quad \text{Operating Revenue} - \text{Operating Expenses} - \text{Disposal costs} - \text{D\&A} \tag{4.21}$$

where D&A stands for Depreciation and Amortization. By analysing Equation 4.21, it is clear that operating profit can be employed as a key performance indicator (KPI) to measure the resulting rewards from the decisions made during each time stage.

With the aim of writing the contribution function as the annual operating profits of an airline, the following problem parameters are written:

f_r := average fare paid by a passenger for one leg on route r;

c_{tir}^{var} := cost of operating route r with aircraft type i;

c_{ti}^{own} := average weekly cost of owning an aircraft of type i in time period t;

c_{ti}^{disp} := penalty cost of disposing an aircraft of type i in time period t;

n_t := number of operating weeks considered in time period t.

Then, the contribution function can be formulated following the definition of the airline operating profits specifically introduced by Repko and Santos (2017):

$$C_t(S_t,a_t) \quad = \quad n_t \sum_{i\in\mathscr{I}} \sum_{r\in\mathscr{R}} \left(2 f_r q_{tr} - (c^{var}_{tir} y_{tir} + c^{own}_{ti} R_{ti} + c^{disp}_{ti} x^{disp}_{ti})\right) \tag{4.22}$$

As observed, the annual operating profit is expressed as the aggregation of all n_t average weekly operating profits. Therefore, all terms within the summation signs are expressed in a weekly basis. The first term in brackets represents the weekly revenue obtained from transporting passengers on route r. It must be noticed that this term is multiplied by two so as to account for both legs of route r. The second term corresponds to the weekly operating costs, which are calculated on a route and frequency basis. It is assumed that operating costs encompass airport and en-route taxes, fuel, crew and maintenance costs for both legs of a route . The third term indicates ownership costs, which can be understood as either lease costs or D&A costs per period due to aircraft purchase. This type of cost is applied to each aircraft in the fleet, but it is independent from any route operation performed. Finally, the fourth term is associated to the total penalty cost of disposing aircraft expressed on a weekly basis. On the one hand, this penalty cost may result from returning leased aircraft before the end of a leasing contract. On the other, if aircraft were purchased this cost may be incurred due to an existent difference between the aircraft selling value and its accounted value.

Objective function

It has already been mentioned that in the case of airline fleet planning, the objective is to maximize current airline profits as well as the expected profits within the next uncertain years. By taking advantage of its recursion nature, the objective function is finally expressed in the form of Bellman equation:

$$\begin{aligned} V_t(S_t) &= \max_{a_t\in\mathscr{A}_t} \left(C_t(S_t,a_t) + \gamma_{t+1}\mathbb{E}\{V_{t+1}(S_{t+1})|S_t\}\right) \\ &= \max_{a_t\in\mathscr{A}_t} \left(C_t(S_t,a_t) + \gamma_{t+1}\sum_{s'\in\mathscr{S}} \mathbb{P}\left(s'|S_t,a_t\right) V_{t+1}(s')\right) \end{aligned} \tag{4.23}$$

where γ_{t+1} is the discount factor translating the profit of period $t+1$ into its present day value and $\mathscr{A}_t$ is the set of feasible decisions a_t at time t. Furthermore, it should be noted that the solution of a dynamic problem is a random variable $\mathbb{A}_t(S_t(\omega_t))$, since its value depends on a random process $\omega_{[t]}$. Therefore, we are not looking for a single optimal decision but an optimal policy $A^\pi(S_t)$: the best rule for making decisions a_t in function of the given state S_t at that time, which results from the previous multivariate stochastic process $\omega_{[t]}$ (Powell , 2007):

$$a_t = A^\pi(S_t) \tag{4.24}$$

If demand uncertainty was not considered, then the objective function would be simplified for the deterministic case by eliminating the expectation of the value function:

$$V_t(S_t) \quad = \quad \max_{a_t\in\mathscr{A}_t} \left(C_t(S_t,a_t) + \gamma_{t+1} V_{t+1}(S_{t+1})\right) \tag{4.25}$$

Constraints

In the same way as the contribution function, the model constraints are adopted from the work of Repko and Santos (2017). Nevertheless, instead of defining a general set of constraints for all time stages, the present model will differentiate a set of constraints for each time period. In other words, the problem constraints compose the set $\mathscr{A}_t$ of feasible decisions a_t at time t. This is done to respect the state-space modelling framework. Before introducing its mathematical expression, the following problem parameters are defined below:

OT_r := average time required to fly a leg of route r ;

TAT_i := turnaround time of aircraft type i;

BT_i := block time or maximum number of operating hours that an aircraft of type i is allowed per week;

cap_i := seat capacity of aircraft type i;

LF_r := maximum load factor on route r;

Y^{Min}_{tr} := minimum frequency imposed on route r at time t;

$R^{initial}_i$:= number of aircraft of type i initially owned by the airline.

Given R_t and S_t, the set of feasible decisions at time t is expressed as follows:

$$\mathscr{A}_t(R_t, D_t) = \left\{ a_t : \begin{array}{rcll} 2\sum_{r\in\mathscr{R}}(OT_r + TAT_i)y_{tir} & \leq & BT_i R_{ti} & \forall i \in \mathscr{I} \\ q_{tr} & \leq & D_{tr} & \forall r \in \mathscr{R} \\ \sum_{i\in\mathscr{A}} cap_i LF_r y_{tir} & \geq & q_{tr} & \forall r \in \mathscr{R} \\ \sum_{i\in\mathscr{I}} y_{tir} & \geq & Y_{tr}^{Min} & \forall r \in \mathscr{R} \\ R_{0i} & = & R_i^{initial} & \forall i \in \mathscr{I} \end{array} \right\} \quad \begin{array}{r} (4.26) \\ (4.27) \\ (4.28) \\ (4.29) \\ (4.30) \end{array}$$

with:

$$R_{ti} \in \mathbb{Z}^+, x_t^{buy} \in \mathbb{Z}^+, x_t^{disp} \in \mathbb{Z}^+, y_{tir} \in \mathbb{Z}^+, q_{tr} \in\in \mathbb{R}^+ \tag{4.31}$$

The first set of Constraints 4.73 prevents the airline from assigning to an aircraft type i more operating hours than the maximum allowed per week, given the number of aircraft of type i available at time t. In doing so, route frequency assignments to aircraft types are consistent with the number of operating hours available based on current fleet size and composition. Likewise, the set of Constraints 4.74 ensures that the number of passengers transported on each flight leg of route r does not exceed existent demand at time t. The third set of Constraints 4.75 indicates that the total number of passengers transported per route is limited to the total available seats assigned to each specific route. Due to political or strategic reasons for instance, it might be compulsory to operate certain routes over a minimum frequency independently of their profitability. These constraints are reflected by the equation set 4.76. Finally, the last set of constraints make sure that the fleet composition and size at the starting period $t = 0$ matches with the airline current fleet status.

To complete the formal formulation of the mathematical model, we need to interrelate the different time periods by adding consistency between the decisions made a_t and their impact on next state variable S_{t+1}. This is achieved by defining a set of feasible combinations (a_t, R_{t+1}) ruled by the dynamics of the problem:

$$\mathscr{Y}_t(R_t, R_{t+1}) = \left\{ (a_t, R_{t+1}) : \quad R_{ti} + x_{ti}^{buy} - x_{ti}^{disp} = R_{t+1i} \quad \forall i \in \mathscr{I} \right\} \tag{4.32}$$

4.2.3. Model summary

Having defined all problem elements, the problem can be summarised as follows. Given a state variable S_t and decision vector a_t at time stage t

$$S_t = (R_t \quad D_t) \tag{4.15}$$

$$a_t = \left(x_t^{buy} \quad x_t^{disp} \quad y_t \quad q_t \right) \tag{4.16}$$

and considering demand evolution as a multivariate stochastic process

$$\omega_{[t]} := (\omega_0, ..., \omega_{\mathbb{T}-1}) \tag{4.18}$$

with

$$D_{t+1} = D_t + \widehat{D}_{t+1} = (1 + \omega_{t+1}) D_t \tag{4.33}$$

,

the objective of the airline fleet planning problem is to find the best policy $A^\pi(S_t)$

$$a_t = A^\pi(S_t) \tag{4.24}$$

that maximises the expected airline operating profits aggregated throughout the years:

$$V_t(S_t) = \max_{a_t \in \mathscr{A}_t} \left(C_t(S_t, a_t) + \gamma_{t+1} \mathbb{E}\{V_{t+1}(S_{t+1}) | S_t\} \right) \tag{4.23}$$

In Equation 4.23 $C_t(S_t,a_t)$ corresponds to the airline operating profit earnt at time stage t:

$$C_t(S_t,a_t) = n_t \sum_{i\in\mathscr{I}} \sum_{r\in\mathscr{R}} \Big(2 f_r q_{tr} - (c_{tir}^{var} y_{tir} + c_{ti}^{own} R_{ti} + c_{ti}^{disp} x_{ti}^{disp})\Big). \tag{4.22}$$

Likewise, $\mathscr{A}_t(R_t,D_t)$ denotes the set of feasible decisions at time stage t, while $\mathscr{Y}_t(R_t,D_t)$ defines the set of feasible combinations (a_t,R_{t+1}) dictated by the dynamics of the problem and that interrelate subsequent time periods:

$$\mathscr{A}_t(R_t,D_t) = \Big\{ a_t : \quad 2\sum_{r\in\mathscr{R}} (OT_r + TAT_i) y_{tir} \le BT_i R_{ti} \quad \forall i \in \mathscr{I} \tag{4.73}$$

$$q_{tr} \le D_{tr} \quad \forall r \in \mathscr{R} \tag{4.74}$$

$$\sum_{i\in\mathscr{A}} cap_i LF_r y_{tir} \ge q_{tr} \quad \forall r \in \mathscr{R} \tag{4.75}$$

$$\sum_{i\in\mathscr{I}} y_{tir} \ge Y_{tr}^{Min} \quad \forall r \in \mathscr{R} \tag{4.76}$$

$$R_{0i} = R_i^{initial} \quad \forall i \in \mathscr{I} \quad \Big\} \tag{4.77}$$

$$\mathscr{Y}_t(R_t,R_{t+1}) = \Big\{ (a_t,R_{t+1}) : \quad R_{ti} + x_{ti}^{buy} - x_{ti}^{disp} = R_{t+1i} \quad \forall i \in \mathscr{I} \Big\} \tag{4.80}$$

with:

$$R_{ti} \in \mathbb{Z}^+, x_t^{buy} \in \mathbb{Z}^+, x_t^{disp} \in \mathbb{Z}^+, y_{tir} \in \mathbb{Z}^+, q_{tr} \in\in \mathbb{R}^+ \tag{4.81}$$

In conclusion, the multi-period adaptive fleet planning problem is formulated as a finite-horizon discrete-time dynamic program, which meets the properties of a Markov chain and can also be represented with a scenario tree scheme.

4.3. Implementation of scenario tree model

Previous sections showed that the multi-period fleet planning formulation leads to a hierarchical data structure characteristic of scenario trees. When it comes to implementing an ADP algorithm, it is in fact very convenient to take full advantage of this type of structures so as to track efficiently all the information generated at every stage of the scenario tree (S_t, a_t, ω_t). The reason for this lies in the fact that an ADP algorithm must perform a vast number of computational operations to learn iteratively better policies. The more efficiently the algorithm handles data, the faster computational operations will be performed, thus reducing overall computational times. Therefore, it is highly important to track, access and modify very easily all ADP framework elements describing the full problem. Otherwise, running a sufficient amount of iterations might become unmanageable in terms of computational time.

Given this important requirement, the scenario tree model of the problem will be built upon object-oriented programming (OOP). The choice of this programming paradigm is justified by the fact that it allows to simulate each state-space framework element in a very straightforward and structured way. Indeed, OOP consists in the definition of *objects* which possess different characteristics, also known as *attributes*. As for the multi-period fleet planning problem, these objects can be related to the different situations that an airline can face within the time horizon considered. In this way, each situation possesses a series of attributes, which correspond to the state-space framework elements describing its context (e.g. current demand, fleet status, and airline actions amongst others).

For this purpose, the present MSc thesis uses a Python framework especially developed by Huerta-Cepas et al. (2016) to work with tree data structures: *Environment for Tree Exploration* (ETE). With a clear academic background in bioinformatics, ETE was initially developed as a tool for analysing phylogenetic trees. Nevertheless, this Python toolkit boasts a wide range of features which are very efficient for handling other type of

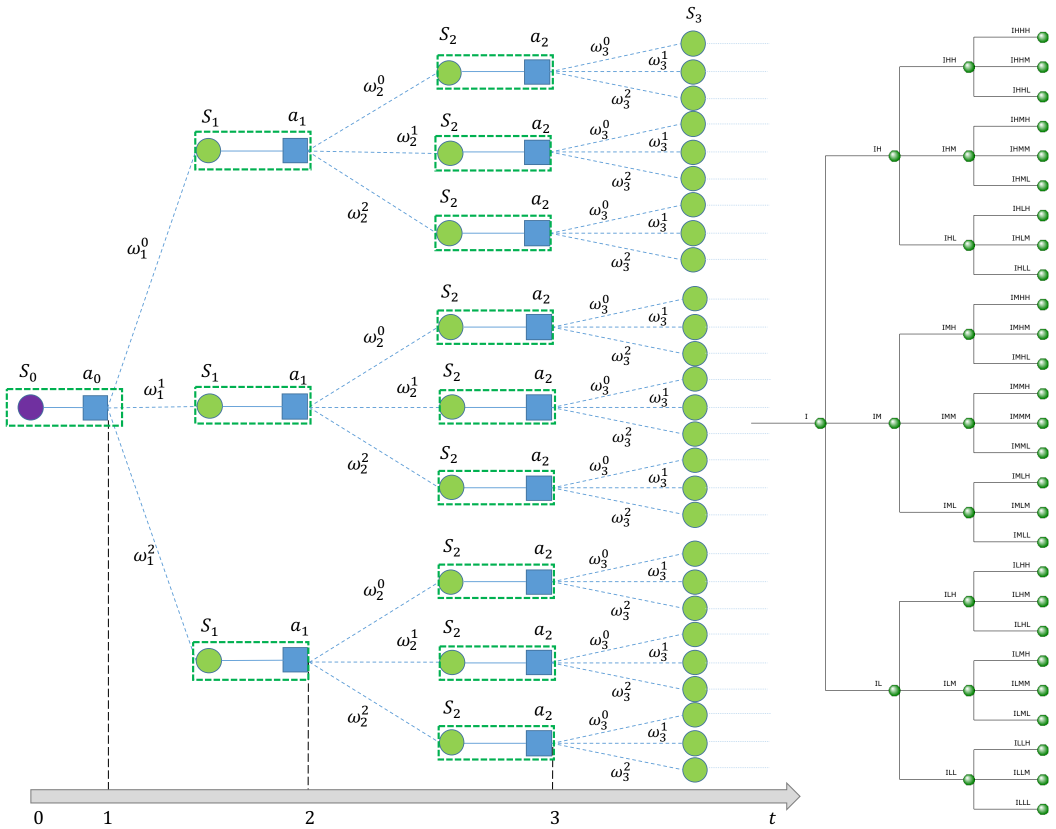


Figure 4.5: Scenario tree modelled with clustered nodes (left) and equivalent ETE clustered tree (right).

complex data tree structures. Amongst all these possibilities, ETE facilitates automated building, manipulation, analysis and visualization of hierarchical trees.

Using the notation of ETE, the scenario tree of the multi-period fleet planning problem can be described with the following terms:

- **Node:** Being the essential component of the scenario tree, a node stores all structured data related to a situation in period t^* resulting from a chain of outcomes $\omega_{[}t]$ that occurred in $t < t^*$. Therefore, every node is assigned a series of attributes that characterise the entire situation in period t^*. For the multi-period fleet planning problem, the most important attributes are the state vector, decision vector, value function approximation, total demand growth and total probability. Likewise, the full set of nodes defines the ETE clustered tree, which is equivalent to the modelled scenario tree for multi-period fleet planning.

- **Branch:** A branch links a pair of nodes and takes place whenever the random variable ω_t takes a new value.

- **Parents and children:** Nodes can be followed by a range of possible outcomes, each of which is represented by a children node also referred as descendant. Likewise, a node results from a past situation, which is modelled by the ancestor or parent node.

- **Root:** A root node is the starting point of the scenario tree, it defines the problem initial conditions (e.g. initial fleet and route demands) and thus, has no parents.

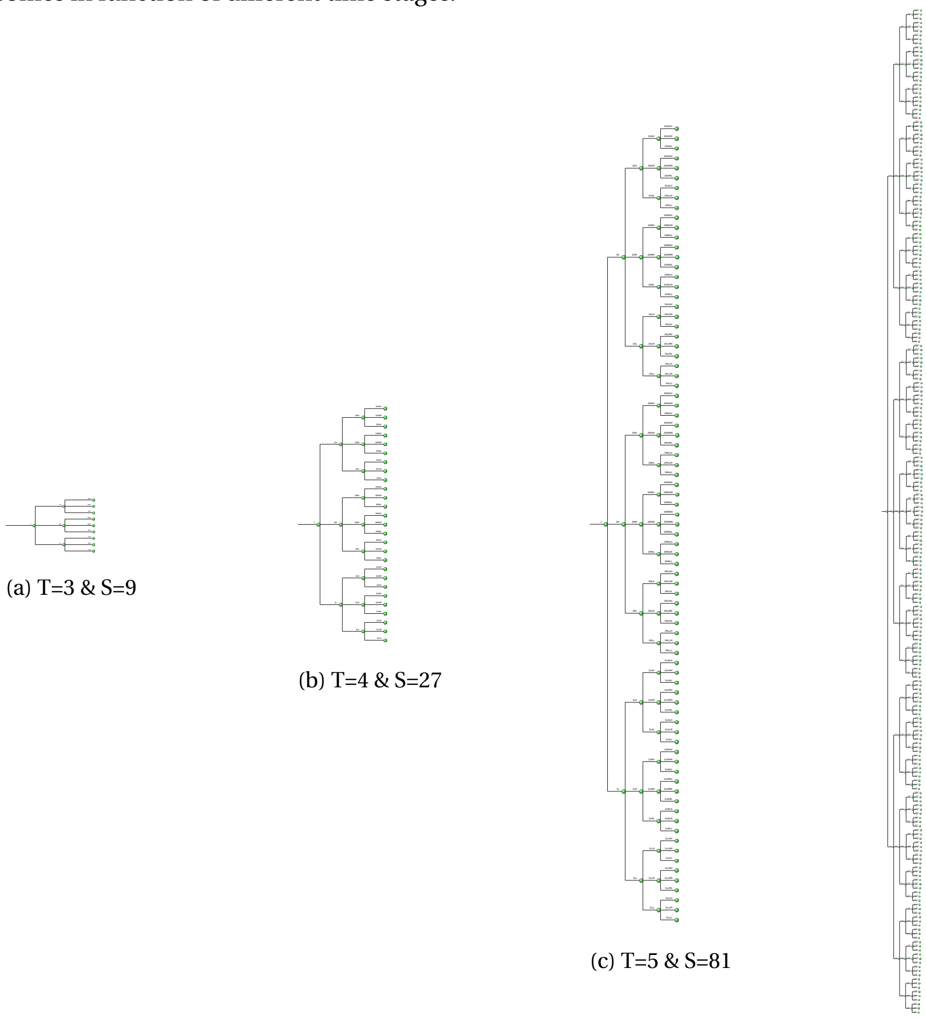
- **Leaf:** A leaf is any node that has no children and represents the planning horizon within the multi-period fleet planning problem.

- **Subtree:** A subtree is the set of nodes and branches that is below a specific node.

Taking into account all these concepts, a specific scenario will be defined unequivocally as the path linking the root node and a specific leaf. Therefore, the number S of possible scenarios represented in tree will increase at exponential rates in function of the number of stages T and possible outcomes m:

$$S = m^{T-1} \tag{4.34}$$

As will be discussed, the exponential relationship between number of scenarios and outcomes will play an essential role in defining the explorations versus exploitation problem as well as the number of iterations required by the algorithm. Figure 4.6 shows the particular growth behaviour for a scenario tree with three finite outcomes in function of different time stages.



(a) T=3 & S=9

(b) T=4 & S=27

(c) T=5 & S=81

(d) T=6 & S=243

Figure 4.6: Exponential growth of scenario tree in function of number of stages

4.4. Implementation of ADP algorithm

By analysing the fundaments of ADP, it is concluded that the number of possible algorithmic strategies to implement is endless, highly dependent on the structure of the problem tackled. Nevertheless, a common pattern can be identified amongst all ADP algorithm versions; they always feature six common building blocks:

- **Step 0:** Value function approximation
- **Step 1:** Generation of Monte Carlo sample
- **Step 2:** Problem optimisation
- **Step 3:** Simulation of next period impact
- **Step 4:** Value function update
- **Step 5:** Stop criteria

The same steps are also followed to deploy an ADP algorithm suitable for the multi-period adaptive fleet planning problem, which will be based on approximate value iteration. To distinguish it from further algorithm improvements that will be performed later in Chapter 6, this initial version is denoted as *baseline ADP algorithm*. The developed algorithm is presented hereunder, whose main building blocks will be described in the upcoming subsections.

Algorithm 1 Baseline ADP algorithm: Approximate value iteration for time-discrete finite horizon problem

procedure : STEP 0. INITIALIZATION OF VALUE FUNCTION APPROXIMATION
  Step 0a. Initialize an approximation for the value function $\overline{V}_t^0(S_t^0, a_t^0)$ for all states S_t and decisions $a_t \in \mathscr{A}_t(R_t, D_t)$ given $t = \{0, 1, ..., T-1\}$.
  Step 0b. Set $n = 1$.
  Step 0c. Initialize S_0^1.

procedure FORWARD PASS
  STEP 1. GENERATION OF MONTE CARLO SAMPLE
  Observe the random variable ω_t and set multivariate sample path $\omega_{[t]}^n$.

  STEP 2. PROBLEM DECOMPOSITION AND OPTIMISATION
  for $t = \{0, 1, ..., T-1\}$ **do**
    Determine the action using ϵ-greedy. With probability ϵ, choose randomly an action $a_t^n \in A_t$ based on a SA-inspired policy. With probability $1-\epsilon$, choose a_t^n using

$$a_t^n = \arg\max_{a_t \in \mathscr{A}_t^n} \quad \overline{V}_t^{n-1}(S_t^n, a_t).$$

    STEP 3. SIMULATION OF NEXT PERIOD IMPACT
    Step 3a. Choose a random sample of outcomes $\widehat{\Omega}^n \subset \Omega$,then sample $W_{t+1}^n = W_{t+1}(\omega^n)$ and simulate the next state $S_{t+1} = S^{Model}(S_t, a_t, \omega_{t+1})$.
    Step 3b. Compute

$$\begin{aligned}
\hat{v}_t^n &= \operatorname*{Max}_{a_t \in \mathscr{A}_t^n} \left(C_t(S_t^n, a_t^n) + \gamma_{t+1} \mathbb{E}\left\{ \overline{V}_{t+1}^{n-1}(S_{t+1}^{n-1}) \mid S_t^{n-1} \right\} \right) \\
&\approx \operatorname*{Max}_{a_t \in \mathscr{A}_t^n} \left(C_t(S_t^n, a_t^n) + \gamma_{t+1} \sum_{\omega \in \widehat{\Omega}^n} p_t^n(\widehat{\omega}_{t+1}^n) \overline{V}_{t+1}^{n-1}\left(S^{Model}\left(S_t^n, a_t, \omega_{t+1} \right) \right) \right)
\end{aligned}$$

    STEP 4. UPDATE VALUE FUNCTION
    Update the value function approximation $\overline{V}_t^n$ using

$$\overline{V}_t^n(S_t^n) = (1-\alpha_{n-1})\overline{V}_t^{n-1}(S_t^n) + \alpha_{n-1}\hat{v}_t^n.$$

  STEP 5. START NEW ITERATION AND EVALUATE STOP CRITERIA
  Increment n
  if $n \leq N_{stop}$ **then**
    go to procedure: STEP 1.

procedure : STEP 6. RESULTS POSTPROCESSING
  Return the value functions $(\overline{V}_t^n)_{t=0}^{T-1}$ and best policy found $A^{\pi}(S_t)$.

4.4.1. Step 0. Approximation strategy for value function

To tackle complex dynamic programs, ADP avoids *the curses of dimensionality* at expenses of finding an effective approximation strategy for the value function. As stated by Powell (2007), the major difficulty in ADP lies in the capability of creating a policy by approximating the unknown future value of being in a state. In fact, the value function approximation should be considered as the cornerstone of any ADP algorithm, since it has a decisive impact on its success.

Mathematical analysis of the value function

When it comes to approximating the value function, it is essential to have a clear picture of the mathematical structure of the problem as well as the manner in which all its framework elements interrelate. For this reason, it is well worth analysing in detail the components of the value function:

$$\begin{aligned} V_t(S_t) &= \max_{a_t \in \mathscr{A}_t} \left(C_t(S_t, a_t) + \gamma_{t+1} \mathbb{E}\{V_{t+1}(S_{t+1})|S_t\}\right) \\ &\approx \max_{a_t \in \mathscr{A}_t} \left(C_t(S_t, a_t) + \gamma_{t+1} \mathbb{E}\left\{\overline{V}_{t+1}(S_{t+1})|S_t\right\}\right) \end{aligned} \tag{4.23}$$

As already highlighted in the modelling section, the value function in period t (V_t) consists of the contribution function at the same time t (C_t) as well as the expected value function in next period ($\mathbb{E}\{V_{t+1}\}$). From this definition, the recursive nature of the problem is clearly visible. Even though it may appear evident, this concept should always be taken into account while studying the problem and defining the approximation strategy. More particularly, the reader should remember that the contribution function C_t measures how well a decision a_t performs in state S_t, whilst the expected value function represents the aggregate impact of our current decisions a_t on future contributions from $t+1$ onwards. Then, it follows that the objective function of any dynamic problem corresponds to the value function at the initial time stage (V_0). As for the multi-period fleet planning problem, the airline objective is to maximise the operating profits of the current period C_t as well as the future operating profits expected throughout the years $\mathbb{E}\{V_{t+1}\}$. In other words, the goal is to find a fleet plan policy $A^{\pi}(S_t)$ that recommends which are the best actions to maximise profits given future uncertain demand growth and current conditions.

By looking at the mathematical expression of the airline operating profits,

$$C_t(S_t, a_t) = n_t \sum_{i \in \mathscr{I}} \sum_{r \in \mathscr{R}} \left(2 f_r q_{tr} - (c_{tir}^{var} y_{tir} + c_{ti}^{own} R_{ti} + c_{ti}^{disp} x_{ti}^{disp}) \right) \tag{4.22}$$

it can be inferred that the contribution function only depends explicitly on certain components of the state vector and decision vector. These components are the number of passengers transported q_{tr}, the route frequency assigned to each aircraft type y_{tir}, the number of aircraft to be disposed per type in next period x_{ti}^{disp} as well as the total number of aircraft owned per type R_{ti}:

$$C_t(S_t, a_t) := C_t\left(R_t, x_t^{disp}, y_t, q_t\right) \tag{4.35}$$

This does not imply however, that the demand vector D_t and decision variables x_t^{buy} do not have a significant influence upon the problem. Whereas the impact of D_t is clearly seen through the set of constraints $\mathscr{A}_t(R_t, D_t)$ and thus transmitted by the number of passengers carried q_{tr}, the variables x_t^{buy} play an essential role in defining the set $\mathscr{Y}_t(R_t, D_t)$ of feasible combinations (a_t, R_{t+1}) between periods.

Amongst all mentioned variables, the resource vector R_{t+1} stands as an evident mathematical link between the contribution function C_t and the expected value function $\mathbb{E}\{V_{t+1}\}$. Given the transition function of the problem, it is clear that the resource vector R_{t+1} depends directly on the fleet-related decisions x_t^{buy} and x_t^{disp} taken during previous time period.

$$\begin{gathered} S_{t+1} = S^{Model}(S_t, a_t, \omega_{t+1}) \\ S_{t+1} = \begin{bmatrix} R_{t+1} \\ D_{t+1} \end{bmatrix} = \begin{bmatrix} R_t + x_t^{buy} - x_t^{disp} \\ D_t + \widehat{D}_{t+1} \end{bmatrix} = \begin{bmatrix} R_t + x_t^{buy} - x_t^{disp} \\ (1+\omega_{t+1}) D_t \end{bmatrix} \end{gathered} \tag{4.20}$$

Likewise, current network-related decisions q_t and y_t are made based on both resource vector R_t and demand vector D_t of time period t. Therefore, it is possible to discern an overlapping structure as shown below.

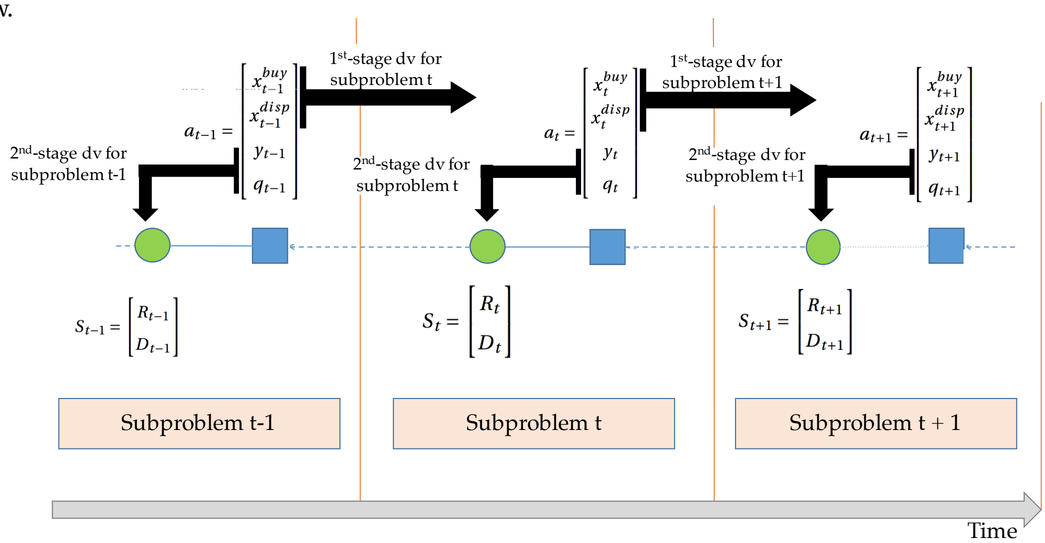


Figure 4.7: Interrelation of decision vector with both current and subsequent subproblems

The conclusion that can be drawn from Figure 4.7 is that the decisions taken at time stage t will have an impact on present and future operations, which will be measured by the contribution function $C_t(S_t, a_t)$ and the expected value of aggregated profits throughout next time periods $\gamma_{t+1}\mathbb{E}\{V_{t+1}(S_{t+1})|S_t\}$. In light of this, the decision vector a_t can be decomposed in two groups of decisions variables:

$$a_t = (\underbrace{x_t^{buy} \quad x_t^{disp}}_{\substack{1^{st}\text{-stage dv for}\\ \text{next subproblem t+1}}} \qquad \underbrace{y_t \quad q_t}_{\substack{2^{nd}\text{-stage dv for}\\ \text{current subproblem t}}} \quad)$$

- **Fleet decisions** $\left(x_{ti}^{buy}, x_{ti}^{disp}\right)$ act as first-stage decision variables of next subproblem $t+1$, since future airline operations in period $t+1$ will be constrained by fleet modifications agreed during period t. Therefore, fleet decisions mainly focus on maximising future expected operating profit by means of optimising the fleet composition and size for the next time period.
- **Network decisions** (q_{tr}, y_{tir}) act as second-stage decision variables of current subproblem t, since they will be treated as recourse measures to optimise the current performance of an airline, whose fleet was modified during previous subproblem (first-stage decisions variables). That is to say, network decisions are intended to maximise current operating profits by optimising the current fleet operations, which are constrained by the previous fleet-related decisions.

Taking into account the variable dependencies of the contribution function shown in previous Equation 4.35

$$C_t(S_t, a_t) := C_t\left(R_t, x_t^{disp}, y_t, q_t\right), \tag{4.35}$$

it can be appreciated that $C_t(S_t, a_t)$ can be divided into two parts, thereby following the same two-stage structure presented by the decision variables of the problem:

$$C_t(S_t, a_t) \quad = \quad \widetilde{C}_t(S_t, y_{tir}, q_{tr}) - \sum_{i \in \mathcal{I}} c_{ti}^{disp} x_{ti}^{disp}. \tag{4.36}$$

In this reformulation, $\widetilde{C}_t(S_t, y_{tir}, q_{tr})$ denotes the annual operating profits in period t without considering the disposal costs:

$$\widetilde{C}_t(S_t, y_{tir}, q_{tr}) \quad = \quad n_t \sum_{i \in \mathcal{I}} \sum_{r \in \mathcal{R}} \left(2 f_r q_{tr} - (c_{ir}^{var} y_{tar} + c_{ti}^{own} R_{ta})\right) \tag{4.37}$$

Since $\widetilde{C}_t(S_t, y_{tir}, q_{tr})$ does not depend on any decisions $\left(x_{ti}^{buy}, x_{ti}^{disp}\right)$ impacting next period, this expression could be understood as both the contribution and objective function of an equivalent 1-stage fleet planning problem (1-FPP). Taking advantage of this new concept, the value function of the problem can be rewritten as:

$$V_t = \underset{a_t \in \mathscr{A}_t}{\text{Max}} \left(\widetilde{C}_t(S_t, y_{tir}, q_{tr}) + \gamma_{t+1}\mathbb{E}\{V_{t+1}(S_{t+1}) \mid S_t\} - \sum_{i \in \mathscr{I}} c_{ti}^{disp} x_{ti}^{disp} \right) \tag{4.38}$$

This reformulation has a major role to play in the definition of the value function approximation. Indeed, considering this two-stage decomposition and assuming that an airline will always try to optimise their current performance, the following deduction can be made:

Provided that the demand vector value for next period D_{t+1} was known, the selection of current fleet decisions $\left(x_{ti}^{buy}, x_{ti}^{disp}\right)$ would imply the knowledge of next resource vector R_{t+1} and consequently the entire state vector S_t. This fact would lead automatically to the unique identification of the optimal recourse actions $\left(q_{t+1r}, y_{t+1ir}\right)$, which maximise the ailine operating profits $\widetilde{C}_{t+1}(S_{t+1}, y_{t+1ir}, q_{t+1r})$ independently of its disposal plans.

In other words, the potential identification of the future state vector $S_{t+1} = (R_{t+1} \quad D_{t+1})$ would allow to solve an equivalent 1-stage fleet planning problem (1-stage FPP) whose decision vector a_{t+1} would be reduced to $\left(q_{t+1r}, y_{t+1ir}\right)$. Since any airline will try to optimise its current performance, this fact leads to the hypothesis that optimal recourse actions will always be chosen. This reasoning, which may appear self-evident for some readers, provides crucial information for drawing an effective value function approximation: it allows to reduce the number of explicit variables to track the maximum values of the contribution function without considering disposal costs, $\widetilde{C}_t$:

$$\widetilde{C}_t^*(S_t, y_{tir}, q_{tr}) \rightarrow \widetilde{C}_t^*(S_t) \tag{4.39}$$

Indeed, Equation 4.39 shows that $\widetilde{C}_t^*$ can be expressed as function of the state vector solely. Furthermore, solving the equivalent 1-stage FPP allows to estimate the impact of previous fleet decisions $\left(x_{t-1i}^{buy}, x_{t-1i}^{disp}\right)$ on current airline profits. By *forcing* several fleet sizes and compositions R_{ti} to this problem, one could simulate a wide range of situations in which the operations of an airline are constrained by former fleet decisions, whose impact might have been positive or negative for the financial performance of the airline. What is more, this reasoning allows to determine the mathematical relationship between the 1-stage FPP contribution function $\widetilde{C}_t$ and the resource vector R_t. Taking advantage of this concept, a reduced state space (R_t, D_t) can be browsed so that the profit impact of meaningful fleet sizes and compositions is observed for a particular demand vector D_t. In other words, one could *force* some values of the resource vector R_t and solve its corresponding 1-stage FPP, so as to shed light on the mathematical behaviour of $\widetilde{C}_t^*$. For a 1-stage FPP featuring three aircraft types and a specific demand per route, $\widetilde{C}_t^*$ takes the form shown in Figure 4.8.

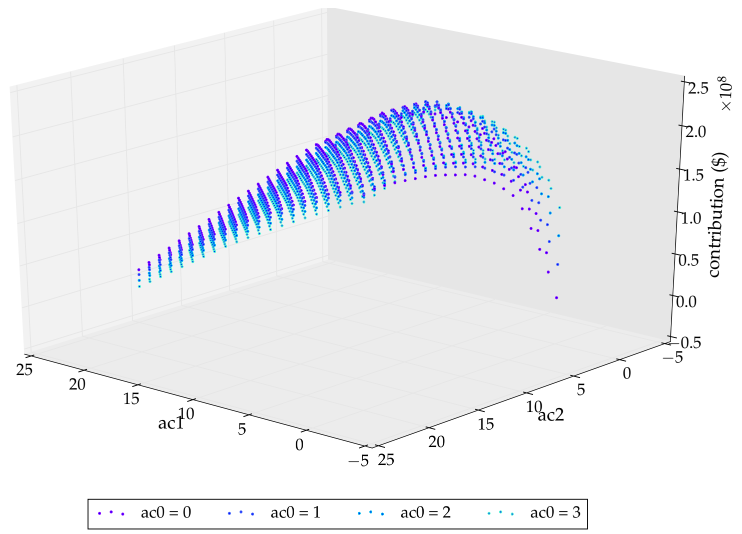


Figure 4.8: Baseline situation: impact of different fleet composition and sizes on $\widetilde{C}_t^*$ for three aircraft types: ac_0, ac_1 and ac_2

More specifically, Figure 4.8 shows the maximum operating profits that an airline could make given a certain fleet under fixed demand conditions without regard to disposals: aircraft of type 1 and 2 are represented with axis x and y respectively, while the colour legend tracks the amount of aircraft of type 0. Therefore, every coordinate (x,y) and colour corresponds to a certain contribution value.

Although the demand vector D_t does not appear explicitly in the contribution function, its influence acting from the set of constraints $\mathscr{A}_t(R_t, D_t)$ is evident. Figure 4.9 shows how the contribution function behaves upon demand growth:

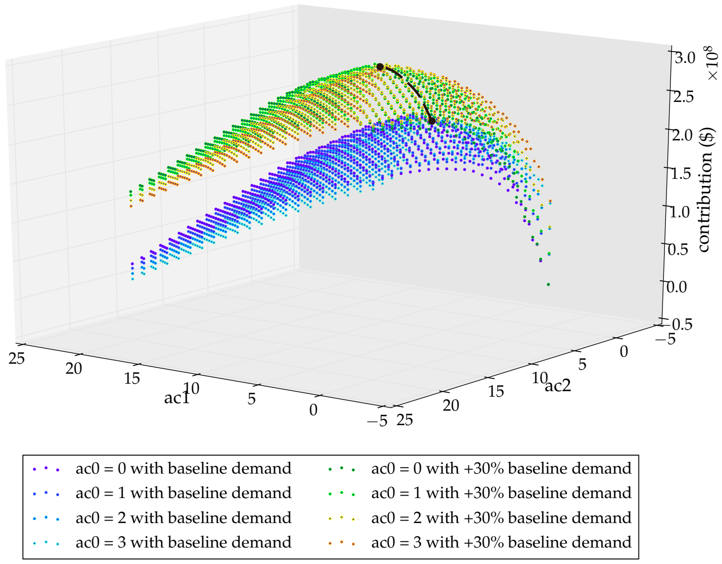


Figure 4.9: Impact of demand growth on contribution function $\widetilde{C}_t^*$ for three aircraft types: ac_0, ac_1 and ac_2

Evidently, the operating profits increase for higher levels of demand captured. Nevertheless, the locations of the greatest values in profits do not remain steady; instead, they shift to major fleet sizes and different compositions with respect to the baseline situation. This trend is highlighted by the discontinuous black segment in Figure 4.9, whose extremes connect the locations of the maximum values of both cases studied. Indeed, this change in optimal fleet can be expected since a larger and different fleet composition might be required to take the greatest advantage of an increase in demand captured.

By analysing the general shape of the contribution function of the 1-stage fleet planning problem in function of the state vector $\widetilde{C}^*$, two main characteristics can be drawn and extrapolated to the original contribution function:

- The relationship between the airline operating profits C_t and the number of aircraft owned per type R_{ti} is a **non-linear concave function**.
- The marginal contribution of having R_{ti} aircraft of type i to the overall profit function C_t is dependent on the entire fleet combination and size R_t as well as the existent demand D_t. That is to say, the value of aircraft of type i_1 is clearly influenced by the number of aircraft of other types $i \neq i_1$ owned already by the airline. Therefore, it exists a **coupling relationship amongst the marginal value of all aircraft types**. For instance, the marginal value of acquiring a certain amount of aircraft of type i would definitely have a positive impact on overall profits if the airline fleet size was zero. Whereas, this same acquisition could have a negative impact, provided that the airline owned already other aircraft types acting as substitute resources with respect to aircraft type i.

All in all, these two conclusions are crucial for the selection of an effective value function approximation strategy, a fact which will be discussed hereafter.

Selection of the value function approximation strategy

As far as approximation strategies are concerned, approximate dynamic programming has drawn upon a vast range of contributions from statistics and machine learning communities. Powell (2007) sums up very well the most popular methods applied within the ADP community, which can be classified into: lookup tables, parametric models and nonparametric models. With the aim of selecting an effective value function approximation for the multi-period fleet planning problem, the main characteristics of each of these three family methods are analysed:

- **Lookup tables:** Ideally suited for discrete state spaces, lookup tables consist in associating to every possible state S_t an estimation of the value of being on that state $\overline{V}_t(S_t)$.
  - **Advantages:** Lookup tables provide an intuitive and flexible way to represent value functions without the need to analyse its mathematical form in detail. Due to their flat representation between states, lookup tables work very well for discrete state spaces, in which they can eventually achieve value function approximations with a high degree of precision.
  - **Disadvantages:** The use of lookup tables implies the need to store an estimated value $\overline{V}_t(S_t)$ for every possible state S_t in the problem. When it comes to a discrete state space of small dimensions, this may not represent a problem; whereas for problems with higher dimensions, this may entail excessive memory requirements. It is for this reason that state aggregation methods are combined with lookup tables to reduce unmanageable dimensions. Furthermore, it should be taken into account that the update of the estimated value of being in a state S_t only occurs when that specific state is visited. Likewise, the value function update in that specific state will not help improve the estimation of other states. Consequently, this requires deciding on whether choosing the most promising actions based on current estimates or other less promising actions just for the sake of learning a better estimation of the value function. This trade-off is commonly known in ADP literature as the exploration versus exploitation problem.
- **Parametric models:** Preferred for estimating values throughout a continuous state space, parametric models revolve around finding analytic functions $\overline{V}_t(S_t|\theta)$ parametrized by a vector θ featuring much smaller dimensions than the number of possible states.
  - **Advantages:** The use of a regression model to approximate the value function implies the estimation of a parametric vector θ, whose dimensions are theoretically much smaller than the state space. Therefore, the ADP algorithm has no longer to update the value functions estimations for each of the states; instead, just these parameters need to be updated. This highly powerful feature allows to estimate simultaneously the value of being in states not yet visited.
  - **Disadvantages:** From a vast set of possible mathematical forms $\mathscr{F}$, any parametric model requires the initial selection of a series of basis function $\phi_f(S_t)$ and their corresponding parametric coefficient θ_f, both of which provide essential information regarding V_t:

$$\overline{V}_t(S_t|\theta) = \sum_{f\in\mathscr{F}} \theta_f \phi_f(S_t) \tag{4.40}$$

  As can be inferred from Equation 4.40, parametric models differ from lookup tables in a way that the former introduce much more rigidity when it comes to approximate a function. Indeed, the most challenging part of designing a successful regression function is to determine what is the best mathematical form of these basis functions ϕ_f and coefficients θ_f, which will provide a meaningful information regarding the behaviour of the value function. While there is no doubt of the approximation power entailed by a well-fitting parametric model, the truth is that seeking for the right mathematical form is very time-consuming and may end up being an entire problem itself. What is more, a poor regression model may introduce so many constraints that may eventually ruin the search of a good fleet policy. Nevertheless, there have been numerous papers in resource allocation and inventory management (e.g. Godfrey and Powell,2001, 2002a, 2002b; Topaloglu and Powell, 2006; Papageorgiou et al., 2014) which have proved the efficiency and flexibility of separable piecewise-linear approximations to estimate nonlinear value functions. These generally take the following mathematical form:

$$\overline{V}_t(R_t) = \sum_{i\in\mathscr{I}} \overline{V}_{ti}(R_{ti}) \tag{4.41}$$

  where $\overline{V}_{ti}(R_{ti})$ is a scalar piecewise-linear function. For this specific function, it must be highlighted that the contributions of each of the resource types are decoupled from each other.

- **Nonparametric models:** Based on nonparametric statistics, nonparametric models perform local approximations to the value function by learning from a series of observations, rather than using analytical functions with a preset behaviour. In fact, nonparametric models lie in between parametric models and lookup tables, being closer to the latter.
  - **Advantages:** These type of models avoid the challenging need of predefining a parametric mathematical form, thereby providing a high level of flexibility when approximating the value function. In contrast to lookup tables, nonparametric models facilitate the updating of the value function estimations in states not visited yet.
  - **Disadvantages:** Local approximations can either be built upon averages (k-nearest neighbour), weighted sums (kernel regression) or learnt basis functions (neural networks) from prior observations tuned with parameters. This means that the quality of the approximations will usually depend on parametric tuning, the quantity of observations of the value function available, the quality of these own observations as well as the existent dispersion between observations. As the case of lookup tables, this fact may imply the need of solving the exploration versus exploitation problem commented before. Evidently, this will depend basically on the specific nonparametric method chosen. Furthermore, the major inconvenience is that nonparametric models aggregate points in a multidimensional space. In fact, the more density of observations there is, the greater likelihood to find an effective approximation there will be. Unfortunately, problems featuring higher dimensions will usually face more difficulties due to dispersion: the density of points in a large multidimensional space will tend to be low, thereby becoming difficult the aggregation of observations to generate an effective approximation.

Once discussed the pros and cons of the different approximation methods and having already considered the mathematical structure of our problem, it is now possible to choose with more awareness an approximation strategy.

By just looking at the problem formulation, an estimation of the dimensions of the problem can be made. If the objective is to deal with a realistic case in the air transport industry, the state vector of the model can surpass very easily the 50 dimensions. This estimation results directly from the size of both resource and demand vectors, which are purely proportional to the number of aircraft types and routes considered respectively. Indeed, it is quite common to see airlines either boasting networks of over 50 destinations or fleets composed by more than 5 aircraft types or configurations. However, the state-space dimensions could be reduced significantly if routes with common characteristics were transformed into segmented groups $g \in \mathcal{G} = \{0, 1, ..., G^{Region} - 1\}$.

$$S_t^g = F^{Reduction}(S_t) \tag{4.42}$$

For instance, the most intuitive strategy would be to group routes according to geographical regions or classify them according to common market trends. In this way, a group of routes could be simply tracked by their averaged total common growth Δ_{tg} with respect to their initial conditions D_{0r}:

$$S_t = (R_t \quad D_t) \quad \rightarrow \quad S_t^g = (R_t \quad \Delta_t) \tag{4.43}$$

where $\Delta_t := \quad [\Delta_{tg}]_{g \in \mathcal{G}}$ and evidently $G^{Region} << R$. This reduction of variables would not imply a loss of information if the grouped routes were undergoing similar rates of demand growth. Likewise the random variable ω_t, which indicates demand growth between periods would be treated as a vector $\omega_t := [\omega_{tg}]_{g \in \mathcal{G}}$. Without losing information, it can be derived from the previous formulated model that:

$$\widehat{D}_{t+1r} = \omega_{t+1r} D_{tr} \quad \rightarrow \quad \widehat{\Delta}_{t+1g} = \omega_{t+1g} \Delta_{tg} \tag{4.44}$$

and thus, the new transition function would be:

$$\begin{aligned} S_{t+1}^g &= S^{Model,g}(S_t^g, a_t, \omega_{t+1}) \\ S_{t+1}^g &= \begin{bmatrix} R_{t+1} \\ \Delta_{t+1} \end{bmatrix} = \begin{bmatrix} R_t + x_t^{buy} - x_t^{disp} \\ \Delta_t + \widehat{\Delta}_{t+1} \end{bmatrix} = \begin{bmatrix} R_t + x_t^{buy} - x_t^{disp} \\ (1 + \omega_{t+1}) \Delta_t \end{bmatrix} \end{aligned} \tag{4.45}$$

Apart from this, the number n of possible values taken by the growth vector ω_t can either be finite or infinite depending on whether we assume a discrete or continuous probability function. In any case, the number of possible scenarios will scale with the number of time periods t at the exponential rate of m^{t-1}. For instance, the combination of three outcomes and six periods will result in 243 scenarios to analyse. All in all, the estimated problem dimensions together with the need of estimating the value of every each of the states lead to the conclusion that the use of lookup tables may entail an excessive need for storage memory and computation.

Furthermore, we already know that the behaviour of the contribution function C_t with respect to the state vector S_t is non-linear and concave. Then, we can deduce from this statement that the value function will also be non-linear and concave, since the value function is composed of the contribution function and the expected value function of next stage. As commented previously, many nonlinear and concave value functions have been approximated by means of using separable piecewise-linear approximations. Its clear success and flexibility proven by many papers dealing with resource allocation and inventory management problems, may encourage the use of this type of regression functions. However, it must be reminded the second conclusion drawn from analysing the behaviour of the contribution function with respect to S_t, as shown in Figure 4.9: it exists a coupling relationship amongst the marginal value of all aircraft types. Because of this fact, the piecewise-linear function for a given aircraft model $\overline{V}_{ti}(R_{ti})$ should be indexed according to the fixed composition of the other aircraft types, which would act as parameters. In any case, this strategy may be feasible when there is just another resource type. Nevertheless, Powell (2007) warns about the steep price to pay when additional resource types also influence the behaviour of the contribution function with respect to R_{ti}. This fact can easily be extrapolated to the multi-period fleet planning problem. By just looking at Figure 4.9, one can imagine the vast number of piecewise-linear functions required to be indexed in function of the different fleet compositions and demand rates.

The above mentioned constraints lead us to pay special attention to nonparametric models, which represent a research area of high potential, actively investigated by the ADP community. Even if less academic references proving its ADP application can be found compared to the case of parametric regressions, nonparametric methods result very promising for our type of problem tackled. In fact, they are expected to provide more flexibility than traditional regression models while still reducing the need of storage memory and computational power. Furthermore, they allow the updating of values in states not yet visited. However, it should be taken into account that nonparametric models need to be initialised with a series of observations, which still need to be computed. Apart from this, the dispersion effects on the ADP algorithm should be carefully monitored.

The exploration of pros and cons of the different approximation methods to our problem prompts us to choose a nonparametric model to approximate the value function of the multi-period fleet planning problem. Nevertheless, the family of nonparametric models is very comprehensive and includes different techniques based on statistical learning. According to Powell (2007), some of the most popular methods that have received major attention from the ADP community are: k-nearest neighbour, kernel regression, local polynomial regression and neural networks. Based on the summary review carried out by Powell (2007), the main characteristics of each method are discussed:

- **Neural networks** are reviewed as the most powerful technique. Indeed, their level of maturity is such that they have provided many industry applications with a highly flexible framework capable of estimating completely unknown functions. In consequence, neural networks have led to a vast research field, thus presenting an extensive literature. When comparing them with other simpler strategies, Powell (2007) warns about the fact that it is not possible to know in advance which problem classes will benefit the most from the additional generality brought by neural networks. From his review, it can also be inferred that it might be worth to firstly assess the performance of simpler methods which will most likely take less time to implement. In case these fail, then it will make more sense to invest more time in developing neural networks.
- **K-nearest neighbour** is described as the simplest form to implement non-parametric regressions. This technique approximates the value of being in a queried state by performing an average sum of the priorly observed values in the k nearest points to the query (training dataset). This entails tuning the number k of points taken for the estimation. Even though the implementation of k-nearest neighbour

is expected to be fast, a weakness of this technique is that the obtained approximation for the queried state changes abruptly according to the number of k-nearest points considered. Therefore, this may lead to significant instabilities in the ADP algorithm.

- **kernel regressions** have been a recurrent topic in statistical learning literature and frequently explored by the machine learning community; however, they have not been widely tested in ADP algorithms. This approximation technique follow the idea of k-nearest neighbour since they apply a weighted sum of prior observations. Nevertheless, the instabilities faced by k-nearest neighbour are mitigated due to the introduction of a weighting function (kernel), whose value decreases with the distance between the observed and queried states. In this way, the value function approximation changes smoothly between queried states and becomes more stable. As far as implementation time is concerned, the application of kernel regression is not expected to be much more complex than k-nearest neighbour.

- **Local polynomial regressions** are a generalization of kernel regressions, in which linear regression models are calculated at the local surroundings of the queried states. This is achieved by solving a least squares problem that minimizes a weighted sum of least squares: instead of weighting the value function observations, this technique applies a classical linear regression to the existent deviation between the observed and queried values. Powell (2007) highlights a significant improvement in approximation accuracy at the expenses of a visible increase in complexity. Unlike neural networks, it could be argued that the accuracy level achieved with this method might not pay off its time cost.

In light of the main conclusions drawn from the review of Powell (2007), Table 4.1 aims at making an evaluation of the discussed methods according to three criteria relevant for the MSc thesis. Due to the thesis tight schedule, both the estimated implementation time and the availability of previous references have been considered as much important as the approximation accuracy that could potentially be obtained with each of the methods. Therefore, all three dimensions are assigned an equal weight for comparison.

Table 4.1: Comparison between different nonparametric techniques scored from 1 (worst) to 5 (best)

Dimensions	K-nearest neighbour	Kernel regressions	Local polynomial regressions	Neural networks
Approximation accuracy	1	3	4	5
Implementation time	5	5	2	1
Past references in ADP	1	2	2	3
Total score	7	**10**	8	9

Certainly, neural networks appear to be one of the most sophisticated techniques in terms of approximation accuracy: it presents an important track record of successful implementations in several engineering applications. Nevertheless, its effectiveness also depends on the problem type tackled. Given this fact, there is no clear guarantee of how well neural networks could work for the multi-period fleet planning problem. This uncertainty combined with the estimated implementation time and the lack of specific ADP references in the air transport industry, leads us to pay more attention to the other simpler methods. Indeed, the idea behind this preference is to take advantage of lesser implementation times so as to be able to discard rapidly the selected method in case it fails.

Amongst the simpler methods, k-nearest neighbour is discarded straightforwardly given its low accuracy expected. On the other hand, the estimated potential of kernel regressions applied to the multi-period fleet planning problem looks promising in the context of the MSc thesis: it allows to reach an acceptable level of approximation accuracy within a reasonable timeframe. Apart from this, the theoretical fundamentals of kernel regressions are clear and sound, fact which compensates for the scarcity of ADP references using kernel regressions. Finally, local polynomial regressions do not seem worth exploring: if it were the case that a more sophisticated method was required, then it would be preferred to invest more time in developing neural networks than local polynomial regressions. As discussed previously, it could be argued that the accuracy level achieved with local polynomial regressions might not pay off its time cost, whereas this would most probably be the contrary for neural networks.

In light of the above discussion, the value function approximation strategy chosen is kernel regressions. Apart from all the mentioned reasons, it must be added the fact that a successful implementation of kernel regressions may entail a significant contribution to the development of new ADP algorithms based on non-parametric approximations. Therefore, the opportunity to contribute to the ADP community is worth the risk of applying a technique that is less established in the field.

Application of kernel regressions

As previously introduced, kernel regression estimate the value $\overline{V}_t^n(S_t^*)$ of being in an unvisited state S_t^* (query), by applying a weighted sum of M prior observations V_t^m of the value function $V_t(S_t)$:

$$\overline{V}^n(S_t^*) \quad = \quad \frac{\sum_{m\in\mathcal{M}_t}^{M} K_h(S_t^*, S_t^m) V_t^m}{\sum_{m\in\mathcal{M}_t}^{M} K_h(S_t, S_t^m)} \tag{4.46}$$

where $K_h(S_t, S_t^m)$ is a weighting function that declines with the distance between the queried state point and the measured state point S^m and h acts as a bandwidth parameter to be tuned. For this particular problem, the weighting function will take the form of a Gaussian kernel, which is well established in statistics literature and is commonly referred as radial basis function. Denoting the Euclidean norm as $||\cdot||$, the Gaussian kernel is defined as follows:

$$K_h(S_t^*, S_t^m) \quad = \quad \exp\left[-\left(\frac{||S_t^* - S_t^m||}{h}\right)^2\right] \tag{4.47}$$

From Equation 4.47, it can be inferred that one of the main advantages of Gaussian kernel is its ability to provide smooth and continuous approximations $\overline{V}^n(S_t^*)$. Nevertheless, it should be noted that a major variation to the general form of kernel regression has been introduced: for this particular problem, the weighted sum is constrained to all those observed states S_t^m sharing the same resource vector R_t^m as the one of the queried state S_t^*. This constraint is introduced with the set of states $\mathcal{M}_t$:

$$\mathcal{M}_t = \{\forall S_t^m \quad | \quad R_t^m = R_t^*\} \tag{4.48}$$

In fact, the introduction of this constraint simplifies the scaling of the Euclidean norm since it allows us to calculate the norm based on the same unit of measurement. From a mathematical viewpoint, it is considered very impractical to add up the differences in aircraft quantity together with the differences in total demand growth. If this was done, the weight of the Gaussian kernel would very likely be biased due to the wrong mix of dimensions within the same calculation. Taking this into account, the Euclidean norm can be calculated as follows:

$$||S_t^* - S_t^m|| = \sqrt{\sum_{i\in\mathcal{I}} \left(R_{ti}^* - R_{ti}^m\right)^2 + \sum_{g\in\mathcal{G}} \left(\Delta_{tg}^* - \Delta_{tg}^m\right)^2} = \sqrt{\sum_{g\in\mathcal{G}} \left(\Delta_{tg}^* - \Delta_{tg}^m\right)^2} \tag{4.49}$$

However, the calculation of kernel weights can be simplified to reduce the computational time of the entire kernel regression. This is achieved by aggregating the total demand growth experienced by each region Δ_{tg} into the equivalent total demand growth experienced by the entire network $\overline{\Delta}_t$:

$$\overline{\Delta}_t = \frac{\sum_{g\in\mathcal{G}} x_{tg} \overline{\Delta}_{tg}}{\sum_{g\in\mathcal{G}} x_{tg}} \tag{4.50}$$

where x_{tg} is the number of routes agrregated in region g. Indeed, $\overline{\Delta}_t$ results from weighting the total demand growth rates experienced by each of the routes across the network. Without losing much accuracy, an equivalent Euclidean norm is drawn:

$$||S_t^* - S_t^m|| \equiv \sqrt{\left(\overline{\Delta}_t^* - \overline{\Delta}_t^m\right)^2} \tag{4.51}$$

Finally, an initial set of observations of the value function $V_t^0(S_t)$ is needed at iteration $n = 0$ as a training dataset for the kernel regression. In terms of accuracy it would be ideal to gather some observations of the real value of $V_t(S_t)$. However, this would require the use of backwards induction, which is basically what we are trying to avoid with approximate dynamic programming. Therefore, this initial set of observations will also be approximated so as to minimise the need for computational power.

In this context, the objective is to generate a training set of *meaningful* and *structured* observations across the scenario tree so as not to compromise significantly the accuracy of kernel regression:

- **Meaningful observations:** By targeting at *meaningful* observations we mean that it is desirable to preserve the essential information of the value function and transmit it to the kernel regression. For this purpose, it is necessary to capture the nonlinear behaviour of C_t with respect to S_t as well as the coupled relationship existing amongst all resource types R_{ti}. By means of browsing a limited state space and solving the corresponding 1-FPP problem, the following approximation can be obtained:

$$\overline{V}_t^0(S_t) \quad = \quad \lambda\,(T-t)\,\widetilde{C}_t^*\,(S_t) \tag{4.52}$$

In Equation 4.52, the value function is approximated as the deterministic aggregation of operating profits from period t onwards, which are all assumed to be equal to $\widetilde{C}_t^*\,(S_t)$: the operating profits that would be obtained when solving the 1-stage fleet planning problem applied to current period t and state vector S_t. Furthermore, λ denotes a control parameter to manage the impact of the initial approximated observations on the entire ADP algorithm. Its important role will be discussed in Chapter 4 but we can advance that its main objective is to expressly underestimate the value of the initial observations $\overline{V}_t^0(S_t)$. Therefore, depending if the operating profits are positive or negative it can take values as follows:

$$\lambda \in \begin{cases} [0,1]\,, & \text{if} \qquad \widetilde{C}_t^*\,(S_t) \geq 0 \\ [1,T]\,, & \text{if} \qquad \widetilde{C}_t^*\,(S_t) < 0 \end{cases} \tag{4.53}$$

where T corresponds to the number of stages of the scenario tree. It must be noted that the interval $[0,1]$ is intended to underestimate positive values of the value function, while the interval $[1,T]$ does the same for negative values.

- **Structured observations:** While capturing *meaningful* observations is crucial to improve the accuracy of kernel regression, it is key to decide on which nodes of the scenario tree these approximated observations will take place. Indeed, observed states need to be spread in a smart way so as to minimise the negative impact of dispersion. For this purpose, it is decided to initially fill with approximated observations the nodes composing the central and extreme scenarios of the scenario tree, highlighted with orange in Figure 4.10. Likewise, the value function in non-observed scenarios will be estimated dynamically from kernel regressions during the subsequent iterations of the ADP algorithm. That is to say, the kernel regression will only be activated when needed. Furthermore, even though Gaussian kernels are designed to decline with the distance between a query and measured point, it may happen that Gaussian kernels still transmit a certain impact coming from very distant points. This transmission may entail adverse effects on the kernel regression and to avoid this, the ADP algorithm makes use of separable kernel approximations. The main idea is to separate the kernel regression according to, at least, two different intervals of the averaged total demand growth $\overline{\Delta}_t$ seen in nodes: the upper interval (U) will be bounded by the growth values of the central scenario and the most optimistic one $\left[\overline{\Delta}_t^{Med}, \overline{\Delta}_t^{Max}\right]$, while the lower interval (L) will be defined by the values of the most pessimistic scenario and the central one as well $\left[\overline{\Delta}_t^{Min}, \overline{\Delta}_t^{Med}\right]$. In this way, one could define a separable kernel regression (subkernel U and subkernel L) as follows:

$$\overline{V}^n(S_t^*) = \begin{cases} \dfrac{\sum_{m\in\mathcal{M}_t^U}^M K_{h_U}(S_t^*, S_t^m)\,V_t^m}{\sum_{m\in\mathcal{M}_t^U}^M K_{h_U}(S_t, S_t^m)}\,, & \text{if} \qquad \overline{\Delta}_t^{Med} \leq \overline{\Delta}_t^* \leq \overline{\Delta}_t^{Max} \\ \\ \dfrac{\sum_{m\in\mathcal{M}_t^L}^M K_{h_L}(S_t^*, S_t^m)\,V_t^m}{\sum_{m\in\mathcal{M}_t^L}^M K_{h_L}(S_t, S_t^m)}\,, & \text{if} \qquad \overline{\Delta}_t^{Min} \leq \overline{\Delta}_t^* < \overline{\Delta}_t^{Med} \end{cases} \tag{4.54}$$

where the observed values feeding the separable kernel regression are filtered according to the sets $\mathcal{M}_t^U$ and $\mathcal{M}_t^L$ defined below:

$$\mathcal{M}_t^U = \left\{\forall S_t^m \quad | \quad R_t^m = R_t^* \quad \text{and} \quad \overline{\Delta}_t^{Med} \leq \overline{\Delta}_t^m \leq \overline{\Delta}_t^{Max}\right\} \tag{4.55}$$

$$\mathcal{M}_t^L = \left\{\forall S_t^m \quad | \quad R_t^m = R_t^* \quad \text{and} \quad \overline{\Delta}_t^{Min} \leq \overline{\Delta}_t^m < \overline{\Delta}_t^{Med}\right\} \tag{4.56}$$

and the Gaussian kernels K_{h_U} and K_{h_L} have different bandwidth values according to the different intervals: h_U and h_L respectively.

As can be inferred from equations 4.54 to 4.56, the nodes, whose averaged total demand growth is very far apart from the one of the queried state, will be automatically not counted for the estimation of that queried state. In this way, the adverse effect of distant nodes with respect to the query is mitigated significantly.

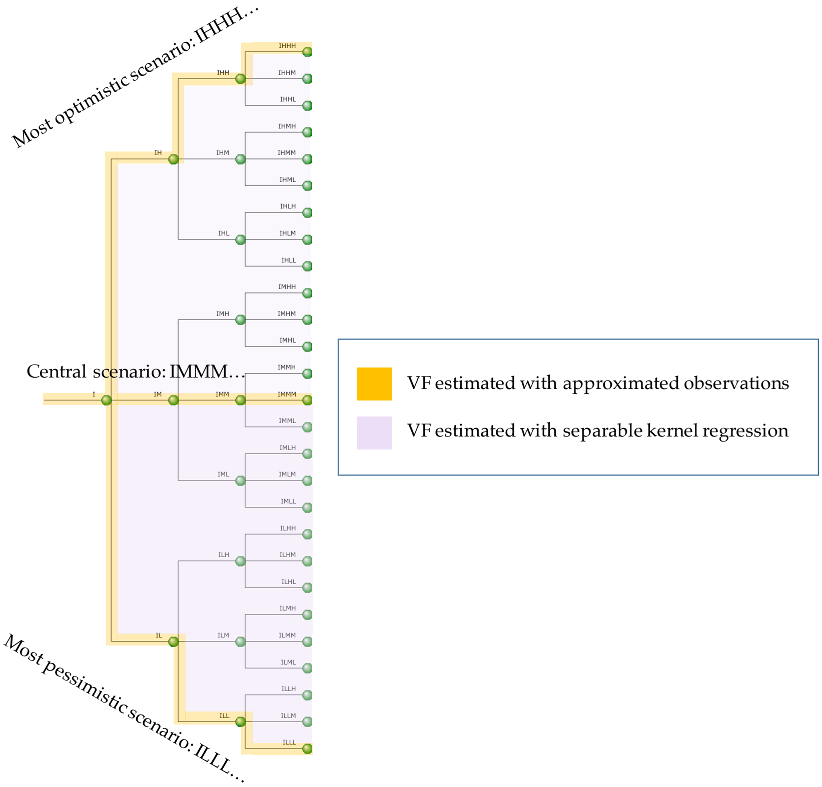


Figure 4.10: Reference scenarios and separable kernel regressions

Calibration of kernel bandwidth

With the aim of achieving an effective weighted sum of observations, the bandwidth parameter h needs to be calibrated within the Gauss kernel expression, where it plays the role of the standard deviation:

$$K_{h_U}(S_t^*, S_t^m) \quad = \quad \exp\left[-\left(\frac{||S_t^* - S_t^m||}{h_U}\right)^2\right] \tag{4.57}$$

$$K_{h_L}(S_t^*, S_t^m) \quad = \quad \exp\left[-\left(\frac{||S_t^* - S_t^m||}{h_L}\right)^2\right] \tag{4.58}$$

Indeed, h is a tunable parameter which determines the range of influence of measured observations. Furthermore, it can be drawn from Figure 4.10 that the value of h should change over time periods since the range of possible scenarios and thus, possible observed $\bar{\Delta}_t^m$ values spreads considerably between time stages. That is to say, the more dispersion there is between observed values, the broader the scope of the gaussian kernel must be. Indeed, if the radius of the gaussian kernel was constant over time periods and independent of the amount of observations obtained throughout the ADP algorithm, the kernel regression would be unstable since it would not be able to eventually capture any observation for higher levels of dispersion. To take into account this dependency, h will be proportional to the standard deviation of the observed values in each of the intervals A and B in period t:

$$h_U(t) = G\sigma_{tU} \tag{4.59}$$

$$h_L(t) = G\sigma_{tL} \tag{4.60}$$

where G is a gain factor, while σ_{tU} and σ_{tL} correspond to the standard deviation of the observed $\bar{\Delta}_t^m$ values found in intervals U and L in period t respectively:

$$\sigma_{tU} = \sqrt{\frac{1}{N_U}\sum_{m\in\mathcal{M}_t^U}^{N_U}\left(\bar{\Delta}_t^m - \mu_{tU}\right)^2} \tag{4.61}$$

$$\sigma_{tL} = \sqrt{\frac{1}{N_L}\sum_{m\in\mathcal{M}_t^L}^{N_L}\left(\bar{\Delta}_t^m - \mu_{tL}\right)^2} \tag{4.62}$$

with N_U and N_L being the number of available observations in the upper and lower intervals at stage t respectively, and μ_{tU} and μ_{tL} denoting the mean of the averaged total demand growth values $\bar{\Delta}_t^m$ observed.

$$\mu_{tU} = \frac{1}{N_U}\sum_{m\in\mathcal{M}_t^U}^{N_U}\bar{\Delta}_t^m \tag{4.63}$$

$$\mu_{tL} = \frac{1}{N_L}\sum_{m\in\mathcal{M}_t^L}^{N_A}\bar{\Delta}_t^m \tag{4.64}$$

Several values of the gain factor G must be tested to assess its impact on the Gaussian kernel and select the best performing one. In order to carry out this calibration, a 6-stage scenario tree with three scalar outcomes ω_t 15% (H), 5% (M) and -5% (L) will be considered. Let IHHHHH, IMMMMM and ILLLLL be the most-optimistic scenario, central scenario and most pessimistic scenario respectively. Therefore, the total averaged demand growth for each scenario in the sixth period would be +101%, +27% and -22% accordingly. All nodes composing this scenarios have an observed value function with the form $\overline{V}_t^0(S_t) = \lambda(T-t)\,C_t^*(S_t)$. The rest of scenarios must be estimated applying the kernel regression.

Figures 4.11 and 4.12 illustrate the behaviour of the Gaussian kernel for the separable kernel regression U. The reader should notice how the weight of the Gaussian kernel goes right down to zero for the interval where demand growth is lower than the one of the central scenario. This is due to the separable kernel regression: the demand growth of the central scenario IMMMMM defines the boundary between the two different kernel regressions. Any queried state whose demand growth is below the one of the central scenario should be estimated with the separable kernel regression L. Furthermore, it is noticeable how an increase in gain G enables to capture better the impact of distant observed nodes and mitigate kernel dispersion. Indeed, the greater the gaussian kernel weight of an observation is, the more influence will have on the kernel weighted sum to estimate the queried state. However, it can be inferred that too high values of G such as 5 or 10 will overestimate the impact of the most distant nodes with respect to the queried state.

Likewise, low values of G such as 0.1 till 1, will reduce too much the kernel bandwidth, thereby constraining the capability of detecting meaningful observations for the kernel regression and enhancing kernel dispersion. For all these reasons, $G = 3$ appears to be a good trade-off that allows to mitigate kernel dispersion without overestimating excessively the impact of distant observations.

Once gain G is calibrated, Figure 4.13 shows how for a queried state with +38% demand growth, the kernel bandwidth increases throughout the stages. That is to say, the kernel bandwidth changes dynamically depending on which stage the state is found. Since higher stages are prone to have more dispersion, the kernel bandwidth will broaden so as to have enough observations.

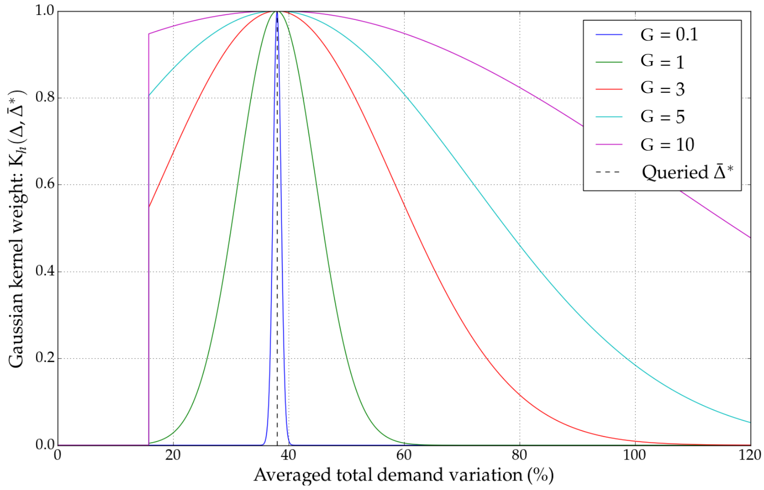


Figure 4.11: Impact of bandwidth gain G on overall kernel weight in period 4 for queried node IHHM

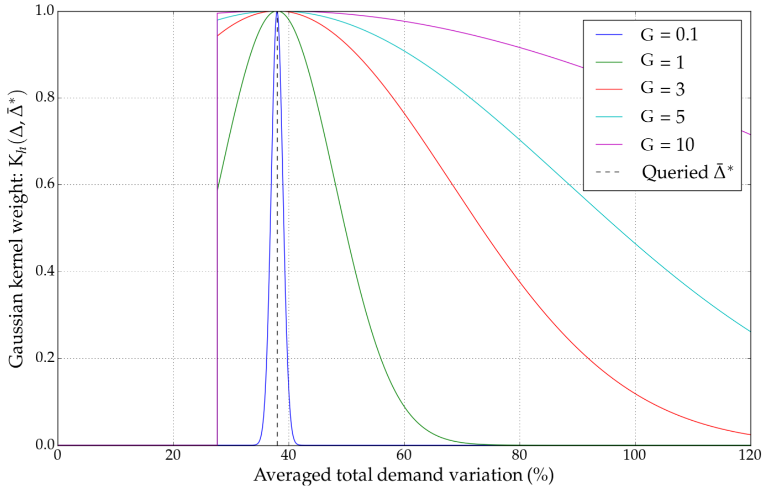


Figure 4.12: Impact of bandwidth gain G on overall kernel weight in period 5 for queried node IHHMMH

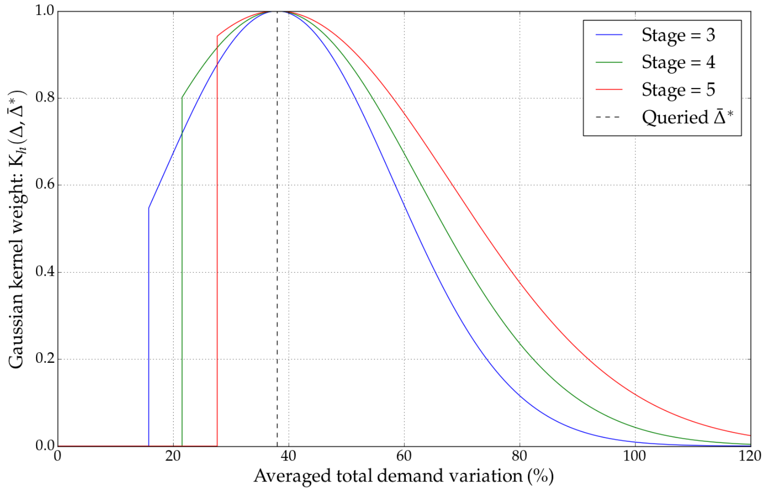


Figure 4.13: Variation of kernel weight according to time periods for $G = 3$

Proposal for generalising the creation of training datasets

While the discussed dynamic control of kernel bandwidth has proven effective between stages 0 to 5, it is evident that it starts to present some limitations for a higher number of stages: points might be so dispersed from the queried state that it is useless to incorporate them into the kernel weighted sum. A solution to this limitation would be to previously estimate more approximated observations in other non-observed scenarios so as to reduce points dispersion. Indeed, this is an area for further research since it would be ideal to determine a correlation between the number and position of required observations and the dimensions of the scenario tree. In any case, follow-up research could revolve around the introduction of a dispersion factor that would help determine an optimal training dataset in terms of number and location of observations.

In the field of probability theory and statistics, Cox and Lewis (1966) defined a normalized measure of data dispersion, also known as the variance-to-mean ratio (VRM) or index of dispersion:

$$D_{tX} = \frac{\sigma^2_{tX}}{\mu_{tX}} \tag{4.65}$$

For the case being studied, μ_{tX} and σ_{tX} would correspond to the mean and standard deviation of the states observed within a given interval X of total demand growth, as defined by previous Equations 4.61 and 4.63. In this way, the index of dispersion would help measure how clustered or dispersed the points would be in each period t and interval X (i.e. for the previous case formulated X refers to either upper or lower intervals). Following this logic, one could determine the level of admissible dispersion D^*_{tX} which, if surpassed, would trigger the partition of the current interval suffering from dispersion. For the sake of simplicity, this interval could be divided in two sub-intervals, whose boundary would be defined by that node with the closest $\overline{\Delta}_t$ value to the mean μ_{tX} of the recently partitioned interval. This node would mark the beginning of a new observed path of consecutive nodes. To guarantee structured observations, this path would follow the next central outcomes M, starting from the given node onwards.

Nonetheless, the computational time that is required to generate the training dataset must not be taken for granted. Evidently, the larger the explored state-space is, the longer its computational time is. Thus, a trade-off must be made between both the minimisation of dispersion and computational time. An alternative to this problem could be the application of kernel regression throughout the observed scenarios: instead of observing a complete scenario from the start to end as done for the extreme and central scenarios, another possibility could be just to observe certain nodes for each of these scenarios and estimate the non-observed ones with other kernel regressions. In this way, there would be two types of separable kernel regressions: a type going along each observed scenario throughout the time periods and another one going in perpendicular to the scenarios in a given time period. Figure 4.14 provides an overview of both proposals to generalise the creation of training datasets.

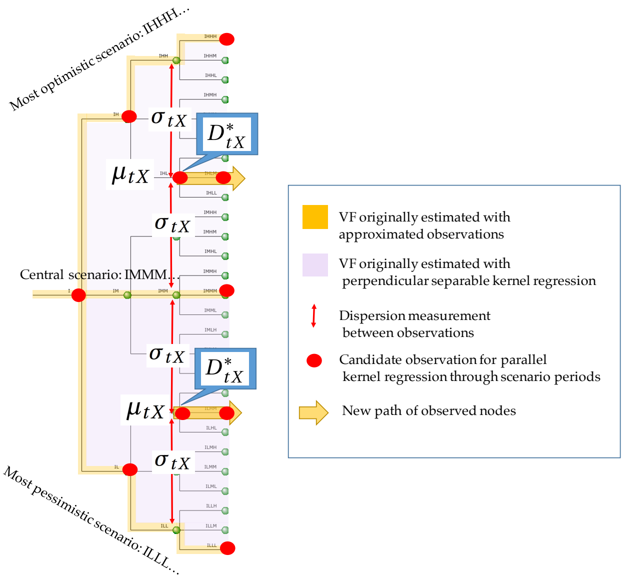


Figure 4.14: Proposal for generalising the creation of training datasets

Verification of kernel regression performance

To verify the correct performance of the designed kernel regression tuned with $G = 3$, it is worth to analyse the correspondence between the estimated value function obtained with kernel regression, and the real approximated value that would be obtained if the value function was initially observed with $\lambda(T-t)\widetilde{C}_t^*(S_t)$. Figures 4.15 to 4.16 show the effective correlation obtained for nodes not belonging to the extremes or central scenarios in different periods. The blue line represents a 1:1 bisection while the green dots refer to the correspondence between the value function estimations obtained for different queried states S_t^*. In this way, it can be appreciated by how much the non-parametric observations differ from the observed value function using $\lambda(T-t)\widetilde{C}_t^*(S_t)$.

While a good correspondence exists throughout all time stages, the impact of dispersed observations is noticeable for the later stage 5: there are kernel value function approximations differing significantly from correlation 1:1 since either they are overestimated or underestimated. Nevertheless, there is a low probability that high correlation errors will affect significantly the ADP algorithm since they happen to be in the least promising states, which will seldom be selected for exploration. What is important to infer from this correlation test is that kernel regression captures the problem structure effectively, thereby allowing the successful identification of the most promising states for the fleet planning problem. Lastly, Figure 4.17 shows the great importance of calibrating well the kernel bandwidth: for instance, an unaware selection of $G = 1000$ may lead to wrong correlations unable to capture the essential information of the problem. An extended discussion related to kernel regression errors can be found in Appendix A.

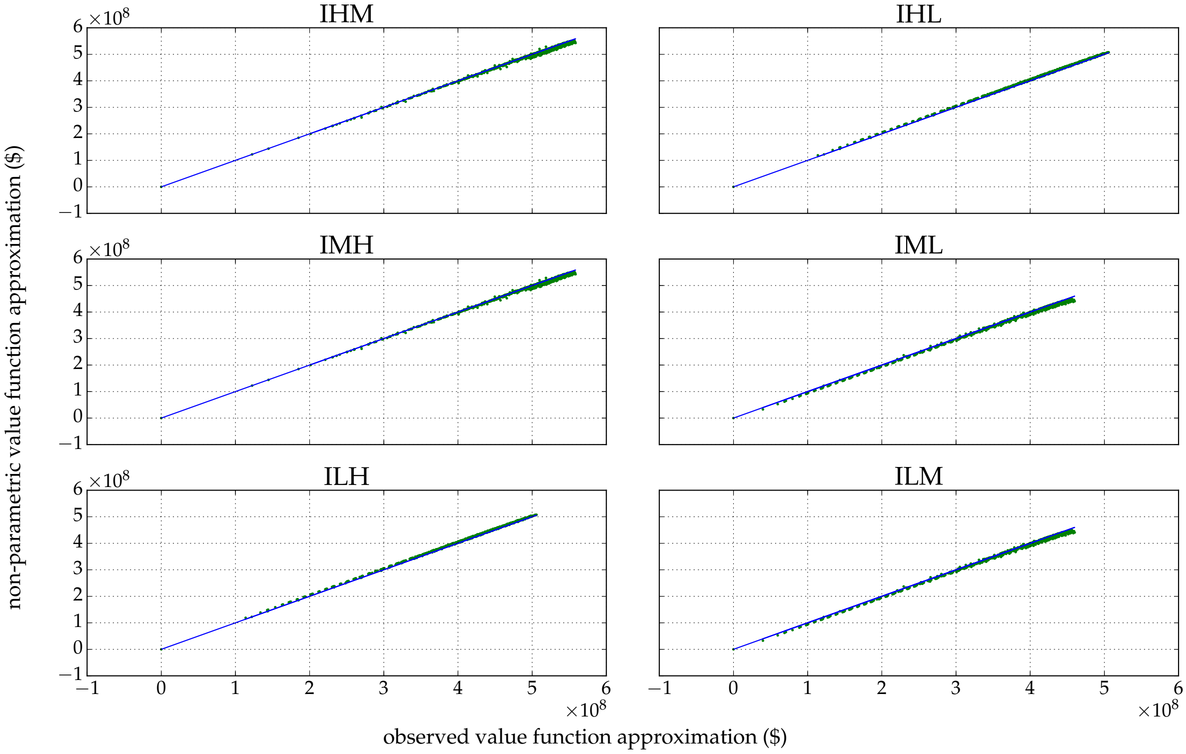


Figure 4.15: Effective value function correspondence at stage 2 with $G = 3$

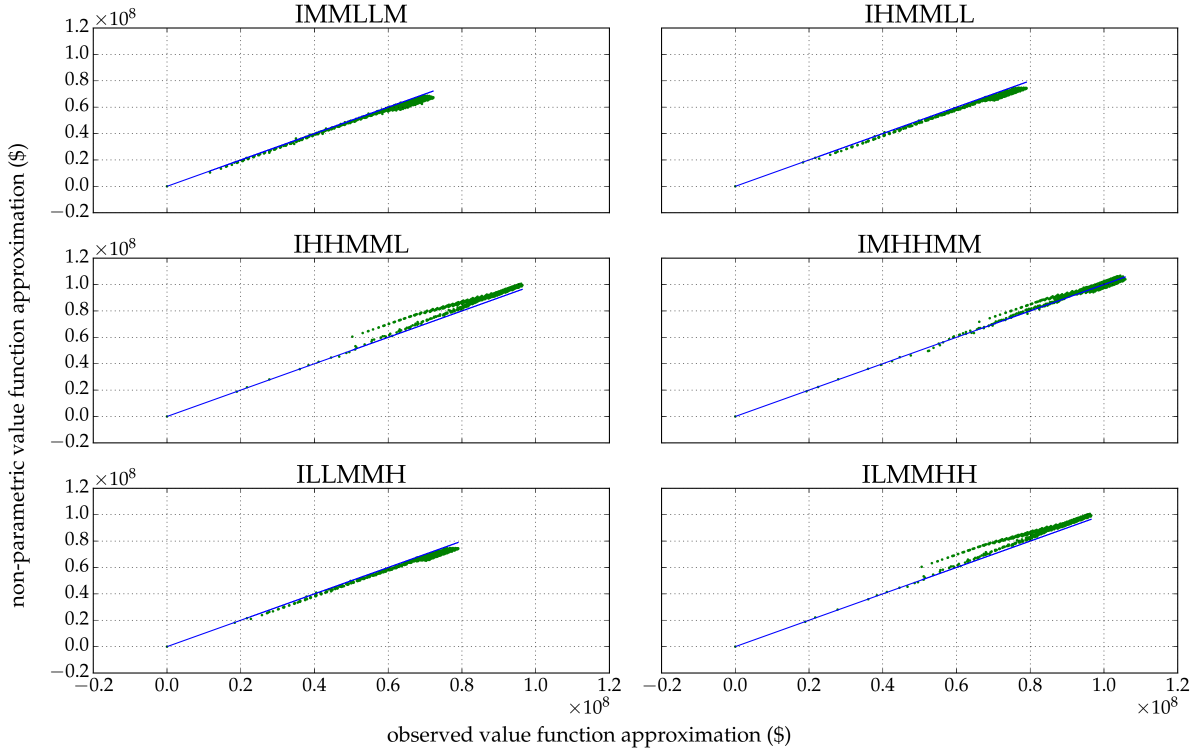


Figure 4.16: Effective value function correspondence at stage 5 with $G = 3$

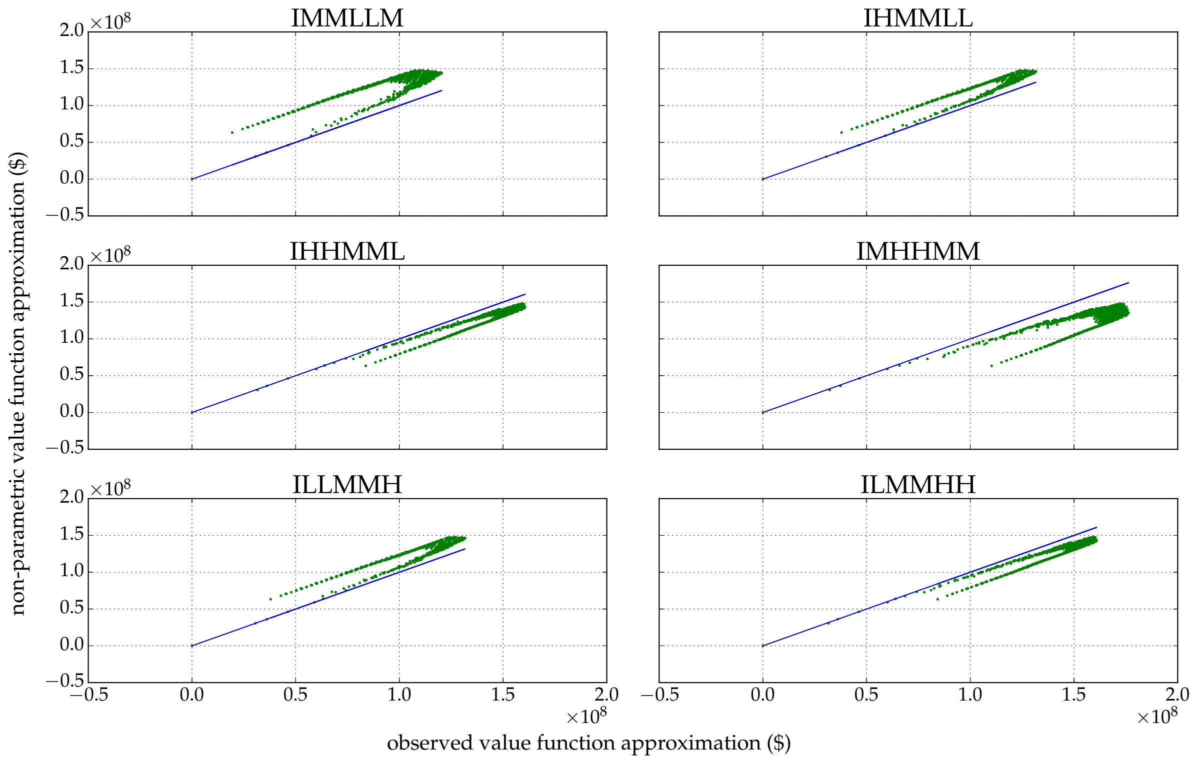


Figure 4.17: Unstable value function correspondence at stage 5 with $G = 1000$

4.4.2. Step 1. Monte Carlo simulations: sampling random demand scenarios

It was stated that approximate dynamic programming can be seen as an optimising simulator since it combines an OR-based optimisation with the flexibility of Monte Carlo simulations and machine learning techniques. In this context, Monte Carlo simulations are employed to provide the ADP algorithm with a sequence of sample realizations of the random variable ω_t. For the multi-period fleet planning problem, this Monte Carlo sampling is performed based on extracting random values from a known distribution of demand growth. Depending on the discrete or continuous nature of the known distribution, we can either have a finite or infinite number of possible random outcomes, thus having a finite or infinite number of possible scenario tree branches. Therefore, the reader should note that all scenario tree schemes presented in the report feature a finite number of possible random outcomes.

A series of N sampled realizations of the random variable ω_t can be extracted computationally assuming a continuous probability function such as the normal distribution:

$$f(Z|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(Z-\mu)^2}{2\sigma^2}} \tag{4.66}$$

$$\widehat{\omega}_t^s = \mu + \sigma Z \tag{4.67}$$

so that the probability of each realization $\widehat{\omega}_t^s$ is defined as:

$$p_t^s(\widehat{\omega}_t^s) = 1/N \tag{4.68}$$

In this way, it is ensured that the accumulation of all outcome probabilities gives 1:

$$\sum_{\omega \in \widehat{\Omega}^s} p_t^s(\widehat{\omega}_t^s) = 1 \tag{4.69}$$

Furthermore, instead of considering a continuous distribution another possibility would be to assume a finite probability function based on experts opinions, which would also have to fulfill Equation 4.69. Therefore, given a set of possible outcomes, every random variable realization would be assigned a certain probability:

$$p_t^n(\omega_t^s) = p_t^s = \begin{cases} p_t^0, & \text{if} \quad \omega_t^s = \widehat{\omega}_t^0 \\ p_t^1, & \text{if} \quad \omega_t^s = \widehat{\omega}_t^1 \\ \vdots & \quad\quad \vdots \\ p_t^m, & \text{if} \quad \omega_t^s = \widehat{\omega}_t^m \end{cases} \tag{4.70}$$

When it comes to choosing a continuous or discrete representation of the stochastic process, both options present different advantages and disadvantages. In terms of modelling, it is evident that real demand growth evolves in a continuous form and that assuming its discrete evolution would definitely entail a biased and incomplete representation of the actual reality. In this sense, adopting a continuos random variable is then preferred since it allows to model the stochastic demand growth in a more realistic way. However, assuming ω_t as a continuous random variable implies dealing with an infinite range of random outcomes, fact which enhances the slow convergence of the ADP algorithm: since there is a vast number of possible branches to explore across the scenario tree, the ADP algorithm faces more hurdles to learn better approximations of the value function. Furthermore, the continuous generation of new branches implies a continuous learning, thereby hampering fast convergence and algorithm stability. On the other hand, a finite distribution function represents the opposite: since the range of finite outcomes is already constrained by Equation 4.70, the number of scenario tree branches will also be constrained. This means a worse representation of reality than the one of a continuous distribution, but, in turn, it provides a faster value function update and consequently, a faster convergence of the ADP algorithm as well as more stable results.

In this MSc thesis, a finite distribution function such as equation 4.70 will be considered to prove the effectivity of approximate dynamic programming as solving method. The reason for this choice is that a finite distribution function allows to assess better the converging behaviour of the ADP algorithm during its proof of concept. Nonetheless, a normal distribution function will then be tested in Section 5.3.5 of Chapter 5 to compare the performance of both Monte Carlo sampling techniques.

4.4.3. Step 2. Problem decomposition and optimisation

In section 4.4.1, the mathematical analysis of the value function led to the a very convenient reformulation of the problem, which was crucial to determine an approximation strategy:

$$V_t = \underset{a_t \in \mathscr{A}_t}{\text{Max}} \left(\widetilde{C}_t(S_t, y_{tir}, q_{tr}) + \gamma_{t+1}\mathbb{E}\{V_{t+1}(S_{t+1}) \mid S_t\} - \sum_{i \in \mathscr{I}} c_{ti}^{disp} x_{ti}^{disp} \right) \tag{4.38}$$

Even better, the same reformulation becomes very powerful since it allows to take full advantage of OR commercial optimisers by means of exploiting a two-part decomposition. Indeed, Equation 4.38 can be decomposed into two parts that will be solved independently by the ADP algorithm:

$$\widehat{v}_t^n = \underset{a_t \in \mathscr{A}_t^n}{\text{Max}} \left(\underbrace{\widetilde{C}_t(S_t^n, y_{tir}^n, q_{tr}^n)}_{\text{Part A}} + \underbrace{\gamma_{t+1}\mathbb{E}\left\{\overline{V}_{t+1}^{n-1}(S_{t+1}^{n-1}) \mid S_t^{n-1}\right\} - \sum_{i \in \mathscr{I}} c_{ti}^{disp} x_{ti}^{disp,n}}_{\text{Part B}} \right) \tag{4.71}$$

- **Part A:** corresponds to the subproblem of calculating the recourse actions $\left(q_{tr}^n, y_{tir}^n\right)$ once previous first-stage decisions $\left(x_{t-1i}^{buy,n}, x_{t-1i}^{disp,n}\right)$ have been made and the state vector S_t^n is already known. Therefore, this subproblem, which can be solved by Gurobi independently from part B and without losing accuracy, can be written in the following way:

$$\widehat{v}_t^{A,n} = \underset{(y_t^n, q_t^n) \in \mathscr{A}_t}{\text{Max}} \quad \widetilde{C}_t(S_t^n, y_{tir}^n, q_{tr}^n) \tag{4.72}$$

subject to:

$$\begin{aligned}
\mathscr{A}_t^n(R_t^n, D_t^n) = \Big\{ \quad q_t^n, y_t^n : \quad 2\sum_{r \in \mathscr{R}} (OT_r + TAT_i) y_{tir}^n &\leq BT_i R_{ti}^n && \forall i \in \mathscr{I} && (4.73)\\
q_{tr}^n &\leq D_{tr} && \forall r \in \mathscr{R} && (4.74)\\
\sum_{i \in \mathscr{A}_t^n} cap_i LF_r y_{tir}^n &\geq q_{tr}^n && \forall r \in \mathscr{R} && (4.75)\\
\sum_{i \in \mathscr{I}} y_{tir}^n &\geq Y_{tr}^{Min} && \forall r \in \mathscr{R} && (4.76)\\
R_{0i} &= R_i^{initial} && \forall i \in \mathscr{I} \quad \Big\} && (4.77)
\end{aligned}$$

with:

$$R_{ti}^n \in \mathbb{Z}^+, y_{tir}^n \in \mathbb{Z}^+, q_{tr}^n \in\in \mathbb{R}^+ \tag{4.78}$$

- **Part B:** refers to the subproblem of choosing the first-stage decision variables for next subproblem $\left(x_{ti}^{buy,n}, x_{ti}^{disp,n}\right)$ so as to maximise the expected value function $\mathbb{E}\left\{\overline{V}_{t+1}^{n-1}(S_{t+1}^n) \mid S_t^n\right\}$ penalized by the disposal costs $\sum_{i \in \mathscr{I}} c_{ti}^{disp} x_{ti}^{disp,n}$ of the present year. It can be formulated as follows:

$$\begin{aligned}
\widehat{v}_t^{B,n} &= \underset{(x_{ti}^{buy,n}, x_{ti}^{disp,n}) \in \mathscr{Y}_t}{\text{Max}} \left(\gamma_{t+1}\mathbb{E}\left\{\overline{V}_{t+1}^{n-1}(S_{t+1}^n) \mid S_t^n\right\} - \sum_{i \in \mathscr{I}} c_{ti}^{disp} x_{ti}^{disp,n} \right)\\
&\approx \underset{(x_{ti}^{buy,n}, x_{ti}^{disp,n}) \in \mathscr{Y}_t}{\text{Max}} \left(\gamma_{t+1} \sum_{\omega \in \widehat{\Omega}^n} p_t^n(\widehat{\omega}_{t+1}^n) \overline{V}_{t+1}^{n-1}\left(S^{Model}\left(S_t^n, a_t^n, \widetilde{\omega}_{t+1}\right)\right) - \sum_{i \in \mathscr{I}} c_{ti}^{disp} x_{ti}^{disp,n} \right)
\end{aligned} \tag{4.79}$$

subject to:

$$\mathscr{Y}_t^n(R_t^n, R_{t+1}^n) = \left\{ x_{ti}^{buy,n}, x_{ti}^{disp,n} : \quad R_{ti}^n + x_{ti}^{buy,n} - x_{ti}^{disp,n} = R_{t+1i}^n \qquad \forall i \in \mathscr{I} \right\} \tag{4.80}$$

with:

$$R_{ti}^n \in \mathbb{Z}^+, R_{t+1i}^n \in \mathbb{Z}^+, x_t^{buy,n} \in \mathbb{Z}^+, x_t^{disp,n} \in \mathbb{Z}^+ \tag{4.81}$$

Part B will need R_t^n as input and will be solved based on the outputs of the kernel regression approximation and the subroutine shown in Algorithm 2:

Algorithm 2 Subroutine for part B optimisation

procedure CALCULATION OF NEXT EXPECTED VALUE FUNCTION
  Step 0. Extract from subproblem Part A current resource vector R_t^n
  Step 1. Extract from previous observations a reduced space $\mathscr{S}_{t+1}^n$ of state vectors S_{t+1} whose approximated performance across all observed tree nodes in period $t+1$ stays above a given performance threshold θ.

$$\mathscr{S}_{t+1}^n := \left\{ \forall \widehat{S}_{t+1} \quad | \quad \overline{V}_{t+1}^{n-1} > \theta \quad \forall \quad \widehat{\omega}_{t+1}^s \right\} \tag{4.82}$$

  Step 2. Calculate expected value function approximation:
  for each *child* of current node **do**
    if *child* has already been observed **then**
      Extract its corresponding $\overline{V}_{t+1}^{n-1}(\widehat{S}_{t+1}^n) \quad \forall \widehat{S}_{t+1}^n \in \mathscr{S}_{\sqcup+\infty}^{\setminus}$.
    else
      Activate kernel regression to obtain its corresponding $\overline{V}_{t+1}^{n-1}(\widehat{S}_{t+1}^n) \quad \forall \widehat{S}_{t+1}^n \in \mathscr{S}^n$
  for each $\widehat{S}_{t+1}^n \in \mathscr{S}_{t+1}^n$ **do**

$$\gamma_{t+1}\mathbb{E}\left\{\overline{V}_{t+1}^{n-1}(\widehat{S}_{t+1}^n)\right\} = \gamma_{t+1} \sum_{\omega \in \widehat{\Omega}^n} p_t^n(\widehat{\omega}_{t+1}^n)\overline{V}_{t+1}^{n-1}\left(\widehat{S}_{t+1}^n\right) \tag{4.83}$$

    if for any $i \in \mathscr{I}$: $\widehat{R}_{t+1i}^n - R_{ti}^n < 0$ **then**
      Add up the disposal costs of aircraft of type i:

$$\gamma_{t+1}\mathbb{E}\left\{\overline{V}_{t+1}^{n-1}(\widehat{S}_{t+1}^n)\right\} = \gamma_{t+1}\mathbb{E}\left\{\overline{V}_{t+1}^{n-1}(\widehat{S}_{t+1}^n)\right\} + c_{ti}^{disp}\left(\widehat{R}_{t+1i}^n - R_{t+1i}^n\right) \tag{4.84}$$

procedure SELECTION OF FIRST-STAGE DECISION VARIABLES
  Step 3. Rank all $\widehat{S}_{t+1}^n$ from highest $\overline{V}_{t+1}^{n-1}(\widehat{S}_{t+1}^n)$ to lowest and **take** the 10 best candidates
  Step 4. Choose the future state S_{t+1}^n to explore or exploit out from the 10 best candidates by applying an ϵ-greedy rule based on a simulated-annealing approach
  Step 5. Infer the first-stage decisions $\left(x_{ti}^{buy,n}, x_{ti}^{disp,n}\right)$ from transition function:

$$R_{ti}^n + x_{ti}^{buy,n} - x_{ti}^{disp,n} \quad = \quad R_{t+1i}^n \tag{4.85}$$

  Step 6. Return expected value function approximation and first-stage decisions

Exploration versus exploitation problem: Epsilon-greedy rule based on a simulated-annealing approach

It should be reminded that the exploration versus exploitation problem plays a major role within any ADP algorithm combined with kernel regressions. This fact requires deciding on whether choosing the most promising actions based on current estimates or other less promising actions just for the sake of learning a better estimation of the value function.

The technique of ϵ-greedy is used as a strategy to explore the state-space and thus, learn better value function approximations. This is achieved by escaping from optimal actions, which are currently estimated at a given ADP iteration n. Indeed, at that iteration n there might be other actions unexplored, which have been considered suboptimal but whose value has actually been underestimated. Therefore, the principle of ϵ-greedy consists in preventing the algorithm from always choosing the best candidate actions known at iteration n: with probability ϵ it chooses randomly a suboptimal action $a_t^n \in \mathscr{A}_t^n$, while with probability $1-\epsilon$ it chooses the estimated best candidate actions.

Nevertheless, a complete random selection of actions makes little sense for the multi-period fleet planning problem. Indeed, it can be deduced straightforwardly that a fleet performing really badly at the initial approximation $\overline{V}_t^0(S_t) = \lambda(T-t)\,C_t^*\,(S_t)$ will most likely not be the right candidate to explore. Following this logic, a simulating annealing (SA) approach is applied so as to control more efficiently the random selection of candidate actions. First of all, the subroutine in algorithm 2 extracts a top-10 list of the best candidate states $\widehat{S}^n_{t+1}$ at iteration n. The probability of the algorithm choosing this list is 1, thereby filtering completely the selection of actions amongst the estimated 10 best. It must be noted that between iterations, the 10 best candidates can vary due to the obtention of new estimations. Given this list, the SA controller enhances state exploration at the early stages and exploitation at the last stages. To do so, the SA controller reduces the probability ϵ monotonically throughout iterations and modifies its distribution across the 10 best candidates.

Given five ordered iteration numbers $N_1 < N_2 < N_3 < N_4 < N_{stop}$, the following intervals can be defined:

- **Free exploration:** occurs in early-mid iterations (around 43% of total iterations) between $0 < n < N_1$. At that iteration, all 10 candidates are giving the same probability ($\epsilon = 0.1$) to be chosen. Therefore, the algorithm is allowed to explore freely amonst the 10 best candidates.

$$\epsilon = [0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1]$$

- **Selective exploration:** takes place in mid iterations (around 31% of total iterations) between $N_1 < n < N_2$ and the probability starts to be greater for the three best candidates, while the 5 last are discarded.

$$\epsilon = [0.4, 0.2, 0.2, 0.1, 0.1, 0, 0, 0, 0, 0]$$

- **Exploitation with punctual explorations:** occurs in mid-advanced iterations between $N_2 < n < N_3$ (around 12% of total iterations) and the probability of selecting the best performing candidates accentuates more.

$$\epsilon = [0.75, 0.1, 0.05, 0.05, 0.05, 0, 0, 0, 0, 0]$$

- **Exploitation of best candidates:** is found in the last number of iterations between $N_3 < n < N_4$ (around 6% of total iterations) and just considers the two best candidates.

$$\epsilon = [0.85, 0.15, 0, 0, 0, 0, 0, 0, 0, 0]$$

- **Exploitation of best candidate:** takes place in the very last iterations between $N_4 < n < N_{stop}$ (around 8% of total iterations) and consists in exploiting the best estimated solution.

$$\epsilon = [1, 0, 0, 0, 0, 0, 0, 0, 0, 0]$$

4.4.4. Step 3. Simulation of next period impact

Once the optimisation subproblem has been decomposed and its corresponding parts A and B have been solved, the simulation of next period impact is directly obtained from Equation 4.86:

$$\begin{aligned}
\widehat{v}^n_t &= \max_{a_t \in \mathscr{A}^n_t} \left(\underbrace{\widetilde{C}_t(S^n_t, y^n_{tir}, q^n_{tr})}_{\text{Part A}} + \underbrace{\gamma_{t+1} \mathbb{E}\left\{ \overline{V}^{n-1}_{t+1}(S^{n-1}_{t+1}) \mid S^{n-1}_t \right\} - \sum_{i \in \mathscr{I}} c^{disp}_{ti} x^{disp,n}_{ti}}_{\text{Part B}} \right) \\
&= \max_{(y^n_t, q^n_t) \in \mathscr{A}_t} \left(\widetilde{C}_t(S^n_t, y^n_{tir}, q^n_{tr}) \right) + \max_{(x^{buy,n}_{ti}, x^{disp,n}_{ti}) \in \mathscr{Y}_t} \left(\gamma_{t+1} \mathbb{E}\left\{ \overline{V}^{n-1}_{t+1}(S^n_{t+1}) \mid S^n_t \right\} - \sum_{i \in \mathscr{I}} c^{disp}_{ti} x^{disp,n}_{ti} \right) \\
&= \widehat{v}^{A,n}_t + \widehat{v}^{B,n}_t
\end{aligned} \tag{4.86}$$

where the optimal values $\widehat{v}^A_t$ and $\widehat{v}^B_t$ are already calculated from parts A and B. Furthermore, the knowledge of decision vector a_t as well as the sampling of next random realization ω_{t+1} allows to simulate next state S^n_{t+1}:

$$S^g_{t+1} = S^{Model,g}(S^g_t, a_t, \omega_{t+1})$$

$$S^g_{t+1} = \begin{bmatrix} R_{t+1} \\ \Delta_{t+1} \end{bmatrix} = \begin{bmatrix} R_t + x^{buy}_t - x^{disp}_t \\ \Delta_t + \widehat{\Delta}_{t+1} \end{bmatrix} = \begin{bmatrix} R_t + x^{buy}_t - x^{disp}_t \\ (1+\omega_{t+1})\Delta_t \end{bmatrix} \tag{4.45}$$

4.4.5. Step 4. Value function update

In approximate dynamic programming complex dynamic problems are decomposed into small approximated subproblems which are optimised stepping forward in time. This core idea is necessary but not sufficient to achieve a good policy. Indeed, as its name suggests, approximate dynamic programming is based on performing just approximations of the value function and thus, several Monte Carlo simulations needs to be run so as to improve iteratively the quality of these approximations. Therefore, another core aspect of any ADP algorithm is its ability to learn iteratively as well as its ability to update the value function approximations. In this context, this section will focus on the different strategies available to update the value function.

As shown in algorithm 1, the most recently calculated observation $\hat{v}^n_t$ will be used to update the value function approximation $\overline{V}^n_t(S^n_t)$ of the following manner:

$$\overline{V}^n_t(S^n_t) = (1-\alpha_{n-1})\overline{V}^{n-1}_t(S^n_t) + \alpha_{n-1}\hat{v}^n_t \tag{4.87}$$

where α_{n-1} denotes a stepsize controlling the updating impact of the new observation on the value function approximation. Its value ranges between [0,1]. From Equation 4.87, it can be inferred that for $\alpha_{n-1} = 1$ the value function approximation will automatically be updated to $\hat{v}^n_t$. Likewise, for lower stepsize values the updated value function will result from a weighted sum between prior observations and the most recent observation. Generally speaking, higher values of α_{n-1} are recommended in an early exploration phase since they allow for faster value function learning rates. Likewise, lower α_{n-1} values are preferably chosen to enhance the algorithm stability during exploitation phases, where the best candidate policies start to be more defined.

Powell (2007) defines *stepsize rule* or *learning rate schedule* as the method for choosing the stepsize α_{n-1}. Moreover, he warns about the great influence that the selection of stepsizes has in the family of approximate value iteration algorithms, which is indeed the type of algorithm being used in our problem. Indeed, a poor selection of α_{n-1} values may hamper seriously the convergence of the algorithm. Consequently, it must be ensured that the stepsize rule guarantees a convergent behaviour of the algorithm. What is more, the most effective stepsize rule would be the one that provides the fastest rate of convergence. In any case, it is often stated in ADP literature that the selection of an effective stepsize rule is problem dependent and highly experimental. That is to say, there is not a clearly established rule for selecting α_{n-1}. Nevertheless, Powell (2007) refers to three basic conditions that any stepsize rule must fulfill to guarantee convergence:

1. Any stepsize value must be greater or equal to zero for all iterations:

$$\alpha_{n-1} \geq 0 \quad n = 1, 2, \ldots, N_{stop} \tag{4.88}$$

2. The infinite sum of all stepsize values must be divergent:

$$\sum_{n=1}^{\infty} \alpha_{n_1} = \infty \tag{4.89}$$

3. The infinite sum of all squared stepsize values must be convergent:

$$\sum_{n=1}^{\infty} (\alpha_{n_1})^2 \leq \infty \tag{4.90}$$

Taking into account this conditions, our ADP algorithm will be tested with several deterministic stepsize rules. For our proof of concept, a staircase stepsize rule will be applied of the form:

$$\alpha_{n-1} = \begin{cases} 1, & \text{if} \quad n < N_1 \\ \bar{\alpha}_1, & \text{if} \quad N_1 \leq n < N_2 \\ \bar{\alpha}_2, & \text{if} \quad N_2 \leq n < N_3 \\ \bar{\alpha}_3, & \text{if} \quad N_3 \leq n < N_4 \\ \bar{\alpha}_4, & \text{if} \quad N_4 \leq n < N_{stop} \end{cases} \tag{4.91}$$

where values follow $\bar{\alpha}_1 \geq \bar{\alpha}_2 \geq \bar{\alpha}_3 \geq \bar{\alpha}_4$ to meet the three previous conditions for convergence. For the proof of concept case, $\bar{\alpha}_n$ takes the sequence of values $\{1, 1, 0.5, 0.4, 0.4\}$. Nevertheless, a sensitivity analysis will be carried out to benchmark the performance of other stepsizes rules such as:

- **Constant stepsize rule**

$$\alpha_{n-1} = \begin{cases} 1, & \text{if} \quad n = 1 \\ \bar{\alpha}, & \text{otherwise} \end{cases} \tag{4.92}$$

- **Inverse stepsize rule**

$$\alpha_{n-1} = \frac{1}{n} \quad n = 1, 2, \ldots, N_{stop} \tag{4.93}$$

- **Generalized harmonic stepsize rule**

$$\alpha_{n-1} = \frac{a}{a+n-1} \quad a > 1 \quad \text{and} \quad n = 1, 2, \ldots, N_{stop} \tag{4.94}$$

4.4.6. Step 5. Stop criteria

For the moment, the number of algorithm iterations is controlled based on a simple stop criteria: if the n^{th} iteration surpasses a limit number N_{stop} of iterations, then the algorithm stops and postprocesses the best ADP policy found. Another possibility would be to implement a dynamic stop criteria so that the own algorithm was capable to stop whenever the error between observations was small enough. This feature is not implemented for the proof-of-concept since having the full control of the ADP algorithm is preferred to analyse its performance more easily. Consequently, the limit N_{stop} will be determined based on the dimensions of the problem tested as well as its complexity. In general, N_{stop} will increase according to higher dimensions. Chapter 5 will provide a deeper overview on this topic.

4.4.7. ADP Implementation summary

With the objective of solving the multi-period adaptive fleet planning problem by means of approximate dynamic programming, the developed methodology has been split into two parts. Firstly, the mathematical structure of the problem has been analyzed in the form of a Markov chain and scenario tree of decisions. In this context, it has been defined a modelling framework suitable for the implementation of an ADP support tool applied to the scenario tree. As previously summarized, this consisted of a state vector, decision vector, random variable, transition function, contribution function and value function together with a set of constraints defining the feasible space of actions. Secondly, once defined the problem, an ADP algorithm has been chosen and structured in its main building blocks:

- **Implementation of a scenario tree model:** an object-oriented programming perspective has been applied to reproduce the scenario tree by means of a tree data structure. Thus, all state and decision vectors are organised in nodes. Simultaneously, nodes possess problem descriptive attributes as well as parent and children classes to respect the Markov chain model.

- **Value function approximation strategy:** the development of an effective value function approximation strategy is crucial for the successful implementation of the ADP algorithm. To this end, several approximation methods have been benchmarked, leading to the conclusion that nonparametric regressions are the most suitable strategy to approximate the value function of the multi-period adaptive fleet planning problem. In particular, a separable Gaussian kernel regression has been applied to obtain smooth approximations across all non observed scenarios. As for its initialisation, an initial set of approximated observations is required to be calculated to provide kernel regression with essential problem information.

- **Monte Carlo simulations:** the ADP algorithm requires several iterations and sample realizations of the random demand growth to learn iteratively better policies for the scenario tree. This is carried out with Monte Carlo sampling, which consists in extracting random values from a known distribution of demand growth. Depending on the discrete or continuous nature of the known distribution, we can either have a finite or infinite number of possible random outcomes, thus having a finite or infinite number of possible scenario tree branches. While a continuous distribution can match better the actual demand uncertainty by slowing down the algorithm convergence, a finite distribution allows for faster learning rates of the ADP algorithm at the expense of less modelling accuracy. Therefore, a trade-off between these two aspects must be made.

- **Problem decomposition and optimisation:** with the aim of taking full advantage of the computational power of commercial optimisers, the maximisation subproblem is decomposed into two independent parts A and B. Part A is solved directly with Gurobi and relates to the calculation of operations-related actions once previous fleet-related decisions have been made. Likewise, part B corresponds to the problem of selecting the fleet-related decisions that will impact future operations. In this case, an epsilon-greedy subroutine based on simulated annealing is developed to let the algorithm explore the state space and choose other states with high potential.

- **Simulation of next period impact:** Once part A and B are solved, the addition of their maximal values leads to a new observation of the value function at that given state, which measures the impact of current decision on next period. Likewise, since all decisions are known, the next state can be simulated with the transition function.

- **Value function update:** By applying a learning schedule rate, new value function observations are accounted in the next estimation of the value function. Depending on the learning schedule rate assigned, the value function approximation will be more or less sensitive to new observations and thus will be updated faster or more slowly. In any case, all updating rules must meet the three mentioned conditions to guarantee convergence.

All in all, the combination of the above mentioned blocks composes the ADP algorithm which is applied to solve the multi-period adaptive fleet planning problem.

5

Proof of concept

The aim of this chapter is to prove the developed ADP algorithm as an effective method to solve the multi-period adaptive fleet planning problem. For this purpose, Chapter 5 will analyse the results obtained for two sets of experiments based on the model parameters initially presented in Section 5.1. The first set of experiments discussed in Section 5.2 consists in solving several deterministic versions of the multi-period fleet planning problem, where future demand evolution is assumed to be known. The interest behind this deterministic versions is that they can be formulated as a mixed integer linear program (MILP), which indeed allows to obtain tight bounds to their optimal objective values. In this way, this first set of experiments will allow to verify and benchmark the quality/CPU performance of the ADP algorithm against the classical methods used by commercial optimisers such as Gurobi. Once verified its correct performance in a deterministic experimental setup, the ADP algorithm will be tested in a second set of experiments in Section 5.3, where the multi-period fleet planning problem will be solved in its stochastic version. In this same section, the correct behaviour of the ADP algorithm will be verified, followed by a sensitivity analysis of the most important parameters of the ADP algorithm. Furthermore, since obtaining upper bounds is not longer possible for stochastic problem, the profitability impact of applying ADP policies will be compared to the airlines best practice of planning for the most-likely forecast.

5.1. Model parameters

The parameters describing the model used for the experimental setup can be mainly classified in global, aircraft-related, route-related and aircraft/route-related parameters. The latest three can be found in Table 5.1, 5.2 and 5.3 respectively. It must be noted that all data presented in these tables have been based on Repko and Santos (2017), who extrapolated data from a reference airline.

In terms of global parameters, it is assumed that every stage is equivalent to a year so the number of weeks per stage considered is $n_t = 52$. For the initial proof of concept, a total of 4 stages will be considered starting from year 0. Apart from this, since the main goal of the proof of concept is to assess the performance of the ADP algorithm and the discount factor γ_{t+1} has barely no influence on it, it is decided to set γ_{t+1} equal to 1. This implies that all future values are equal to their present day value.

From the tables, it can be inferred that the experimental model is based on a medium to long-haul star network of 20 routes with an averaged load factor of 90%. Three types of wide-body aircraft can be chosen to operate this network: B777-200, B777-300 and B787-800. These aircraft present different capacity and cost features as shown in Table 5.1, so they can be treated as competitors aircraft types within the fleet plan. It is important to notice that the fixed and disposal costs are expressed on a weekly basis and are assumed to be constant throughout the time stages considered in the different experimental tests. Furthermore, the airline is assumed to own already 10 B773s and 8 B788s.

As for route-related parameters, Table 5.2 shows the operating time per flight leg OT_r, the average fare f_r which is assumed to be constant through the years, the average demand existing at the initial period for each of the flight legs of a route D_{0r}, the load factor LF_r and the minimum frequency per route Y_{tr}^{min}, which

Table 5.1: Aircraft-related parameters

	c_{ti}^{fix}	cap_i	BT_i	TAT_i	c_{ti}^{disp}	IF_i
Aircraft	[$/week]	[-]	[h/week]	[h/flight]	[$/weekly based]	[-]
B772	140000	322	96	1.5	10000	0
B773	175000	401	96	1.5	10000	10
B788	96000	234	96	1.0	10000	8

Table 5.2: Route-related parameters.

	OT_r	f_r	D_{0r}	LF_r	Y_{tr}^{min}
Route	[h/flight]	[$/flight leg]	[-/weekly]	[%]	[-/week]
1	9.00	435	3365	0.9	0
2	6.10	267	3091	0.9	0
3	8.70	504	1724	0.9	0
4	5.25	226	2137	0.9	0
5	9.00	500	2569	0.9	0
6	3.50	298	768	0.9	0
7	3.00	260	729	0.9	0
8	4.25	325	2655	0.9	0
9	9.60	350	5000	0.9	0
10	10.50	477	1366	0.9	0
11	9.05	418	840	0.9	0
12	8.25	389	2010	0.9	0
13	7.63	386	3450	0.9	0
14	8.71	387	1500	0.9	0
15	5.96	282	2808	0.9	0
16	4.09	326	911	0.9	0
17	5.55	330	1316	0.9	0
18	4.45	314	1311	0.9	0
19	5.43	236	829	0.9	0
20	5.00	353	980	0.9	0

Table 5.3: Aircraft-route variable costs

	c_i^{var} [$/ operated route]		
Route	B772	B773	B788
1	207000	249000	150000
2	132000	158000	95000
3	207000	249000	150000
4	113000	136000	82000
5	225000	269000	161000
6	72000	87000	52000
7	75000	90000	54000
8	93000	112000	67000
9	200000	240000	144000
10	240000	288000	173000
11	210000	251000	151000
12	205000	242000	146000
13	190000	222000	126000
14	207000	249000	150000
15	128000	154000	91000
16	91000	109000	62000
17	119000	142000	85000
18	98000	117000	72000
19	117000	140000	82000
20	105000	125000	77000

is set to 0 to allow the algorithm more freedom when selecting which routes to operate. On the other hand, the aircraft-route variable costs correspond to the operating costs of operating a route with an aircraft type, which are found in Table 5.3.

Finally, a global demand growth rate is considered for all routes in the network, thereby making the assumption that all routes share common market trends. This leads to define the random variable ω_t as a scalar. Likewise, ω_t is expected to take three finite outcomes, which correspond to the most optimistic, most likely and the most pessimistic scenarios respectively. The values are:

$$\omega_t \in \left\{\omega_t^0, \omega_t^1, \omega_t^2\right\} = \{\mathrm{H}, \mathrm{M}, \mathrm{L}\} = \{+15\%, +5\%, -5\%\}$$

and its discrete probability distribution is assumed to be:

$$p_t^n(\widehat{\omega}_t^s) = p_t^s = \begin{cases} 30\%, & \text{if} \quad \widehat{\omega}_t^s = +15\% \\ 50\%, & \text{if} \quad \widehat{\omega}_t^s = +5\% \\ 20\%, & \text{if} \quad \widehat{\omega}_t^s = -5\% \end{cases}$$

5.2. Deterministic experiments

As introduced, the first set of experiments consists in solving several deterministic versions of the multi-period fleet planning problem. In any deterministic version, future demand evolution is known in advance so that ω_t has only one possible outcome at every stage t. This fact leads to a simplification of Bellman equation since there is no longer need to calculate future expected values given by the presence of uncertainty:

$$V_t(S_t) \quad = \quad \max_{a_t \in \mathscr{A}_t} \quad \big(C_t(S_t, a_t) + \gamma_{t+1} V_{t+1}(S_{t+1})\big) \qquad (4.25)$$

Indeed, this problem can be easily written in its equivalent mixed integer linear program (MILP) and consequently, be solved with exact methods. In this way, it is possible to obtain tight bounds to their optimal objective values and compare them to the results obtained with the ADP algorithm. Therefore, deterministic experiments provide the advantage of assessing both the effectivity and the limitations of the ADP algorithm in a much clearer setup than another characterised by the influence of stochastic processes. This first set of experiments is seen then as a preparation to fully implement the ADP algorithm in a stochastic scenario. To this end, the 4-stage scenario tree depicted in 5.1 is taken as reference to carry out these deterministic experiments: each of its 27 scenarios are treated as deterministic and independent, solved one-by-one with both Gurobi and the ADP algorithm. Regarding the ADP algorithm, 80 iterations were run within an averaged time of 120s and with in order to run several ADP pure loops, it was required to calculate one time beforehand a training dataset of observations, which lasted 2h. As for Gurobi, a time limit of 1000s was imposed. The results of these experiments are shown in Table 5.4, where demand variation represents the ratio between the final stage and initial demands.

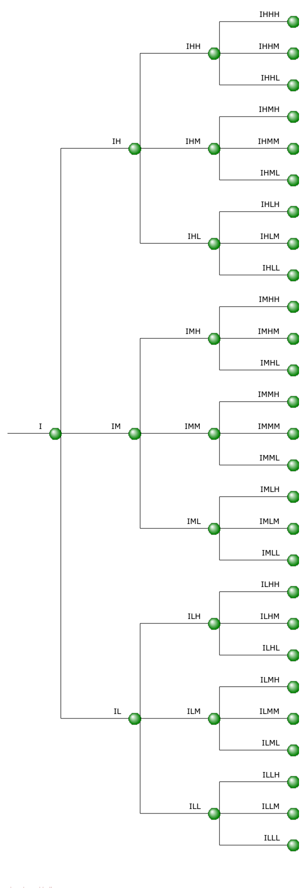


Figure 5.1: 4-stage scenario tree

Table 5.4: Results comparison between Gurobi and ADP algorithm per each deterministic tree scenario - profits expressed in USD on an averaged weekly basis

				Gurobi solution		**ADP solution with $\lambda = 0.6$**	
ID	Stage	Scenario	Demand variation	Optimality Gap	OF	Dif.Best	OF
0	3	IHHH	52.1%	0.03%	4858543	0.00%	4858543
1	3	IHHM	38.9%	0.06%	4733020	-0.10%	4728376
2	3	IHHL	25.6%	0.05%	4596565	-0.11%	4591591
3	3	IHMH	38.9%	0.02%	4616520	-0.20%	4607262
4	3	IHMM	26.8%	0.14%	4489211	0.02%	4489949
5	3	IHML	14.7%	0.02%	4371340	-0.16%	4364476
6	3	IHLH	25.6%	0.06%	4372230	-0.22%	4362763
7	3	IHLM	14.7%	0.02%	4264448	-0.24%	4254063
8	3	IHLL	3.8%	0.04%	4150656	-0.15%	4144385
9	3	IMHH	38.9%	0.10%	4512719	-0.25%	4501661
10	3	IMHM	26.8%	0.11%	4390457	-0.17%	4383097
11	3	IMHL	14.7%	0.07%	4268208	-0.10%	4264018
12	3	IMMH	26.8%	0.08%	4294605	-0.07%	4291778
13	3	IMMM	15.8%	0.06%	4185912	0.00%	4185912
14	3	IMML	4.7%	0.13%	4067694	-0.10%	4063596
15	3	IMLH	14.7%	0.01%	4064153	-0.18%	4056916
16	3	IMLM	4.7%	0.07%	3961840	-0.23%	3952813
17	3	IMLL	-5.2%	0.03%	3859255	-0.06%	3856892
18	3	ILHH	25.6%	0.07%	4171585	-0.14%	4165896
19	3	ILHM	14.7%	0.07%	4062362	-0.20%	4054393
20	3	ILHL	3.8%	0.04%	3950261	-0.29%	3938892
21	3	ILMH	14.7%	0.01%	3964795	-0.21%	3956298
22	3	ILMM	4.7%	0.04%	3862483	-0.26%	3852568
23	3	ILML	-5.2%	0.07%	3758912	-0.02%	3758080
24	3	ILLH	3.8%	0.15%	3747829	-0.15%	3742278
25	3	ILLM	-5.2%	0.20%	3654140	-0.05%	3652352
26	3	ILLL	-14.3%	0.12%	3570850	0.00%	3570850
		Average		0.07%	4177800	-0.13%	4172211

From these results, it can be stated that within a limit of 1000 seconds Gurobi solutions present an average, maximum and minimum optimality gap of 0.07%, 0.20% and 0.00% respectively. By comparison, ADP results take an average time of 120 seconds to provide an accurate fit to the optimal results obtained with Gurobi, with an average, maximum and minimum optimality gap of 0.20%, 0.35% and 0.00% respectively. When benchmarking their CPU performance, Gurobi takes an average of 35s to provide optimality gaps of the same order of magnitude as the ones presented by the ADP algorithm. This represents a quarter of the 120 seconds taken by Gurobi to run 80 iterations. The given result does not come as a surprise since the performance of Gurobi is completely optimised to solve these small test cases. Therefore, the benefits of the ADP algorithm are expected to be noticeable in those problems with a vast number of decision variables and multiple stages.

As a comparative indicator, the first column of the ADP solution (Dif.Best) shows the relative error between the objective value calculated with ADP and the one with Gurobi. What is important to infer from these results is the influence of positioning well the number of initial approximated observations that will train the kernel regression. Indeed, it is noticeable that scenarios featuring a higher density of observed nodes generally present lower relative errors than the ones fully estimated with kernel regressions. For instance, optimal solutions are obtained for the extreme and central scenarios (IHHH, IMMM, ILLL), while there is a slight error of 0.05% to 0.10% for scenarios with relatively less density of observations (e.g. IMMH, ILLM,IHHM). Likewise, scenarios with the highest dispersion of available observations (e.g. ILHL, ILMM, IMHH) may reach a maximum error of 0.3%, which is deemed acceptable for the kernel regression. This slight general increase in error occurs due to the fact that these scenarios have been fully estimated using kernel regressions. Despite this general trend, Table 5.4 proves that kernel regressions may not always perform worse than Gurobi. Indeed, the particular case of IHMM represents an exception to this tendency since its optimality gap is 0.12% and the one of Gurobi is 0.14%. Even though there is not a great difference between values, the similarity between all results is another reason to back up the ADP algorithm developed.

Overall, the average ADP relative error is 0.13%, which implies an average optimality gap for ADP of 0.2%. This magnitude is highly comparable to the optimality gaps currently seen in ADP literature. For instance, a similar behaviour was seen in Figure 2.6, where Topaloglu and Powell (2006) illustrate the typical performances for linear (L), piece-wise linear (P) and hybrid (PL) value-function approximations for a deterministic multicommodity flow problem. This fact encourages to further investigate kernel regression as an effective value function approximation strategy.

Despite these encouraging good results, a critical perspective must be applied still so as to increase the awareness regarding the limitations of the ADP algorithm and its kernel regression in particular. Figures 5.2 to 5.7 show an overview of the general learning behaviour presented by the ADP algorithm, when solving the extreme scenarios (IHHH,ILLL) and the central one (IMMM) for different values of λ: in early iterations, the algorithm learns better value function approximations due to the free and selective exploration phases, where high levels of noise are appreciated during the first set of observations. When the number of iterations advances, the algorithm reaches the exploitation phases and starts to converge, thereby reducing significantly the standard deviation between observations. However, this general behavior is highly influenced by the value of λ, which was previously introduced as a control parameter to manage the impact of the initial approximated observations on the entire ADP algorithm. Since the initially approximated value function is assumed to be the deterministic aggregation of equal operating profits $\widetilde{C}_t^*$ from period t onwards

$$\overline{V}_t^0(S_t) \quad = \quad \lambda\,(T-t)\,\widetilde{C}_t^*\,(S_t)\,, \tag{4.52}$$

it may happen that the value function is overestimated or underestimated in certain scenarios. This can be easily noticed amongst the first observations: if the first observations are above the optimal value, then the initial approximation is overestimated and viceversa. From Figure 5.6, it is crucial to realize that overestimations have adverse effects on the convergence of the ADP algorithm. Indeed, since the value of being in all states is initially overestimated, the subsequent updating of values will make high potential states fall from the most promising category to the least promising ones. Since all states might be overestimated, this will trigger the adverse loop of continuously visiting the most promising candidates, which will in turn fall from the best to least promising positions. It is not until all states have been visited and updated that this loop will stop and the algorithm will start to converge. Evidently, visiting all possible states is infeasible in terms of time and should be completely avoided. For this reason, the control parameter λ is introduced as a mitigating measure. Its high impact is very visible from the left to right figures. For this particular case, a reduction of λ

from 1 to 0.6 guarantees the convergence of the ADP algorithm in the extreme and central scenarios (IHHH, IMMM and ILLL). However, it should be taken into account that the lower the value of λ is, the higher level of noise will be found amongst observations, thereby slowing down the rate of convergence. Therefore, lowering too much the value of λ can also entail negative effects on convergence. Due to the satisfactory results obtained with $\lambda = 0.6$, the same value was applied to solve the rest of the deterministic scenarios shown in Table 5.4. Furthermore, it should be highlighted that if the value of $\widetilde{C}_t^*$ was negative, λ equal to 0.6 would not do anything else apart from enhancing the overestimation. In this case, λ should be at least greater than 1 to decrease more the negative value function $\overline{V}_t^0(S_t)$

Finally, the behavior of the ADP algorithm is quite similar across the observed and non observed scenarios. That is to say, kernel regression does not affect significantly the algorithm rate of convergence once λ is properly fixed to avoid overestimations. In case the reader is interested to see this behavior, Appendix B presents the ADP behavior seen in other scenarios fully estimated with kernel regressions. The optimal fleet plans for each of the independent scenarios can also be found in the same appendix.

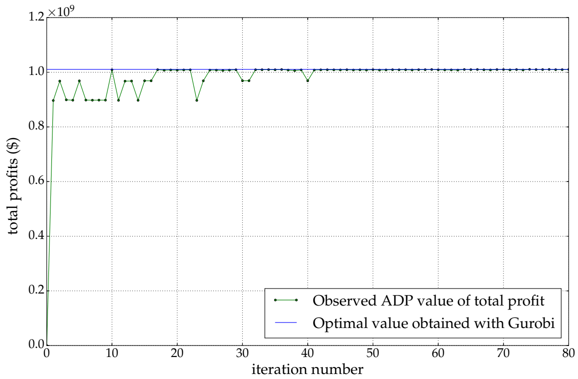


Figure 5.2: IHHH scenario with $\lambda = 1$

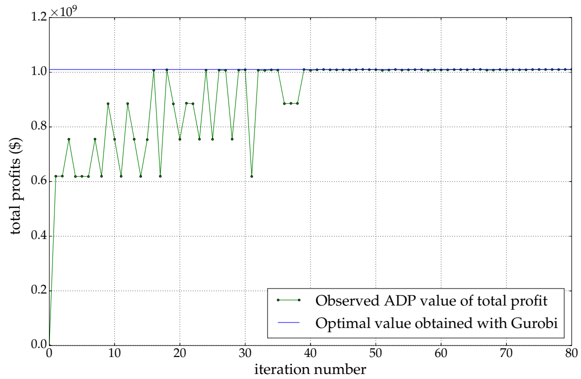


Figure 5.3: IHHH scenario with $\lambda = 0.6$

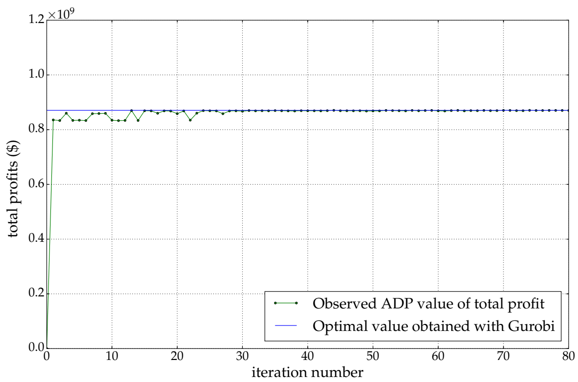


Figure 5.4: IMMM scenario with $\lambda = 1$

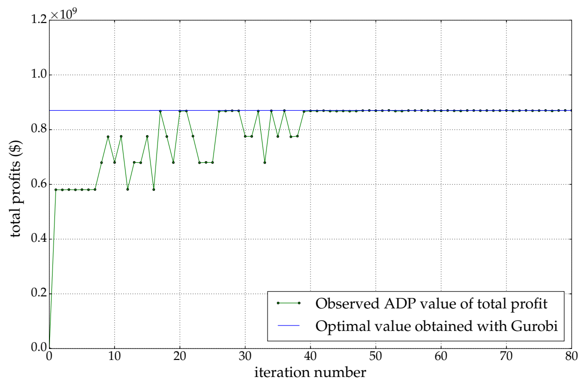


Figure 5.5: IHHH scenario with $\lambda = 0.6$

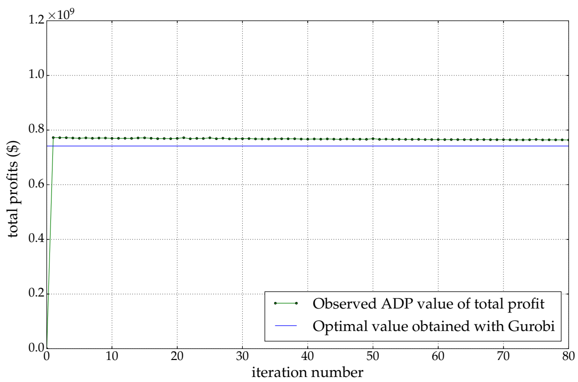


Figure 5.6: ILLL scenario with $\lambda = 1$

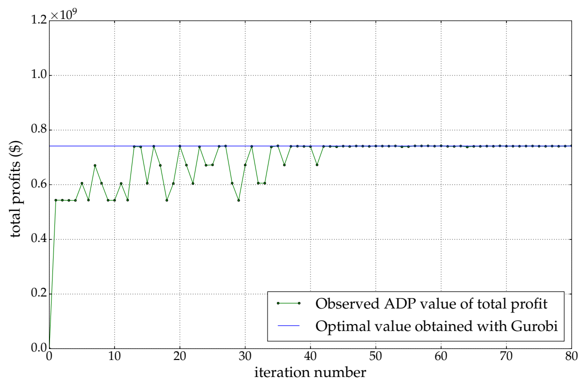


Figure 5.7: ILLL scenario with $\lambda = 0.6$

5.2.1. Conclusions of deterministic experiments

The assessment of the baseline ADP algorithm in a deterministic setup leads to the following conclusions:

- Firstly, the ADP results provide an accurate fit to the optimal results obtained with Gurobi. While Gurobi solutions present an average, maximum and minimum optimality gap of 0.07%, 0.20% and 0.00% within a limit of 1000 seconds, the ADP algorithm reaches within less than 120 seconds (80 iterations) an average, maximum and minimum optimality gap of 0.20%, 0.35% and 0.00% respectively.

- Secondly, Gurobi stands as the best technique to solve these small test cases in terms of CPU performance: it takes an average of 35s to provide optimality gaps of the same order of magnitude as the ones presented by the ADP algorithm within 120s. Nevertheless, a time limit of 1000s is imposed to Gurobi when solving each independent scenario, since it can easily take more than 1h to reach a fully optimal solution for certain scenarios.
  This results are not surprising as the benefits of the ADP algorithm are expected to be noticeable in those problems with a vast number of decision variables and multiple stages. Indeed, it is in that context that the initial 2h-calculation of the kernel training dataset starts to make sense.

- Thirdly, the approximation strategy provides a good initial estimation and a fast convergence: in less than 40 iterations an optimality gap of 0.5% is achieved. Nevertheless, the control parameter λ needs to be correctly tuned so as to avoid convergence issues due to adverse value function overestimations.

- Lastly, the ADP algorithm has proven to be a satisfactory robust method to solve effectively the deterministic version of the multi-period fleet planning problem.

5.3. Stochastic experiments

For the proof of concept, the stochastic version of the multi-period fleet planning problem is modelled with a 4-stage scenario tree to model the impact of demand uncertainty. In contrast to the previous deterministic experiments, here the demand outcome in next stages is uncertain and modelled by different tree branches stemming from common nodes. Thus, each branch represents a possible outcome of the stochastic demand. That is to say, the multi-period adaptive fleet planning problem will be solved as a Markov-chain with all its scenarios coupled. By coupled scenarios is meant that the model takes into account all interdependencies existent amongst scenarios with common nodes. Therefore, scenarios will no longer be solved independently as done in the deterministic experiments, but making sure that the same decisions are taken in their common nodes. Consequently, the introduction of stochastic processes eliminates predictability since value function updates are also completely coupled amongst them: the value update of being in a state under certain scenario conditions may affect the value of being in another state under different scenario conditions.

It must be reminded that the approximation parameters are set to the same values as in the deterministic version: $\lambda = 0.6$ and $G = 3$. This choice is initially based on the satisfactory results reported by the verification of the deterministic experiments. Nevertheless, the reader must note that these values are subject to be tuned in next sections: what performs well in a deterministic setup may not necessarily be the most adequate option within a stochastic environment. It is for this reason that a sensitivity analysis and calibration of the most important ADP parameters will be carried out after having verified the algorithm behaviour for the firstly selected parameters.

5.3.1. Verification of the ADP algorithm behaviour

A 4-stage stochastic fleet planning problem is solved with the same model parameters presented in Section 5.1. Even though the parameters are the same as the ones used in the deterministic experiments, the stochastic case is completely different since it integrates all node interdependencies existent amongst scenarios. In contrast to a single deterministic scenario, a much higher number of iterations is expected to be necessary to guarantee the convergence of the ADP algorithm. Given that the stochastic problem features multiple scenarios, longer exploration times are needed to visit a wider range of possible states. In this way, the initial value function approximation will be improved iteratively until reaching convergence and coherent results.

Figure 5.8 illustrates the behaviour of the ADP algorithm when solving the stochastic problem for 3000 iterations. As in the deterministic experiments, the green dots correspond to the observations of the total operating profit expected to be earned during the 4 periods: $V_0(S_0)$. The red line shows the rolling average for the last 20 subsequent observations. The latter has been plotted so as to picture the general trend of the observations throughout iterations.

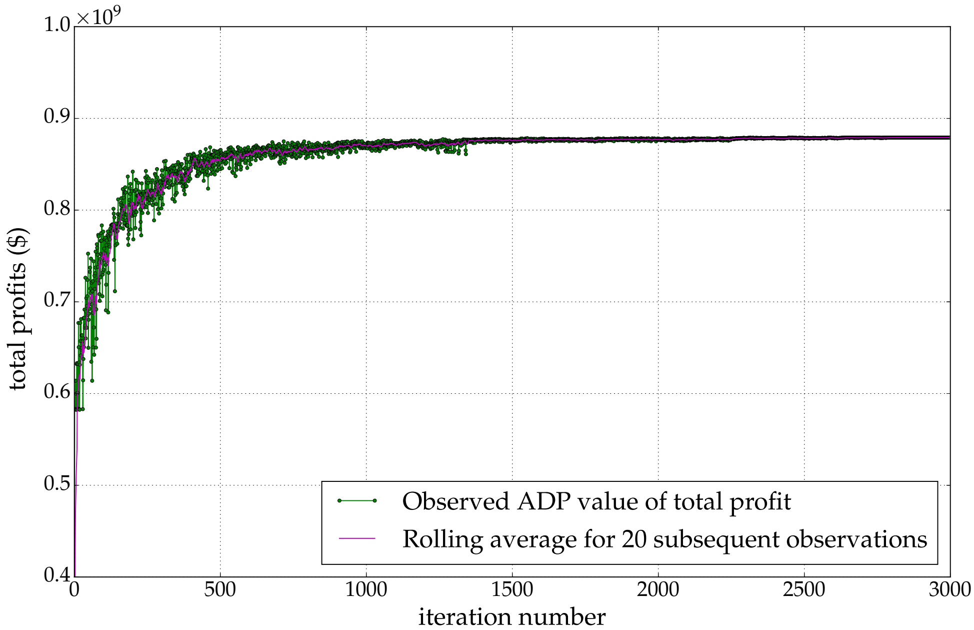


Figure 5.8: Behaviour of the ADP algorithm in solving the 4-period fleet planning problem for 3000 iterations

The behaviour of the ADP algorithm shares certain similarities to the one appreciated in a deterministic setup, but introduces more complexities. As expected, the algorithm is highly influenced by the presence of the stochastic process $\omega_{[t]}$: Monte Carlo simulations sample continuously different scenarios, thus making state exploration more complex across scenarios. In addition to this, value function estimates are continuously updated with random observations coming across all time periods and scenarios. Since value function updates are completely coupled amongst them, the value update of being in a state under certain scenario conditions may affect the value of being in another state under different scenario conditions. This leads to high levels of noise in early iterations and thus, higher margins for the standard deviation between value function observations. This is clearly seen in Figure 5.9 and 5.10, where the rolling mean and standard deviation for 20 subsequent observations are plotted together for the iterations intervals [0,1500] and [1500, 3000]. In fact, both graphs illustrate in a more neater way the main trends of the ADP algorithm behavior together with an overview of its CPU performance.

Figure 5.9 shows that despite the high oscillations experienced in the initial iterations, the algorithm keeps steadily improving its estimations of the value function. Starting from $\$4x10^8$ and eventually surpassing $\$8.5x10^8$, it is during the first half of iterations that the algorithm presents the best value function learning rate and starts to reach convergence. In fact, Figure 5.10 shows that a smaller improvement is still made during the last half of iterations. Although this improvement may not be seen as essential, the performance of different simulations has shown that it is precisely during this last phase that the best found ADP policy achieves major coherence and thus, higher quality. Apart from this, it is interesting to see the exponential reduction experienced by the rolling standard deviation. This tendency is depicted in both figures 5.11 and 5.12. Starting from high levels during the exploration phases, the standard deviation values are exponentially reduced when reaching the exploitation phases.

Even though a minimum of 1500 iterations is needed to reach convergence, several simulations led to the conclusion that 3000 iterations provided much better values of the objective function as well as a more coherent ADP policy. Nevertheless, running 3000 times the algorithms presented in Chapter 4 (ADP pure loop) translates into a CPU time cost of 17804s ($\approx$ 5h), without considering the creation of the training dataset that lasted 4h. The reader can see the computational timeline of the ADP pure loop in both Figures 5.9 and 5.10. The sharp increase in computational time is evident, thus becoming a crucial matter to be particularly discussed and investigated in upcoming Chapter 6.

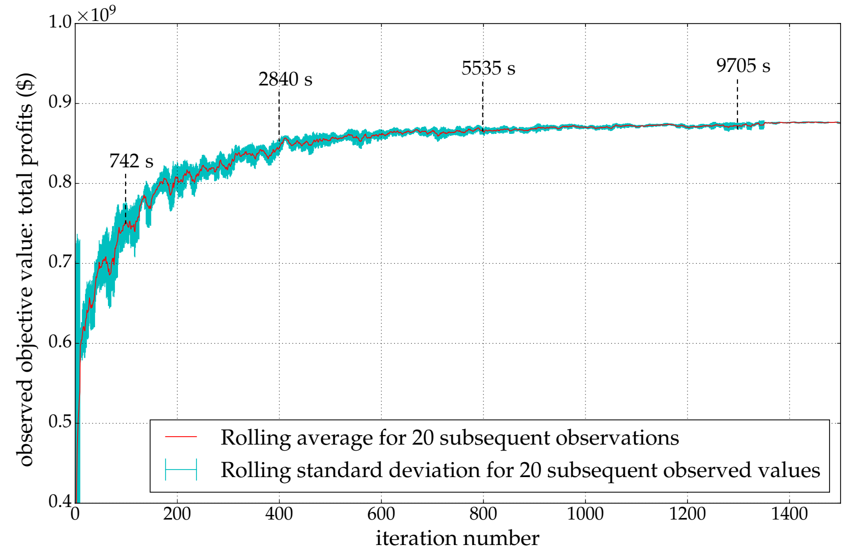


Figure 5.9: Rolling mean and standard deviation for the first half of iterations

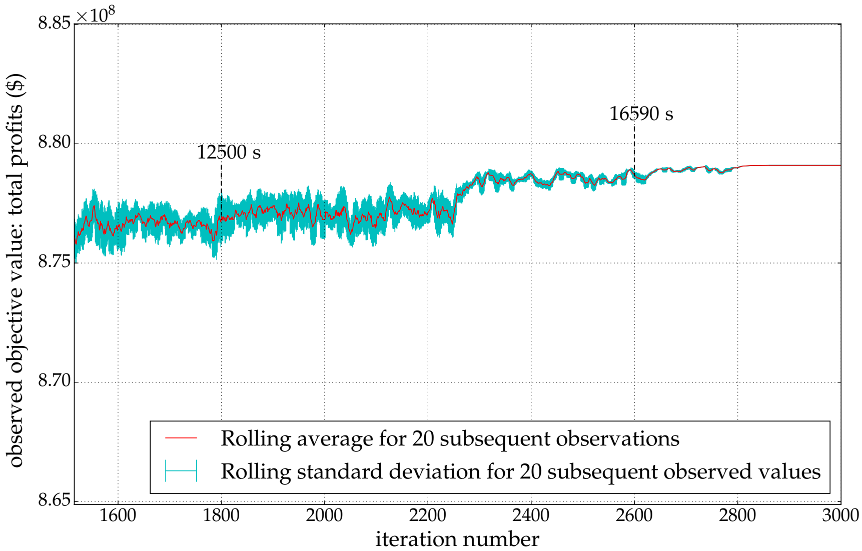


Figure 5.10: Rolling mean and standard deviation for the second half of iterations

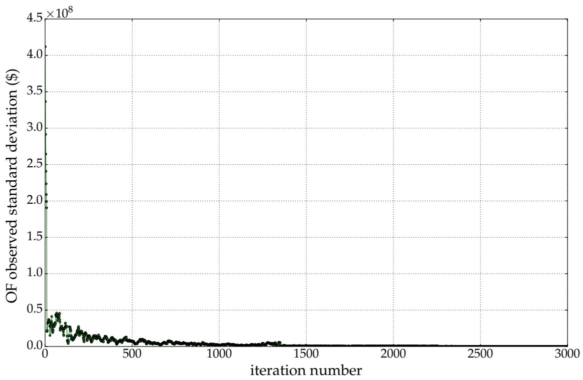


Figure 5.11: Overview of the exponential reduction in standard deviation

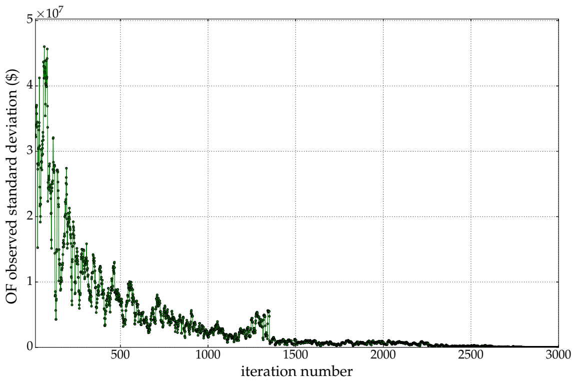


Figure 5.12: Detail of exponential reduction after abrupt decay of the highest deviation values existing in early iterations

5.3.2. Recommended ADP fleet policy and its profitability impact

While presenting the methodology in Chapter 4, it was stated that solving a multistage stochastic program does not lead to a single result, but a policy $A^{\pi}(S_t)$: the best rule for making decisions a_t in function of the given state S_t in time t, which results from the previous multivariate stochastic process $\omega_{[t]}$.

After finishing the 3000 iterations, the ADP algorithm recommended the adaptive fleet policy outlined in Figure 5.13. All fleet data is expressed in the vector format (B772/B773/B788). In this way, blue vectors express each of the values of the resource vector R_{ti} associated to a state S_t. For instance, the initial resource vector describes the initial fleet status of the airline: it has 0 aircraft of type B772, 10 aircraft of type B773 and 8 aircraft of type B778. Likewise, yellow vectors correspond to the fleet related decisions (x_t^{buy}, x_t^{disp}) recommended to be made during period t given the state S_t. Thus, positive values imply new aircraft acquisitions for next year whereas negative values, disposals for next year.

Table 5.5: Optimal fleet plans for deterministic scenarios

					Optimal fleet for independent scenarios			
ID	Stage	Scenario	Total probability	Demand variation	year 0	year 1	year 2	year 3
0	3	IHHH	2.70%	52.1%	0/10/8	0/10/12	0/13/12	0/16/12
1	3	IHHM	4.50%	38.9%	0/10/8	0/10/12	0/13/12	0/14/12
2	3	IHHL	1.80%	25.6%	0/10/8	0/10/12	0/12/12	0/12/12
3	3	IHMH	4.50%	38.9%	0/10/8	0/10/12	0/11/12	0/13/13
4	3	IHMM	7.50%	26.8%	0/10/8	0/11/10	0/12/11	0/12/12
5	3	IHML	3.00%	14.7%	0/10/8	0/10/12	0/11/12	0/10/12
6	3	IHLH	1.80%	25.6%	0/10/8	0/10/11	0/10/11	0/12/12
7	3	IHLM	3.00%	14.7%	0/10/8	0/10/11	0/10/11	0/10/12
8	3	IHLL	1.20%	3.8%	0/10/8	0/10/11	0/10/11	0/9/11
9	3	IMHH	4.50%	38.9%	0/10/8	1/10/8	1/12/9	1/15/9
10	3	IMHM	7.50%	26.8%	0/10/8	1/10/8	1/12/9	1/12/11
11	3	IMHL	3.00%	14.7%	0/10/8	1/10/8	1/12/9	0/12/9
12	3	IMMH	7.50%	26.8%	0/10/8	1/10/8	0/11/9	0/14/9
13	3	IMMM	12.50%	15.8%	0/10/8	1/10/8	0/11/9	0/11/11
14	3	IMML	5.00%	4.7%	0/10/8	1/10/8	1/11/8	1/10/8
15	3	IMLH	3.00%	14.7%	0/10/8	1/10/8	0/10/8	0/10/12
16	3	IMLM	5.00%	4.7%	0/10/8	1/10/8	0/10/8	1/10/8
17	3	IMLL	2.00%	-5.2%	0/10/8	1/10/8	0/10/8	0/10/8
18	3	ILHH	1.80%	25.6%	0/10/8	0/10/8	0/11/9	0/12/12
19	3	ILHM	3.00%	14.7%	0/10/8	0/10/8	0/11/9	0/11/10
20	3	ILHL	1.20%	3.8%	0/10/8	0/10/8	0/11/9	0/10/9
21	3	ILMH	3.00%	14.7%	0/10/8	0/10/8	0/10/8	0/10/12
22	3	ILMM	5.00%	4.7%	0/10/8	0/10/8	0/10/8	1/10/8
23	3	ILML	2.00%	-5.2%	0/10/8	0/9/9	0/9/9	0/9/9
24	3	ILLH	1.20%	3.8%	0/10/8	0/9/9	0/8/9	1/9/10
25	3	ILLM	2.00%	-5.2%	0/10/8	0/9/8	1/8/8	1/8/9
26	3	ILLL	0.80%	-14.3%	0/10/8	0/9/9	0/8/9	0/8/8

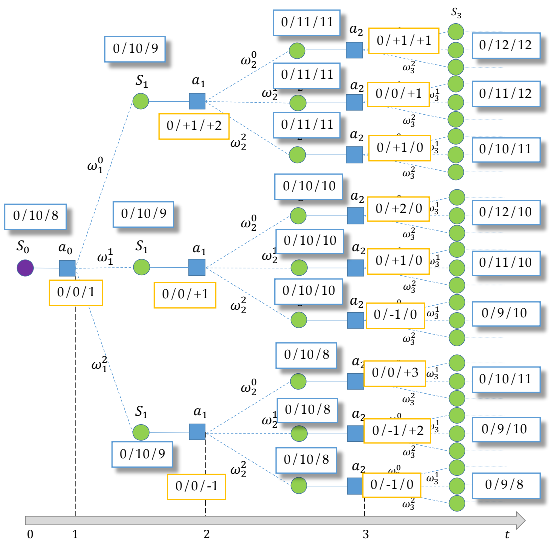


Figure 5.13: ADP fleet policy

Given an airline with an initial fleet of 18 aircraft in total (0/10/8) and the route demands seen in Table 5.2, the best action to take at that moment is to acquire one B788 more. The fleet (0/10/9) is expected to be the best performer across all forecasted scenarios of possible demand growth (H,M,L) within next year 1. Once the demand for year 1 is less uncertain, the ADP policy will provide different recommendations depending on the stochastic development of demand growth forecasted for year 2. In very optimistic scenarios (IH), all 20 long-haul routes would experience a global demand increase of 15%, which would imply the recommendation to buy 3 aircraft more: one B773 and two B788. However, this situation is only likely to happen at a maximum probability of 30%. Indeed, the most-likely scenario with 50% probability is the one in which demand experiences a general growth of 5% for next period (IM). Since in this scenario demand still grows but at lower rate, the recommendation consists in acquiring just one B788 more. A possible decay of -5% in demand should also be considered with 20% of probability. If this outcome happened to be certain for year 1, then the fleet policy would advise to dispose one B788 for next year 2. Throughout the subsequent time periods, fleet recommendations are simultaneously based on current airline performance and future demand forecasts. This is clearly appreciated in year 2 and 3: higher levels of aircraft investment are noticeable for those states benefiting from a history of continuous growth (IHH, IHM, IMH, IMM) and a good outlook for demand. Whereas, states affected by a decay in demand will most likely face worse demand forecasts and thus, will be advised with more preventive measures consisting of maintaining the fleet size or even reducing it (IHLL, IMLL, ILLL).

In light of these results, it is evident that the fleet policy adapts to the different conditions defining each of the states. This is the reason why multistage stochastic models lead to an adaptive fleet policy, whose recommendations are intended to adjust as much as possible to the optimal fleet plans for each independent scenario. This is appreciated when comparing both the optimal fleet plans for each deterministic scenario in Table 5.5 and the fleet policy schematized in Figure 5.13. Even though the recommended fleet might not be exactly the same for a given scenario, in most of the cases the fleet size is adjusted so as to have the best performance across all common states branching from each node. Furthermore, these differences tend to be minimized to respect as much as possible the optimal fleet compositions and sizes, which would indeed correspond to the equivalent deterministic scenarios. Apart from that, it must be noticed that the scenarios with higher probability exert greater influence on the resulting fleet policy, which is adjusted more closely to their optimal fleet sizes and compositions. Indeed, the combined probability of forecast H and M amounts to 80%, which implies a higher weight when considering next fleet decisions. Nevertheless, a total fit is very hard to happen since the adaptive policy also takes into account the potential occurrence of other scenarios with less probability. For instance, it may happen that a certain fleet, which is deemed suitable for the next forecasted optimistic scenarios, performs so poorly for the pessimistic ones that it could be immediately rejected by the ADP algorithm. All in all, the presence of demand uncertainty explains why the overall adaptive policy tends to be more conservative in its decisions, when compared to the deterministic scenarios.

Furthermore, a very unique feature of the adaptive fleet policy is its capability of capturing the path dependency effect amongst the different results. Indeed, it may happen that two scenarios in year 3 share the same total demand variation with respect to year 0. An example of this situation can be seen in scenarios IHHL and IHLH shown in Table 5.6. Both with a total demand variation of 25.6% in year 3, the two scenarios have different fleet compositions and sizes eventually. This is a direct consequence of the different stochastic demand process $\omega_{[t]}$ undergone by each of the scenarios. On the one hand, IHHL benefits from an outstanding continuous growth until the last period, where an abrupt demand decay takes place. In this context, a higher risk is taken by the algorithm due to the excellent financial performance experienced so far and the good demand prospects expected: there is only a 20% probability that the total demand variation will be lower than 25.6% in stage 3. This results in having the fleet (0/12/12). On the other hand, IHLH attains the same demand variation by experiencing a different stochastic chain of events. Due to a sudden reduction of -5% in year 2, its fleet expansion is slowed down significantly, thereby taking more precautions against future demand uncertainty. Indeed, at that stage the forecasted probability of attaining the same total growth as in IHHL is just 30%. Clearly affected by the path followed, the fleet in year 3 of scenario IHLH only reaches a size of (0/10/11). However, the adaptive policy is also capable of overcoming the adverse effects of a past demand reduction provided that a positive sequence of demand growth follows. This is appreciated in scenario ILHH, where the airline fleet is initially reduced - (0/10/8) - due to a demand decrease in year 1 and then expanded for next year due to demand recovery - (0/10/11) -. In this situation, the path dependency is still very visible between ILHH and IHHL: even though IHHL has the same total demand variation, it features a greater fleet of (0/12/12) due to its good historical data.

Table 5.6: Example of path dependency amongst scenarios with same demand variation

				Recommended fleet plan			
ID	Scenario	Total probability	Demand variation	year 0	year 1	year 2	year 3
2	IHHL	1.8%	25.6%	0/10/8	0/10/9	0/11/11	0/12/12
6	IHLH	1.8%	25.6%	0/10/8	0/10/9	0/11/11	0/10/11
18	ILHH	1.8%	25.6%	0/10/8	0/10/9	0/10/8	0/10/11

In light of the previous discussion, it is important to understand that adaptive fleet policies offer a new perspective in resource planning: they do not aim at simply forecasting the future, but at including uncertainty into the decision-making process. As will be seen in Chapter 7, the final objective of solving wide scenario trees consists in generating meaningful big data so as to draw signposts for fleet planners. In other words, the solution of wide scenario trees would pave the way for the development of advanced analytics tools capable of defining preventive, corrective and even predictive measures to steer strategic plans towards success.

Since obtaining an upper bound is not longer possible for stochastic problems, the profitability impact of applying ADP policies will be compared to the airlines best practice of always planning for the most-likely forecast (IMMM). Indeed, it is a common practice amongst airlines to forecast the most-likely scenario and plan their fleet according to it, regardless of the existent uncertainty that other very different scenarios may occur in the future.

In this sense, Table 5.7 shows a comparison between the profits that would be obtained when just planning for the most-likely scenario and the ones that would be expected when following the fleet planning recommendations from the adaptive policy. With the aim of providing an effective reference for the analysis, every scenario composing the tree is also optimized individually as an equivalent deterministic problem. Evidently, their optimal objective values coincide with the values presented in the deterministic experiments, since they do not consider uncertainty nor the interdependencies between scenarios at every tree node. Therefore, the optimal fleet plan obtained for a single deterministic scenario will always be better or equal to the one recommended by the ADP policy for that specific scenario. Applying the same logic, the optimal fleet plan for the most-likely scenario does not necessarily have to be the optimal fleet plan for the rest of scenarios and thus, its corresponding objective value does not imply a maximum.

In view of the above, Table 5.7 is divided into three main columns: best solution for independent scenarios, most-likely solution and ADP policy. Each column shows the weekly average operating profits that would be obtained from applying each method. As discussed, the best solutions for independent scenarios can be

Table 5.7: Performance analysis of recommended ADP policy against most-likely fleet plan and optimal fleet plans for independent deterministic scenarios - operating profits expressed on an averaged weekly basis in USD

					Best deterministic solution		**Most-likely solution**		**ADP policy**	
ID	Stage	Scenario	Total probability	Demand variation	Optimality Gap	OF	Dif.Best	OF	Dif.Best	OF
0	3	IHHH	2.70%	52.1%	0.03%	4858543	-1.65%	4778148	-0.89%	4815384
1	3	IHHM	4.50%	38.9%	0.06%	4733020	-1.21%	4675713	-0.56%	4706454
2	3	IHHL	1.80%	25.6%	0.05%	4596565	-0.89%	4555486	-0.38%	4579012
3	3	IHMH	4.50%	38.9%	0.02%	4616520	-0.90%	4575030	-0.46%	4595088
4	3	IHMM	7.50%	26.8%	0.14%	4489211	-0.52%	4465826	-0.20%	4480146
5	3	IHML	3.00%	14.7%	0.02%	4371340	-0.49%	4350090	-0.49%	4350065
6	3	IHLH	1.80%	25.6%	0.06%	4372230	-0.33%	4357830	-0.46%	4352260
7	3	IHLM	3.00%	14.7%	0.02%	4264448	-0.27%	4253117	-0.40%	4247546
8	3	IHLL	1.20%	3.8%	0.04%	4150656	-0.99%	4109537	-1.12%	4103967
9	3	IMHH	4.50%	38.9%	0.10%	4512719	-0.64%	4483846	-0.46%	4491971
10	3	IMHM	7.50%	26.8%	0.11%	4390457	-0.36%	4374641	-0.31%	4377028
11	3	IMHL	3.00%	14.7%	0.07%	4268208	-0.22%	4258906	-0.50%	4246948
12	3	IMMH	7.50%	26.8%	0.08%	4294605	-0.14%	4288516	-0.27%	4283176
13	3	IMMM	12.50%	15.8%	0.06%	4185912	4185912	0.00%	-0.08%	4182543
14	3	IMML	5.00%	4.7%	0.13%	4067694	-0.68%	4039959	-0.28%	4056129
15	3	IMLH	3.00%	14.7%	0.01%	4064153	-0.54%	4042312	-0.46%	4045551
16	3	IMLM	5.00%	4.7%	0.07%	3961840	-1.32%	3909490	-0.36%	3947527
17	3	IMLL	2.00%	-5.2%	0.03%	3859255	-2.49%	3763271	-0.73%	3831189
18	3	ILHH	1.80%	25.6%	0.07%	4171585	-0.48%	4151630	-0.82%	4137477
19	3	ILHM	3.00%	14.7%	0.07%	4062362	-0.38%	4046916	-0.53%	4040630
20	3	ILHL	1.20%	3.8%	0.04%	3950261	-1.19%	3903337	-0.45%	3932507
21	3	ILMH	3.00%	14.7%	0.01%	3964795	-0.95%	3927296	-0.43%	3947860
22	3	ILMM	5.00%	4.7%	0.04%	3862483	-1.76%	3794475	-0.40%	3846972
23	3	ILML	2.00%	-5.2%	0.07%	3758912	-2.94%	3648255	-1.01%	3720966
24	3	ILLH	1.20%	3.8%	0.15%	3747829	-2.99%	3635870	-0.58%	3726117
25	3	ILLM	2.00%	-5.2%	0.20%	3654140	-4.21%	3500408	-0.39%	3639948
26	3	ILLL	0.80%	-14.3%	0.12%	3570850	-5.75%	3365645	-0.76%	3543811
			Expected Values		0.07%	4243477	-0.82%	4208652	-0.40%	4226538

seen as a reference which bounds the maximum profits achievable in each scenario of the tree. In order to measure the performance of both the ADP policy and the most-likely solution, their respective OF values will be compared to the best deterministic OF value for each independent scenario. The differences with respect to the best value (Dif.Best) are expressed in percentage.

By analyzing the percentage difference between the objective values of each method in every scenario, it can be concluded that the adaptive policy is the most robust method since it clearly outperforms the most-likely solution in the majority of scenarios. Yet, the ADP lowest scores do not differ significantly from the ones presented by the most-likely solution, thereby making clear the robustness of the adaptive policy. Evidently, the most-likely solution performs better in the most-likely scenario but its performance starts to stall the further we move away from the scope of the most-likely scenario. Its worst scores are found in the neighborhood of the extreme scenarios: For the most pessimistic scenarios the most-likely solution can imply weekly operating losses ranging between 3% and 6%, whereas in the most optimistic scenarios losses remain between 2% and 1%. In contrast, the adaptive fleet policy mitigates losses in all extreme scenarios, which never surpass the 1.1%.

Finally, the last row of Table 5.7 shows the expected weekly operating profits for the most-likely fleet plan and the ADP policy. By analyzing these last results, it can be concluded that the adaptive policy reduces by 50 % the percentage difference of the most-likely solution. This conclusion reaffirms the advantage of applying adaptive fleet policies and most importantly, proves approximate dynamic programming as an effective method to calculate them.

5.3.3. Verification of ADP results

The objective of this subsection is to verify that the ADP algorithm behaves as it is intended to do. To this end, the response of the ADP algorithm will be measured according to several changes in input data so as to assess the logic of the ADP results. It is important to notice that only model parameters will be changed in this analysis so as to decouple the impact of the ADP parameters (e.g. λ , G , N_{stop}). Furthermore, the ADP kernel regression will only be trained by the original dataset of initial observations, which were generated using the baseline parameters of the model. In other words, initial observation will not be updated according to the changes in input data. This is intended to assess the robustness of the approximation strategy and thus, the ADP algorithm in general when varying parameters such as fixed, variable and disposal costs as well as demand growth.

Impact of varying fixed and variable costs

Once set the baseline case as the multi-period fleet planning problem defined with the model parameters of Section 5.1, two additional variations of the baseline case were solved for modified values of fixed and variable costs. The first variation represents a 50% increase with respect to the baseline fixed and variable costs, while the second case consists in a 50% cost reduction.

Figure 5.14 shows the responses of the ADP algorithm when varying both costs: due to a sharp reduc-

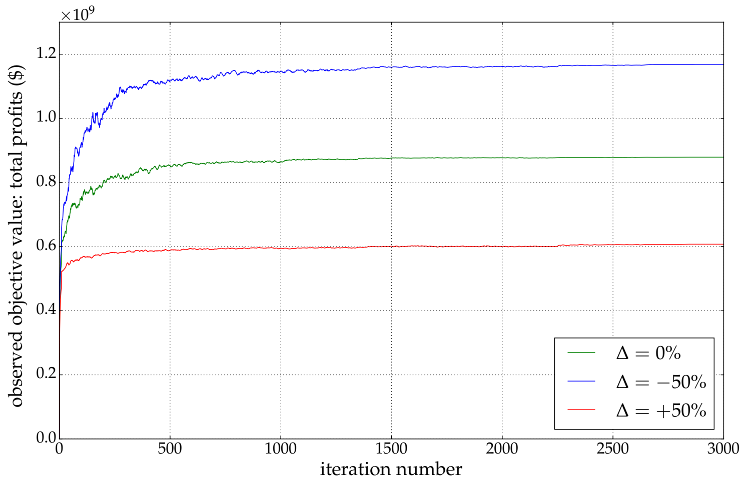


Figure 5.14: Responses of the ADP algorithm when varying fixed and demand costs

tion in costs ($\Delta = -50\%$), the total expected profits increase substantially with respect to the baseline case. In the same way, the total expected profits undergo a significant reduction given the 50% increase in costs. Therefore, it can be concluded that the ADP response behaves as expected. Furthermore, all value functions reach convergence, thereby implying the effective stabilization of the approximation strategy despite using only initial observations from the baseline case. However, it can be inferred from the case $\Delta = -50\%$ that the further the initial approximated observations are from the real tested case, the harder will be to reach convergence. Indeed, even though the value function observations improve steadily along the observations for $\Delta = -50\%$, a higher degree of noise is visible for a longer interval of iterations. This local instabilities experienced by the value function are definitely caused by poor initial estimations. Likewise, the case comprising a 50% increase in cost happens to be closer to the initial estimations, thereby converging much faster than the previous case. This good performance may appear unexpected since the initial set of observations was generated for the baseline case. However, it should be reminded that the the values of these initial observations were intentionally reduced by calibrating $\lambda = 0.6$, as a way to prevent the adverse effects entailed by potential overestimations. Consequently, it is deduced that the ADP algorithm will start to fail for a cost increase greater than 50%. If this was needed, then the kernel regression should be initialized with a new set of approximated observations generated from the new required parameters. A similar logic could be applied when decreasing the costs: if a reduction of more than 50% is needed, it would take more iterations for the algorithm to converge. Nevertheless, the probability of requiring a cost reduction of more than 50% is quite low. In this sense, it can be concluded that the behavior of the ADP algorithm is robust and highly stable, boasting a range of [-50% , 50%] for changing cost values without losing accuracy.

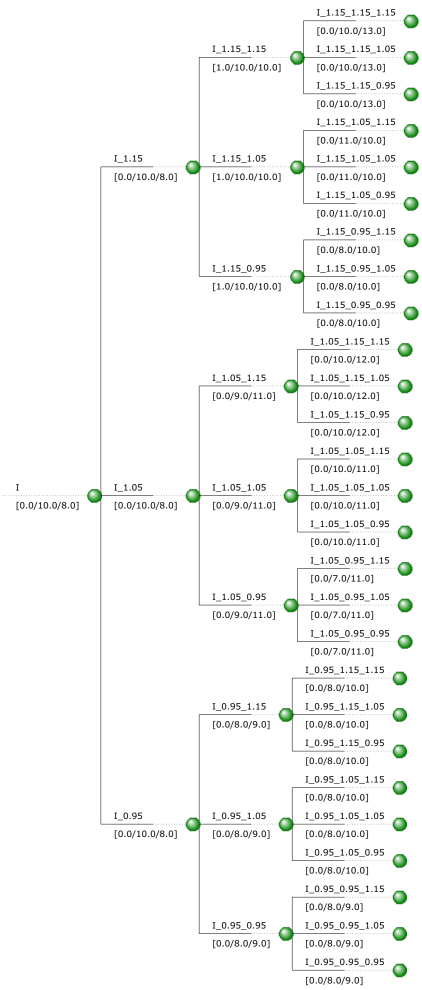


Figure 5.15: Fleet impact on increasing 50% fixed and variable costs

Finally, the impact of increasing by 50% the costs is also very visible in the recommended fleet policy. Figure 5.15 shows how the ADP policy tends to reduce the fleet size for the majority scenarios, as a measure to reduce the negative effects of an increase in costs for the same levels of demand. In terms of notation, the scenarios are labelled by their own stochastic process of demand growth. Being I the node representing initial conditions, IMMM is equivalent to I_1.05_1.05_1.05 since 0.05 is the corresponding demand growth for outcome M. Likewise, ILLL would be I_0.95_0.95_0.95 and IHHH would be I_1.15_1.15_1.15, since the respective demand growth of outcomes L and H is -0.05 and and 0.15 respectively.

Impact of varying disposal costs

Changes in disposal costs are not affected by the initial set of approximated observations feeding the kernel regression. This is due to the fact that the initial approximated observations are taken from $\overline{V}_t^0(S_t) = \lambda(T-t)\widetilde{C}_t^*(S_t)$, where disposal costs do not intervene in the optimal values of the contribution function $\widetilde{C}_t^*$. Therefore, the influence of the disposal costs is directly transmitted by the Algorithm 2 presented in the methodology chapter. This allows to vary the disposal costs as much as desired and independently of the ADP parameters and initial observations.

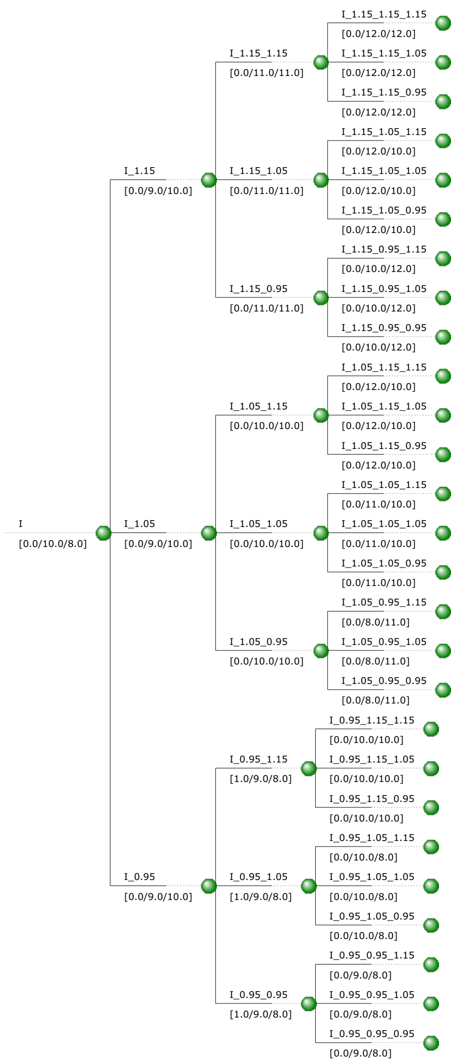


Figure 5.16: Fleet policy with $c_t^{disp} = 0$

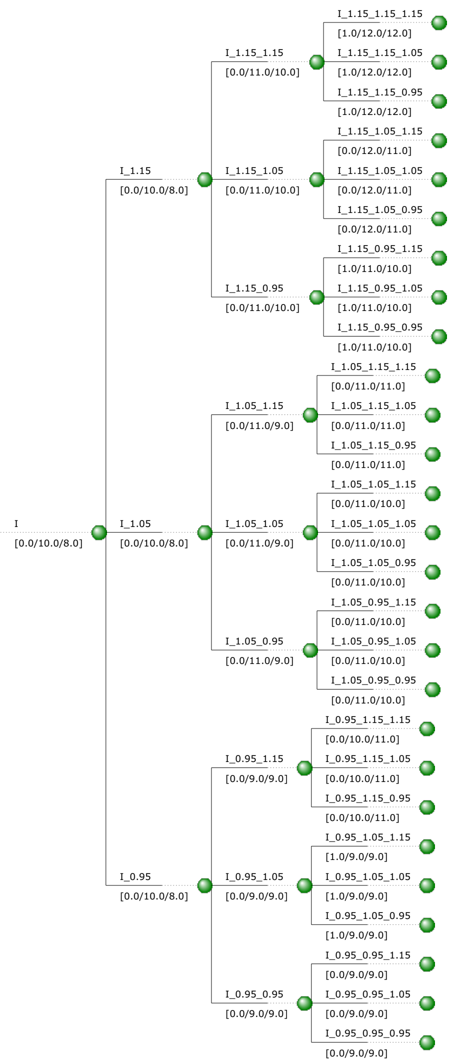


Figure 5.17: Fleet policy with $c_t^{disp} = 100000$

To verify the correct performance of the ADP algorithm, the disposal costs were varied from \$0 to \$1000000. In this context, the most intuitive aspect to assess is how an increase in disposal costs impacts on the recommended fleet policy. Figures 5.16 and 5.17 show the expected behavior: low disposal costs will lead to more flexible policies in which several fleet changes are easily allowed between stages. In contrast, high disposal penalizations will translate into more uniform fleet policies, where aircraft are less likely to be disposed in high quantities.

Apart from that, it must be highlighted that higher disposal costs will lead to a more difficult convergence: the search for the best policy gets harder for the ADP algorithm since it has to perform a complex trade-off between approaching to the optimal fleets and facing higher disposal costs due to the corresponding fleet modifications.

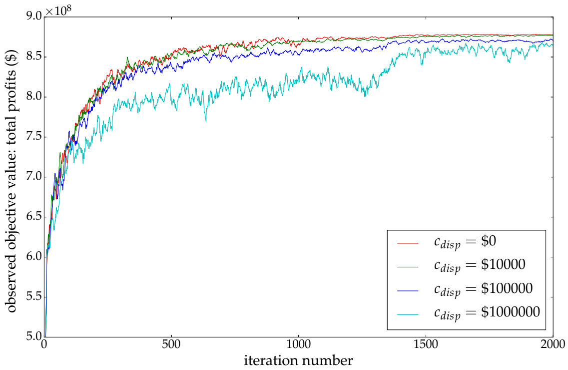


Figure 5.18: ADP response for different disposal costs

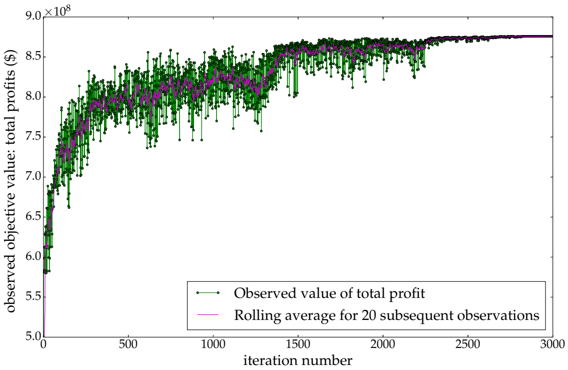


Figure 5.19: Exacerbated noise amongst value function observations due to an increase in disposal cost of $1000000

Impact of varying demand growth

The ADP response to varying demand growth can directly be extracted from previous Table 5.7. By plotting the profits earned per scenario in function of their corresponding demand variation, it is verified then that for scenarios with higher demand growth, more passengers are carried, thus making more profit in general.

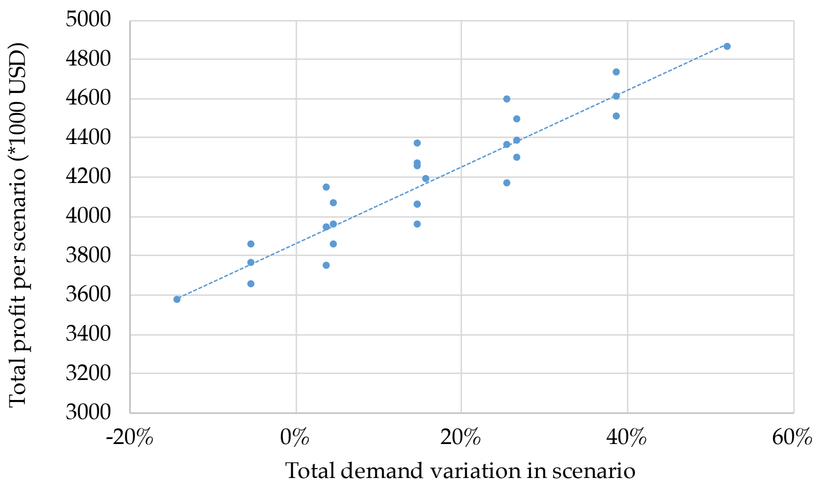


Figure 5.20: Profit growth in function of total demand variation per scenario

This general tendency is depicted in Figure 5.20, where the effect of capturing path dependency can also be appreciated amongst scenarios sharing similar demand growths. As previously commented, different scenarios can attain the same total demand variation by experiencing different stochastic growth processes $\omega_{[t]}$. This leads to the choice of different fleet compositions, since each scenario has been constrained by its own probability of happening as well as the node interdependencies existing with other scenarios branching out from common nodes. On this basis, it can be concluded that the ADP algorithm behaves logically and as intended.

5.3.4. Sensitivity analysis and calibration

In the methodology chapter, the entire ADP algorithm was described together with the introduction of its most important parameters. In particular, it was mentioned the high importance of calibrating well the following parameters:

- Control parameter to manage the impact of the initial approximated observations, λ
- Stepsize rule defining the value function learning rate, α
- Gain factor controlling the radius of Gauss kernels, G

Since the impact of the gain factor G was already analyzed and calibrated during the methodology chapter, the goal of this sensitivity analysis is to understand better how the other two most important ADP parameters influence the behavior of the algorithm and how this impacts the quality of the results. Based on this knowledge, the parameters λ and α_{n-1} will be calibrated so as to achieve optimal behavior for the algorithm.

Influence of λ

The parameter λ allows to control the impact of the initial approximated observations and its values range between:

$$\lambda \in \begin{cases} [0,1], & \text{if} \quad \widetilde{C}_t^*(S_t) \geq 0 \\ [1,T], & \text{if} \quad \widetilde{C}_t^*(S_t) < 0 \end{cases} \tag{4.53}$$

As introduced in the deterministic experiments in Section 5.2, the reason for its existence is its capability of avoiding the negative effects entailed by the presence of initial overestimations of the value function. It should be reminded that a situation in which the majority of states are overestimated will trigger the adverse loop of continuously visiting the most promising candidates, which will in turn fall from the best to least promising positions. Due to this fact, convergence is not guaranteed within an admissible computational time and thus, the obtained policies will most probably lack in coherence and quality. It must be noted that the interval $[0,1]$ is intended to underestimate positive values of the value function, while the interval $[1,T]$ does the same for negative values.

With the aim of understanding better the influence of λ and its impact on ADP results, several experiments were run for different values of λ. Figure 5.21 summarizes the ADP behavior observed when varying λ. As all observed $\widetilde{C}_t^*$ are positive, the sensitivity analysis focuses on the range $[0,1]$.

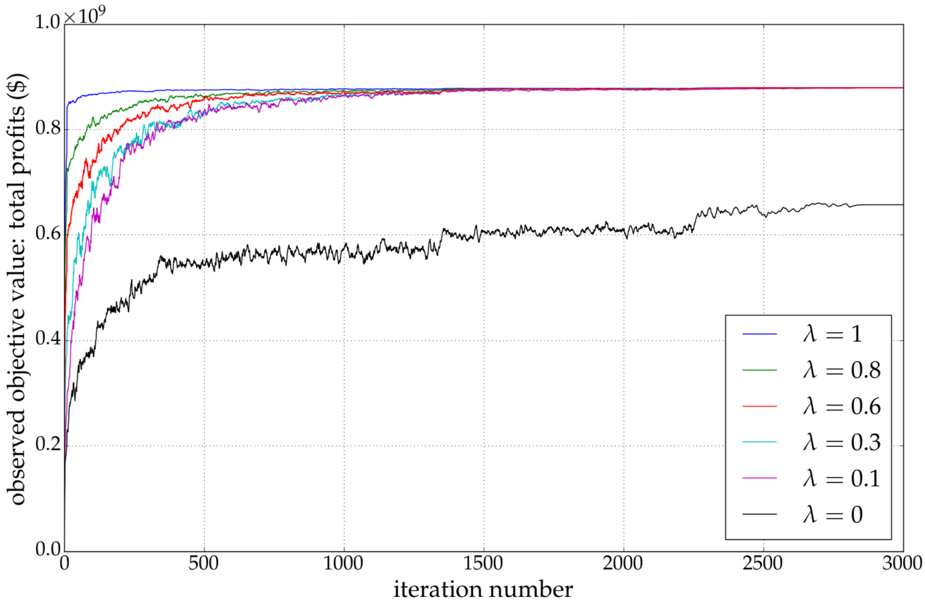


Figure 5.21: Impact of varying λ on ADP algorithm behaviour

Given these ADP responses, one could appreciate certain similarities between the behavior visible in Figure 5.21 and the one already seen in 5.14. Even if both situations are different, once again we can see that the more we underestimate the value of the initial approximation (in this case underestimation is done by decreasing λ and not decreasing the cost parameters), then the longer it takes for the algorithm to converge. Indeed, the value function starts to oscillate more for lower λ values ($\lambda \leq 0.6$), but its approximation keeps improving to the extent that in some occasions, it surpasses the value function observations made with higher λ values ($\lambda > 0.6$). On this basis, Table 5.8 provides a performance comparison between the different values of λ tested.

Table 5.8: Performance of ADP results for different values of λ - OF operating profits expressed in USD on a weekly basis

	Deterministic	Most-likely solution		$\lambda = 0$		$\lambda = 0.1$		$\lambda = 0.3$		$\lambda = 0.6$		$\lambda = 0.8$		$\lambda = 1$		$\lambda = 1.2$	
Scenario	OF	Dif.Best	OF	Dif.Best	OF	Dif.Best	OF	Dif.Best	OF	Dif.Best	OF	Dif.Best	OF	Dif.Best	OF	Dif.Best	OF
IHHH	4858543	-1.65%	4778148	-15.18%	4120869	-1.12%	4804080	-1.18%	4801077	-0.89%	4815384	-1.11%	4804382	-0.89%	4815384	-2.33%	4745547
IHHM	4733020	-1.21%	4675713	-17.40%	3909515	-0.72%	4699111	-0.85%	4692921	-0.56%	4706454	-0.69%	4700176	-0.56%	4706454	-1.53%	4660721
IHHL	4596565	-0.89%	4555486	-19.84%	3684623	-0.57%	4570269	-0.71%	4563814	-0.38%	4579012	-0.46%	4575564	-0.38%	4579012	-1.44%	4530544
IHMH	4616520	-0.90%	4575030	-18.67%	3754726	-0.58%	4589861	-0.71%	4583779	-0.46%	4595088	-0.46%	4595088	-0.46%	4595088	-1.91%	4528524
IHMM	4489211	-0.52%	4465826	-20.83%	3554320	-0.30%	4475805	-0.43%	4469886	-0.20%	4480146	-0.20%	4480146	-0.20%	4480146	-1.10%	4439867
IHML	4371340	-0.49%	4350090	-23.72%	3334619	-0.38%	4354541	-0.62%	4344027	-0.49%	4350065	-0.49%	4350065	-0.49%	4350065	-2.27%	4272162
IHLH	4372230	-0.33%	4357830	-24.79%	3288296	-0.42%	4353691	-0.52%	4349331	-0.46%	4352260	-0.57%	4347318	-0.62%	4345267	-2.32%	4270893
IHLM	4264448	-0.27%	4253117	-27.47%	3092846	-0.27%	4253099	-0.39%	4247636	-0.40%	4247546	-0.44%	4245624	-0.46%	4245013	-2.55%	4155757
IHLL	4150656	-0.99%	4109537	-29.93%	2908546	-0.79%	4117921	-0.72%	4120604	-1.12%	4103967	-0.77%	4118592	-0.52%	4128866	-3.31%	4013079
IMHH	4512719	-0.64%	4483846 .	-17.33%	3730593	-0.44%	4492846	-0.51%	4489906	-0.46%	4491971	-0.58%	4486353	-0.46%	4491971	-3.51%	4354331
IMHM	4390457	-0.36%	4374641	-19.60%	3529999	-0.29%	4377904	-0.43%	4371783	-0.31%	4377028	-0.34%	4375320	-0.31%	4377028	-2.88%	4263948
IMHL	4268208	-0.22%	4258906	-22.44%	3310298	-0.48%	4247823	-0.75%	4236304	-0.50%	4246948	-0.31%	4254931	-0.50%	4246948	-2.75%	4150646
IMMH	4294605	-0.14%	4288516	-24.38%	3247381	-0.19%	4286410	-0.35%	4279458	-0.27%	4283176	-0.27%	4283176	-0.27%	4283176	-1.03%	4250519
IMMM	4185912	0.00%	4185912	-27.13%	3050261	-0.03%	4184730	-0.17%	4178825	-0.08%	4182543	-0.08%	4182543	-0.08%	4182543	-1.77%	4112017
IMML	4067694	-0.68%	4039959	-29.74%	2857762	-0.57%	4044424	-0.38%	4052411	-0.28%	4056129	-0.28%	4056129	-0.28%	4056129	-3.02%	3944862
IMLH	4064153	-0.54%	4042312	-28.21%	2917524	-0.63%	4038648	-0.45%	4045905	-0.46%	4045551	-0.57%	4040814	-0.72%	4035000	-2.39%	3966840
IMLM	3961840	-1.32%	3909490	-30.65%	2747606	-0.69%	3934598	-0.42%	3945017	-0.36%	3947527	-0.41%	3945554	-0.53%	3940828	-1.93%	3885543
IMLL	3859255	-2.49%	3763271	-33.23%	2576706	-1.54%	3799909	-1.04%	3819011	-0.73%	3831189	-0.39%	3844064	-0.36%	3845352	-2.96%	3745128
ILHH	4171585	-0.48%	4151630	-28.16%	2996786	-0.73%	4141109	-0.35%	4157063	-0.82%	4137477	-0.82%	4137584	-0.62%	4145886	-1.62%	4104152
ILHM	4062362	-0.38%	4046916	-31.04%	2801336	-0.45%	4044262	-0.20%	4054090	-0.53%	4040630	-0.53%	4040827	-0.42%	4045295	-1.78%	3990178
ILHL	3950261	-1.19%	3903337	-33.75%	2617035	-0.36%	3936139	-0.82%	3917797	-0.45%	3932507	-0.54%	3929088	-1.02%	3910116	-1.55%	3888874
ILMH	3964795	-0.95%	3927296	-33.88%	2621430	-0.59%	3941419	-0.26%	3954407	-0.43%	3947860	-0.59%	3941551	-0.77%	3934228	-2.79%	3854092
ILMM	3862483	-1.76%	3794475	-36.55%	2450857	-0.44%	3845478	-0.16%	3856383	-0.40%	3846972	-0.42%	3846291	-0.50%	3843279	-2.18%	3778359
ILML	3758912	-2.94%	3648255	-39.33%	2280611	-0.57%	3737605	-0.50%	3740045	-1.01%	3720966	-0.38%	3744802	-0.24%	3749839	-2.27%	3673709
ILLH	3747829	-2.99%	3635870	-36.05%	2396632	-0.92%	3713232	-0.79%	3718404	-0.58%	3726117	-0.65%	3723353	-0.98%	3711221	-1.32%	3698205
ILLM	3654140	-4.21%	3500408	-38.70%	2240115	-1.73%	3591074	-0.60%	3632235	-0.39%	3639948	-0.43%	3638375	-0.72%	3627904	-1.08%	3614594
ILLL	3570850	-5.75%	3365645	-41.59%	2085624	-3.07%	3461277	-0.97%	3536098	-0.76%	3543811	-0.57%	3550596	-1.84%	3505283	-0.87%	3539819
Expected value	4243477	-0.82%	4208652	-25.57%	3158499	-0.50%	4222752	-0.47%	4223186	-0.40%	4226538	-0.41%	4226011	-0.42%	4225545	-2.07%	4155786

Just for the sake of clarity, the ADP results for $\lambda = 1.2$ are included so as to provide a clearer understanding of the effects of overestimating the value function approximation. However, λ is not intended to take higher values than 1, thereby representing a percentage controlling the impact of the initial value function approximation. From the performance data shown in Table 5.8, it can be drawn that both too low ($\lambda < 0.1$) and too high values of λ ($\lambda > 1$) worsen completely the ADP performance to the extent of expecting a general score as low as -26% for $\lambda = 0$ and -2.07% for $\lambda = 1.2$. Having introduced these scores, it is well worth to remind the reader about the paramount importance of providing *meaningful* observations, which preserve the essential information of the problem. Indeed, setting λ equal to 0 cancels completely the transmission of the problem structure and thus, the algorithm lacks any guidance when it comes to visit potential states: actions are definitely chosen at random. In light of this, it does not come as a surprise that $\lambda = 0$ performs very poorly across all scenarios. In contrast, the situation of $\lambda = 1.2$ is slightly different: since it enhances overestimation across all states, the continuous search of the most promising states makes them be revalued and fall down into the least promising categories. In this way, the algorithm will most likely keep on looking for other overestimated values which apparently look more promising than the ones already visited. Eventually, this adverse loop will wreck all problem structure previously transmitted to the kernel regression.

Evidently, both results go beyond the targeted order of magnitude set by the most-likely solution, which is expected to perform at a -0.82% difference with respect to the deterministic upper reference. Since the reason for solving wide scenario trees is indeed the improvement of current airline best practices, the ADP algorithm will only prove to be a better method in those λ intervals where its performance score surpasses the one of the most-likely fleet solution. Therefore, we are only interested in Dif.Best scores greater or equal to -0.82%, which happen to be found in Table 5.8 between the interval $0.1 \leq \lambda \leq 1$. Although all these ADP responses surpass clearly the most-likely solution, they still present certain differences in terms of performance. On the one hand, significant reductions in λ ($\lambda < 0.6$) will lead to a general deterioration in profits across all scenarios. This is a direct consequence of underestimating the value function approximation since in doing so, the algorithm takes more time to converge and reach better policies. On the other hand, greater values ($\lambda > 0.8$) will entail the presence of some scenarios being overestimated, thus penalizing too much the overall good performance of the policy across the rest of scenarios. From these observed trends, it follows that a trade-off must be done to calibrate correctly the value of λ. For this purpose, it is insightful to perform a spline interpolation between the expected performance scores found in the last row of Table 5.8. For the sake of clarity, the values $\lambda = 0$ and $\lambda = 1.2$ are not accounted due to the abrupt discontinuity they entail and their lack of interest for the calibration of λ.

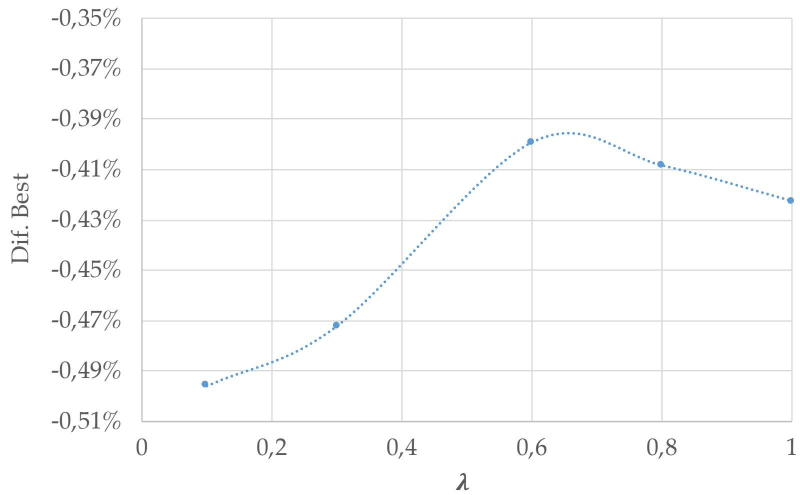


Figure 5.22: Unstable value function correspondence at stage 5 with G = 1000

From Figure 5.22 it can be concluded that optimal ADP performances will be found between the interval $0.6 \leq \lambda \leq 0.8$, being more close to 0.6 than 0.8. Thus, λ can either be calibrated to 0.6 or 0.7 since both values are expected to provide the best ADP results possible. By extrapolating this results to a case where $\widetilde{C}_t^*$ is negative, the calibrated interval would be found in between $1.2 \leq \lambda \leq 1.4$.

Influence of α

As discussed in Chapter 4, the stepsize α_{n-1} indicates the rate at which the value function is updated throughout the iterations:

$$\overline{V}_t^n(S_t^n) = (1-\alpha_{n-1})\overline{V}_t^{n-1}(S_t^n) + \alpha_{n-1}\hat{v}_t^n \tag{4.87}$$

The stepsize values range between [0,1]. The closer α_{n-1} is to 1, the faster the value function approximation will be updated to the current observation $\hat{v}_t^n$. In the same way, the lower is its value, then the more slowly the value function will be updated. In other words, the stepsize can be understood as a parameter to control the updating response of the algorithm: Lower stepsize values will provide more stability and thus, will slow down the response of the algorithm. This is due to the fact that the algorithm will take more iterations to update the value function estimations. As previously commented, higher values of α_{n-1} are recommended in an early exploration phase since they allow for faster value function learning rates. Likewise, lower α_{n-1} values are preferably chosen to enhance the algorithm stability during exploitation phases, where the best candidate policies start to be more defined.

With regard to all these considerations, several simulations are run so as to assess the response of the ADP algorithm to several stepsize rules, which are depicted in Figure 5.23. By just looking at the form of the different learning rules, it can be expected that stepsizes decreasing too fast throughout the iterations (e.g.

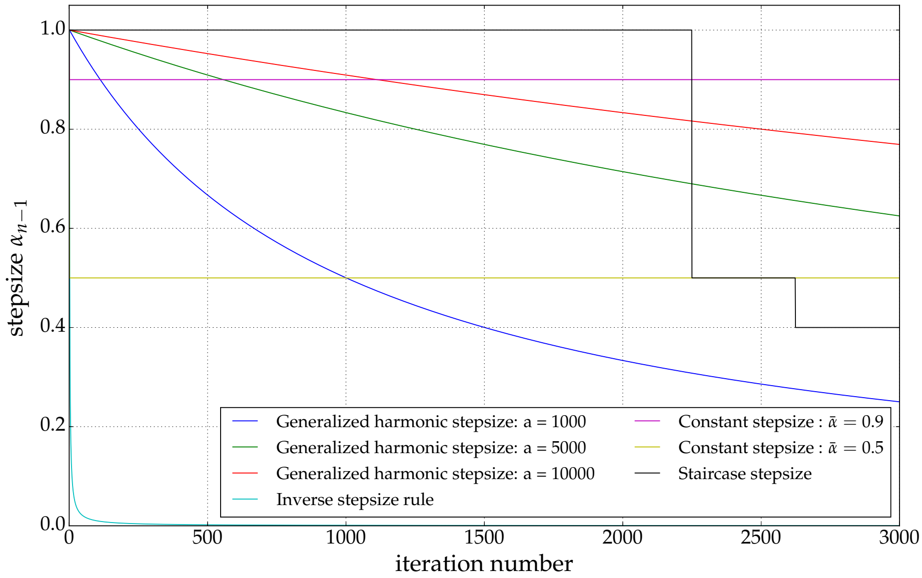


Figure 5.23: Stepsize evolution according to different learning rates schedules

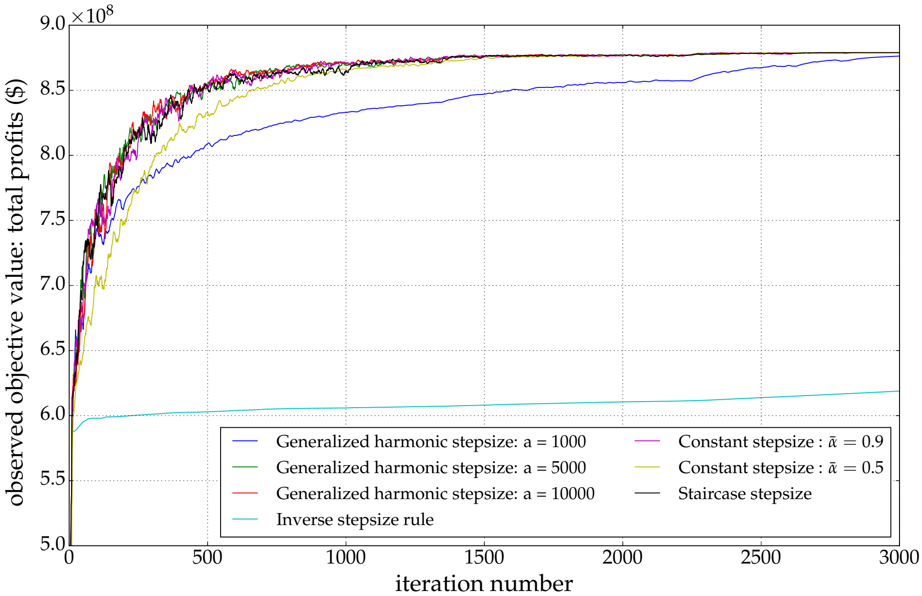


Figure 5.24: ADP responses to different learning rates schedules

inverse step) will lead to the stalling of value function updates: after the first iterations, the value function estimates will not improve. In the same way, those rules maintaining significant values of α_{n-1} for the later stages (e.g. generalized harmonic stepsizes, staircase stepsize), will continue to help improve the subsequent observations of the value function. This behavior is appreciated in Figure 5.24: in the inverse stepsize the iterative improvement of the value function decays completely after few iterations. Meanwhile, it remains steady for those stepsize rules that eventually preserve the highest stepsize values for a longer time such as the generalized harmonic stepsizes for $a = 10000$ and $a = 5000$, the constant stepsize rule for $\alpha_{n-1} = 0.9$ and the staircase stepsize. Furthermore, it is visible how lower initial values like $\bar{\alpha}_{n-1} = 0.5$ present a less steep learning curve than the ones with initial higher values. Overall, the learning rate that has shown best results is the generalized harmonic stepsize for $a = 5000$. Nevertheless, it must be noted that all stepsize rules maintaining high values of α_{n-1} present much better results than the ones with lower values $\bar{\alpha}_{n-1} = 0.5$ or decaying too fast (e.g. inverse stepsize and generalized harmonic stepsize for $a = 1000$) .

This general preference for higher values of α_{n-1} can be explained by the characteristics of the approximation strategy used combined with the way in which the exploration vs exploitation problem is solved. When it comes to updating the value function, kernel regressions resemble more lookup tables in its functioning than parametric regressions. Indeed, in a kernel regression the value of being in a state is updated at a local level. Even if this new observation will influence the estimations in other states, its impact will be much less noticeable than the one transmitted by updating the parameters of a global parametric approximation. While this prevents the algorithm from updating much faster the value function, this brings the advantage of obtaining more reliable approximations at a local level. This is the main reason why maintaining high stepsize values is beneficial for our ADP algorithm. Additionally, it must be highlighted that the ADP algorithm features a SA controller to handle more efficiently the epsilon-greedy approach when choosing decisions at random. This also enhances the preference for higher values of α_{n-1}: it is rather improbable that the value function approximation suffers too high instabilities by the exploration of new values, which is basically what lower values of α_{n-1} are intended to mitigate. Therefore, there is no real need to decrease the value of α_{n-1}. Indeed, it can be assumed that the following simplification of the value function update may result effective:

$$\overline{V}_t^n(S_t) = \begin{cases} \hat{v}_t^n, & \text{if} \quad S_t = S_t^n \\ \overline{V}_t^{n-1}(S_t), & \text{otherwise} \end{cases} \tag{5.1}$$

5.3.5. Monte Carlo sampling of a normal distribution

This section provides an overview of the major differences found when sampling $\omega_{[t]}$ with two different probability functions: a discrete and a normal probability distribution with $\mu = 5\%$ and $\sigma = 5\%$. The rest of parameters remain the same with respect to the stochastic baseline case with discrete outcomes. In this context, Figure 5.25 shows the behaviour of the ADP algorithm for both methods.

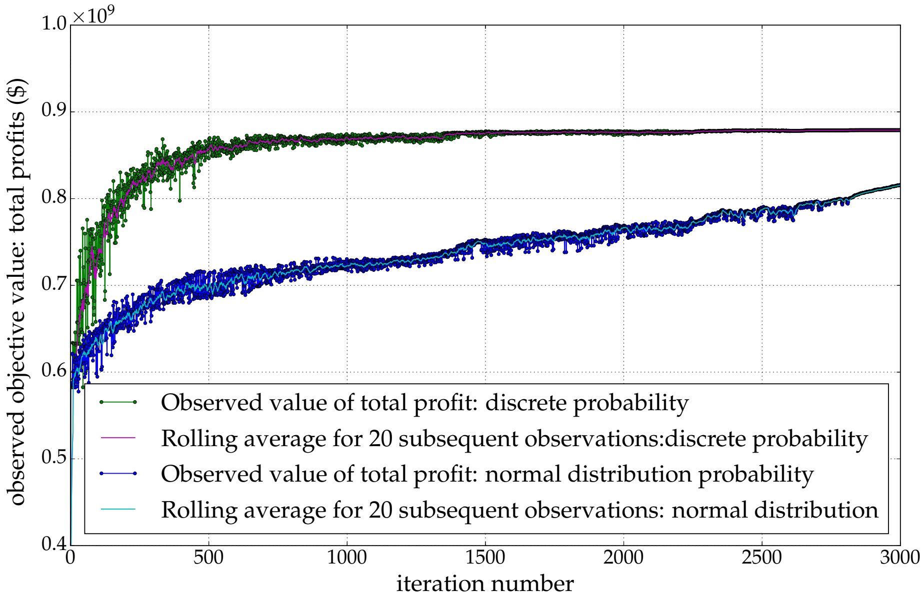


Figure 5.25: Performance comparison between discrete sampling and continuous sampling

The trends observed confirm the discussion hold in previous Section 4.4.2: assuming ω_t as a continuous random variable leads to the infinite branching of nodes across the scenario tree. This lack of restriction on generating new scenarios hinders the convergence speed of the algorithm. For every iteration, a new scenario is generated, thereby increasing exponentially the state space of solutions. In light of this, the frequencies of visiting again the same scenarios or even the same states are substantially reduced. Thus, the ADP algorithm faces more hurdles to learn better approximations of the value function. Furthermore, the continuous generation of new branches implies a continuous learning which avoids the possibility to stabilize at some point.

All these translates into slow convergence and algorithm instability. Nevertheless, it should not be overlooked that sampling a continuous distribution provides much more realistic scenarios, embracing a wide range of possible outcomes. Due to its unmanageable dimensions, Appendix C shows some details of the continuous scenario tree obtained with normal distribution sampling.

5.3.6. Conclusions of stochastic experiments

Based on the full set of stochastic experiments, the following points can be concluded:

- Modelling stochastic processes with a scenario tree introduces a coupling relationship amongst all scenarios with common nodes: value function estimates are continuously updated with random observations coming across all periods and states. This fact leads to an increase in noise levels, thereby complicating the convergence of the ADP algorithm. Whereas only 80 iterations were required to achieve optimal or near-optimal solutions for the deterministic case, the stochastic version of the problem needs around 3000 iterations to reach convergence and provide a coherent policy.

- In terms of computational performance, running 3000 iterations translates in a CPU time cost of 5h, without considering the 2h punctually invested in creating an initial set of observations to train the kernel regression. Thus, the sharp increase in computational time with respect to the deterministic case is evident and must definitely be improved in case of tackling real-sized problems.

- Solving the stochastic version of the multi-period fleet planning problem does not lead to a single result, but to an adaptive fleet policy: the best rule for making fleet and operational decisions in function of a certain demand and financial conditions happening in a given time period. In this context, it can be concluded that the resulting fleet policy adjusts as much as possible to the optimal fleet plans corresponding to each deterministic scenario solved independently. A very unique feature of the adaptive fleet policy is its capability of capturing the path dependency effect amongst different scenarios. Therefore, the adaptive policy simulates the decision-making process of fleet planning in a very realistic way.

- When it comes to profitability performance, the adaptive policy is the most robust method for fleet planning: across the majority of scenarios it clearly excels the performance of the optimal fleet plan for the most-likely scenario. Evidently, this most-likely solution performs better in those scenarios similar to the most-likely scenario. Nevertheless, its performance starts to stall towards differing scenarios. Indeed, its worst scores are found in the neighborhood of the extreme scenarios: For the most pessimistic scenarios the most-likely solution can imply weekly operating losses ranging between 3% and 6%, whereas in the most optimistic scenarios losses remain between 2% and 1%. In contrast, the adaptive fleet policy mitigates losses in all extreme scenarios, which never surpass the 1.1%. By analyzing the general expected performance, it can be concluded that the adaptive policy reduces by 50 % the losses entailed by the most-likely solution.

- Apart from this, the verification analysis proves the stable behaviour of the ADP algorithm. By modifying the parameters of the baseline case within different intervals, it is concluded that the kernel approximation strategy can still be trained reasonable well using the same dataset of observations. Furthermore, a sensitivity analysis helps understand better the influence of the most relevant parameters within ADP algorithm. On the one hand, higher stepsize values ($\alpha_{n-1} \approx 1$) provide the best results for value iteration. On the other hand, the correct calibration of the control parameter λ is essential to guarantee the convergence of the ADP algorithm as well as the coherence of the obtained results. Indeed, the problem of value function overestimations represent a major challenge to tackle in follow-up research.

- Lastly, the application of a continuous random variable normally distributed has been benchmarked against the performance of discrete random variables. This experiment has led to the statement that the use of continuous random variables entails slow convergence and higher instability for the algorithm. In any case, it should not be overlooked that sampling a continuous distribution provides much more realistic scenarios, thus embracing a wide range of possible outcomes. Nevertheless, the size of the scenario tree becomes unmanageable.

All in all, this set of conclusions confirm the advantage of applying adaptive fleet policies and most importantly, proves approximate dynamic programming as an effective method to calculate them.

6

Algorithm CPU performance enhancement

In previous Chapter 5 it was concluded that for a experimental problem setting of 20 routes and 3 aircraft types, the ADP algorithm required a minimum of around 1500 iterations to reach convergence. However, much better values of the objective function and a more coherent ADP policy were achieved for 3000 iterations, which entailed a CPU time cost of 17804s ($\approx$ 5h). Indeed, this represents a very sharp increase in computational time with respect to the 2 minutes required in a deterministic setting, where tight optimality gaps of 0.2% and optimal solutions could be achieved easily. As for the initial dataset to train the kernel regression, its punctual calculation required 2 hours regardless of whether the problem was stochastic or deterministic.

Evidently, the ADP algorithm as it stands right now would most likely face several time constraints if more realistic sizes of the multi-period fleet planning problem under demand uncertainty were required to be solved. Since the final aim of this MSc thesis is to assess the performance of the ADP support tool in a real working environment, this issue becomes a crucial matter to be particularly analyzed and improved in this chapter. To tackle this problem, a decision tree analysis is initially carried out in Section 6.1 to pinpoint the main sources of CPU time loss and implement measures to mitigate them. Once implemented in the code, Section 6.2 discusses the results of the algorithm performance enhancement.

6.1. Algorithm performance analysis

Figure 6.1 illustrates the structure of this decision tree as well as the several levers that have been implemented in the code.

The analysis starts from the core objective already mentioned: increasing the code efficiency to tackle problems of more realistic dimensions. To do so, it is well worth to identify where the code spends the most part of the computational time. The most time-consuming tasks are estimated for the stochastic baseline case previously solved. These are expressed as an indicative percentage for reference:

- **The initialization of the value function approximation** is estimated to spend 30% of the total CPU time (ADP pure loop plus generation of training dataset) for the stochastic baseline problem, which means an average of 2 hours. This percentage is estimated to increase in function of the initial state space explored, whose dimensions are defined by the number of aircraft types considered as well as the number of tree nodes observed.
- **Solving Part A of the 1-stage FPP subproblem with the Gurobi module** is estimated to spend around 40% of the total CPU time ($\approx$ 3h). However, the more routes and aircraft types are considered, the more it will take to Gurobi to reach optimality and thus, the higher the time percentage will be.
- **Solving Part B of the 1-stage FPP subproblem with the subroutine module** is estimated to spend around 30% of the total CPU time ($\approx$ 2h). Nevertheless, it should be pointed out that this percentage increases with the number of observations performed. The code takes more time to analyze the estimated values of each state assigned to every tree node observed and thus, the subroutine slows down considerably in those problems featuring more than 3 aircraft types.

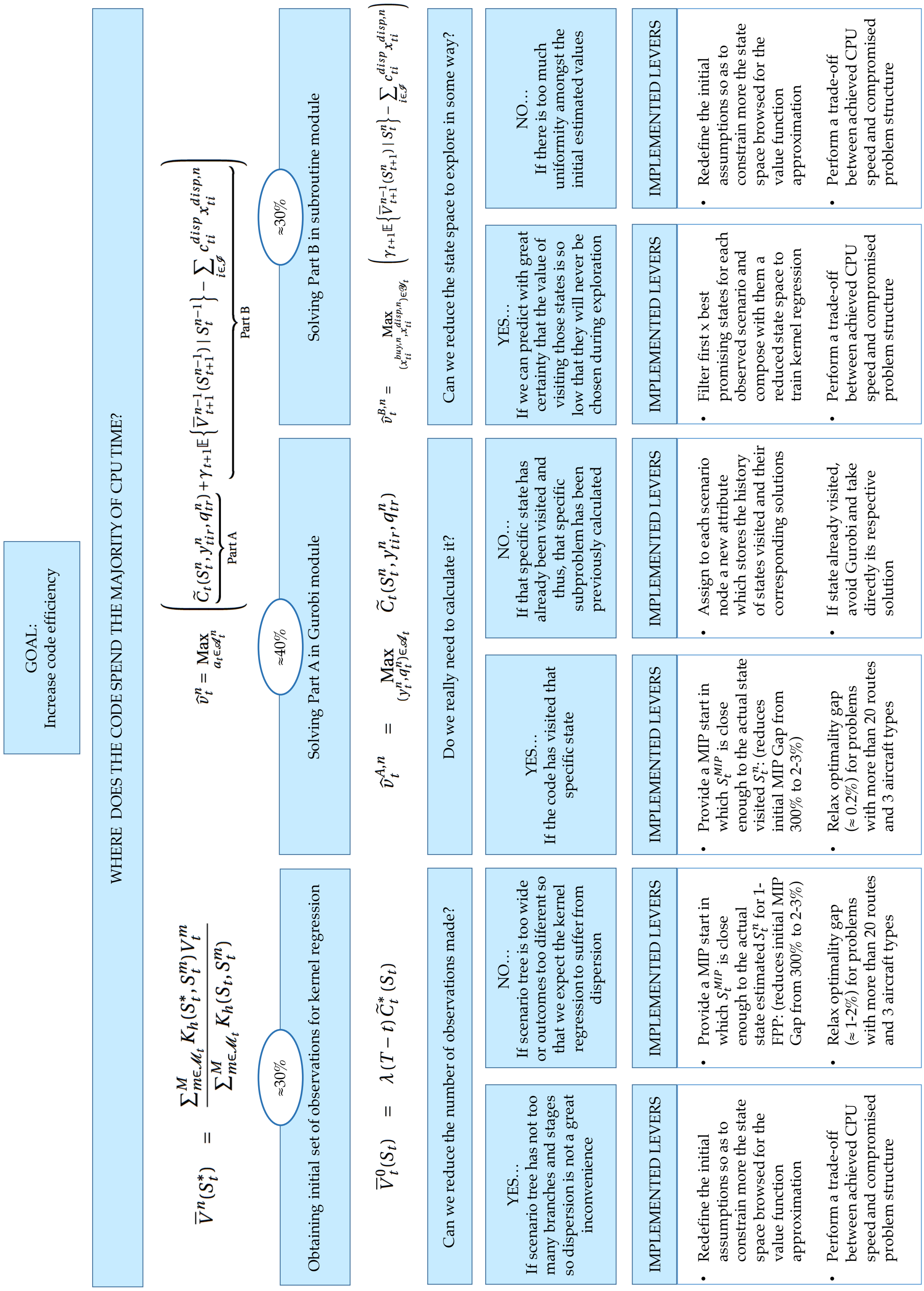


Figure 6.1: Decision tree analysis for increasing CPU efficiency

As schematized in the decision tree analysis, several levers have been implemented to reduce significantly the computational impact of the most time-consuming tasks. From the scheme, it can be inferred that the principal measures taken to boost the code efficiency are:

1. **Provide a MIP start** to Gurobi so as to reduce the initial MIP Gap of the optimisation of part A
2. **Relax significantly the optimality gap** ($\approx 1-2\%$) when generating the initial set of observations to feed the kernel regression
3. **Relax moderately the optimality gap** ($\approx 0.2-0.3\%$) when solving part A of the 1-FPP subproblem
4. **Avoid recalculations with Gurobi in part A** of states already visited
5. **Reduce the state space explored** so as to speed up subroutine in part B as well as the initialization of kernel regression

In any case, it is highly recommended to pay close attention to all the details presented in Figure 6.1, since this helps to have a better understanding of how each implemented lever reduces the computational time required by every task.

6.2. Results of the algorithm performance improvement

Having implemented all levers, it is time to assess their impact on time reduction by comparing the optimised algorithm with the baseline algorithm. For the sake of clarity, we refer to the baseline algorithm as the original algorithm without any implemented lever.

To begin with, it must be noted that by just relaxing the optimality gap to 2%, the initial 2 hours spent in initialising the value function approximation are reduced to 20 minutes. Indeed, this is a very powerful lever that allows for an 80% reduction of CPU time and thus, enhances the possibility of dealing with more aircraft types and larger fleet sizes. Nevertheless, it should be taken into account that this presents certain limits, given the fact that the required calculation times will increase exponentially with the number of aircraft types considered. Taking this improvement into account, the remainder of this section will discuss the improvements achieved for the actual ADP loop.

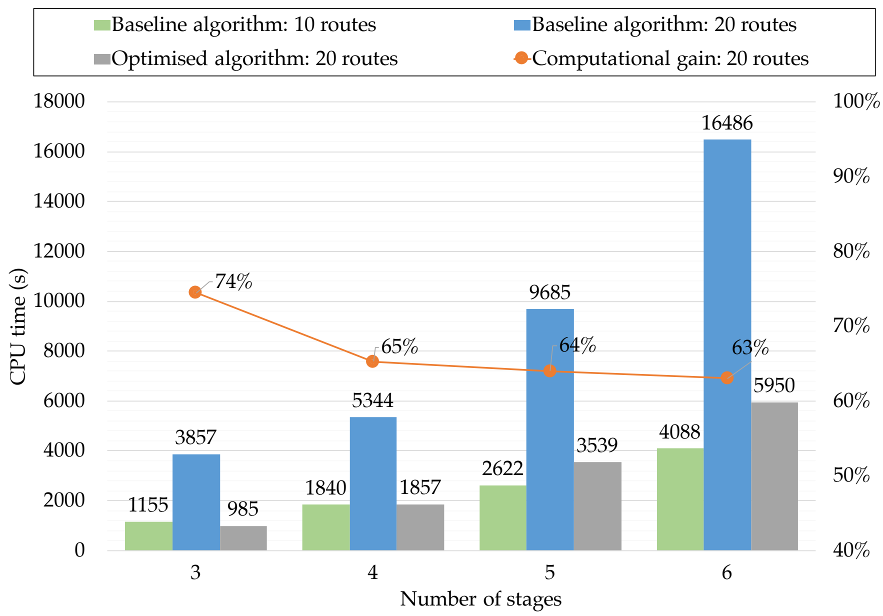


Figure 6.2: Increase in computational time in function of the number of stages for both baseline and optimized algorithm for N = 800

To begin with, it is insightful to compare the evolution of the required computational times when the number of stages is increased for both baseline and optimized algorithms with a fixed number of iterations. Figure 6.2 shows that for a very reduced problem of 10 routes, 3 aircraft types and 800 iterations the computational increase is moderate, whereas it becomes sharp when augmenting the number of routes. This visible trend stresses again the importance of enhancing the algorithm efficiency if problems of more realistic dimensions are targeted. The effect of the optimized algorithm is indicated by the grey bars. While a significant reduction of 70-60% can be appreciated across different stages, it is a bit surprising to see a deterioration in computational gain for higher stages.

To shed some light on this curious behavior, we can fix the number of stages and the problem dimensions while increasing the number of iterations. In this way, the trend of the computational gain can be assessed in Figures 6.3 and 6.4 below. Interestingly, it can be appreciated that for more stages the computational gain increases as well. Therefore, the optimized algorithm performs better when a higher number of iterations is run.

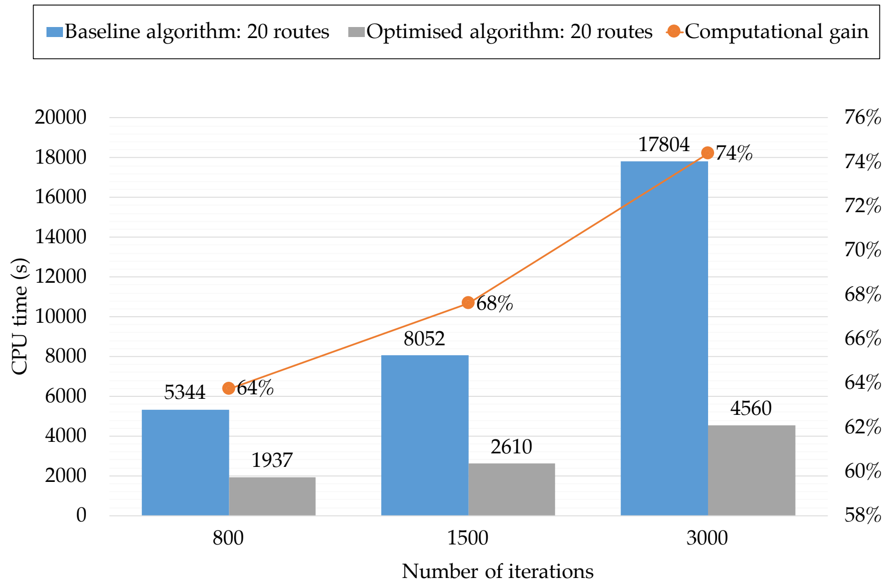


Figure 6.3: Performance of the optimized algorithm in function of the number of iterations for the problem with 20 routes, 3 aircraft types and 4 stages

This pattern is mainly due to the implemented lever of avoiding recalculations with Gurobi in part A: every node in the scenario tree is assigned a track record attribute of the already visited states as well as their corresponding $\widetilde{C}^*(S_t)$ values. In this way, the algorithm is prevented from wasting time in solving parts already calculated: whenever a state is revisited, then the algorithm takes its corresponding value previously calculated. The impact of this measure is clearly seen in Figure 6.5, where the number of iterations is represented against the required computational time to process them. In this context, it can be appreciated that the speed at which nodes are processed is not constant and varies in function of time and number of stages: for 800 iterations, the algorithm achieves greater velocity the lower the number of stages is. However, the most interesting aspect is found in the curves of 3 and 4 stages: the number of processed nodes per time increases gradually until reaching a maximum speed. Indeed, this behavior is explained by the fact that in early iterations the algorithm is exploring new states, thereby calling more times Gurobi. When a certain number of iterations has passed, the algorithm is less likely to fall into an unvisited state and consequently, will eventually invest less time optimising part A with Gurobi. More particularly, there is a moment in which the algorithm reaches maximum velocity by avoiding completely Gurobi and just operating with the data stored in the track records of tree nodes. In the same way, the more stages a scenario tree has, the more states there are available to visit. Given this fact, it is reasonable to see that for higher stage numbers, the algorithm will initially take more time to process as many iterations as in the case of 3 stages. Evidently, a 6-stage problem will have more scenarios and states to explore than a 3-stage case and thus, Gurobi will be called more

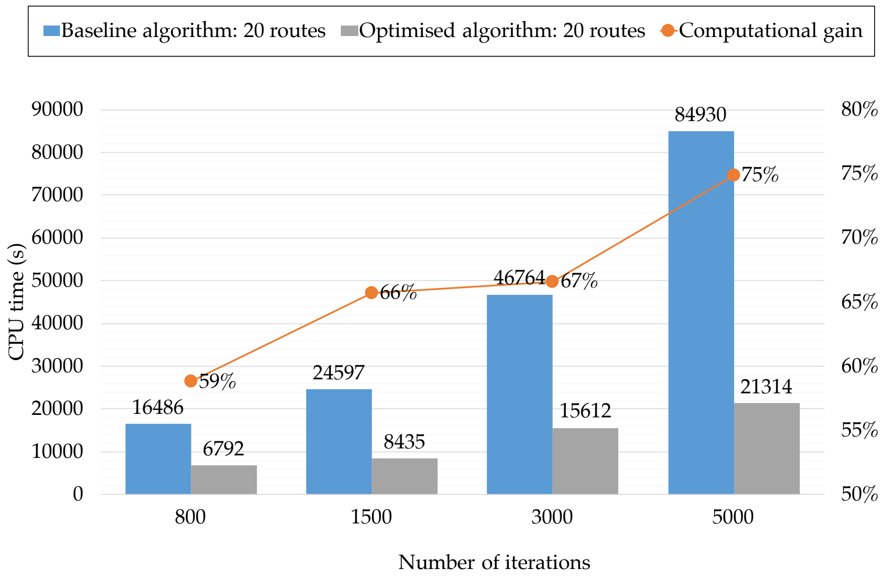


Figure 6.4: Performance of the optimized algorithm in function of the number of iterations for the problem with 20 routes, 3 aircraft types and 6 stages

times before the algorithm can use completely the node track records. This is why the algorithm performs better when increasing the number of iterations for problems of higher stages. For more iterations run, the algorithm is more likely to avoid recalculations and consequently, saves more computational time. Indeed, this behavior is seen in Figure 6.6, where the same 4-stage problem is solved for 3000 iterations and adopts a similar form to the 3-stage curve in Figure 6.5. The increased efficiency translates into a great computational saving of 74%, thereby reducing the computational time from 5h to 1h. Indeed, while the speed of the optimised algorithm undergoes a gradual augmentation, it remains basically constant for the baseline algorithm. In any case, it should be noted that the speed of the baseline algorithm improves moderately for the latest iterations. This is due to the last exploitation phase in which more similar solutions are recalculated, thereby reducing the starting MIP gap and saving some time. All in all, it can be stated that the fact of avoiding recalculations with Gurobi is the measure providing the biggest impact in terms of time reduction.

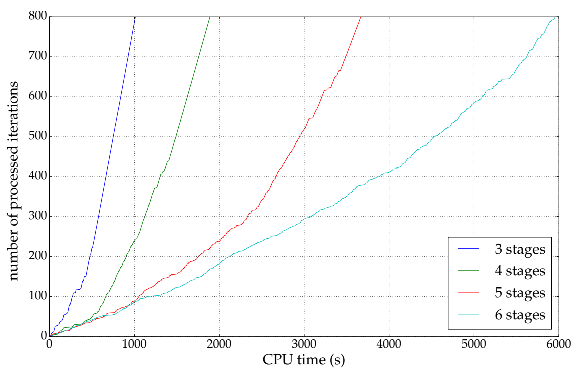


Figure 6.5: Processing speed of optimised algorithm in function of number stages with N= 800

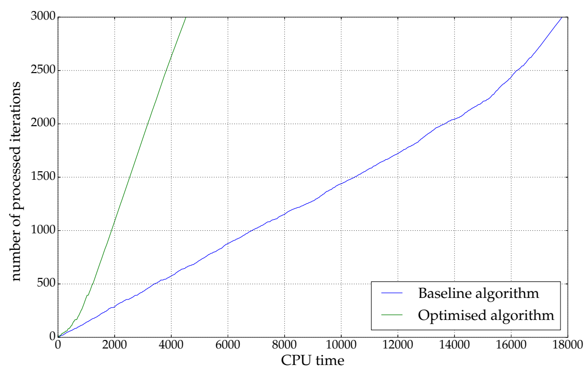


Figure 6.6: Comparison between processing speeds of the optimised and baseline algorithm when solving the 4-stage problem with 20 routes, 3 aircraft types and N = 3000

Since the main reason for implementing an ADP algorithm was to be able to cope with more realistic sizes of the multi-period fleet planning problem under demand uncertainty, the optimised performance of the algorithm will be assessed for an extended version of the baseline problem. This extended problem features the same 20 routes and aircraft types presented in chapter 5 but including 6 time stages. As can be observed in Figure 6.7, both optimised and baseline algorithms performances were tested for 800, 1500, 3000 and 5000 iterations. Once again it can be appreciated that the computational gain increases in function of the total number of iterations run. This trend is of great importance since high dimensional problems will entail the need of running more iterations to reach convergence and obtain results of good quality. By analysing the CPU times in Figure 6.7, it can be drawn that the algorithm optimisation was completely necessary to pave the way for solving more realistic problems: in order to solve the 6-stage problem, running 5000 iterations with the baseline algorithm would lead to a computational time of 22h, whereas the optimised algorithm would reduce it to 5h. Still, more complex problems are expected to require more than 5000 iterations. Finally, Figure 6.8 shows how running a too low amount of iterations may affect completely the convergence and the quality of results for problems of higher dimensions: while the non-convergent value function reaches an observed total profit of $\$1.3x10^8$, the convergent value function can attain a value of $\$1.4x10^8$.

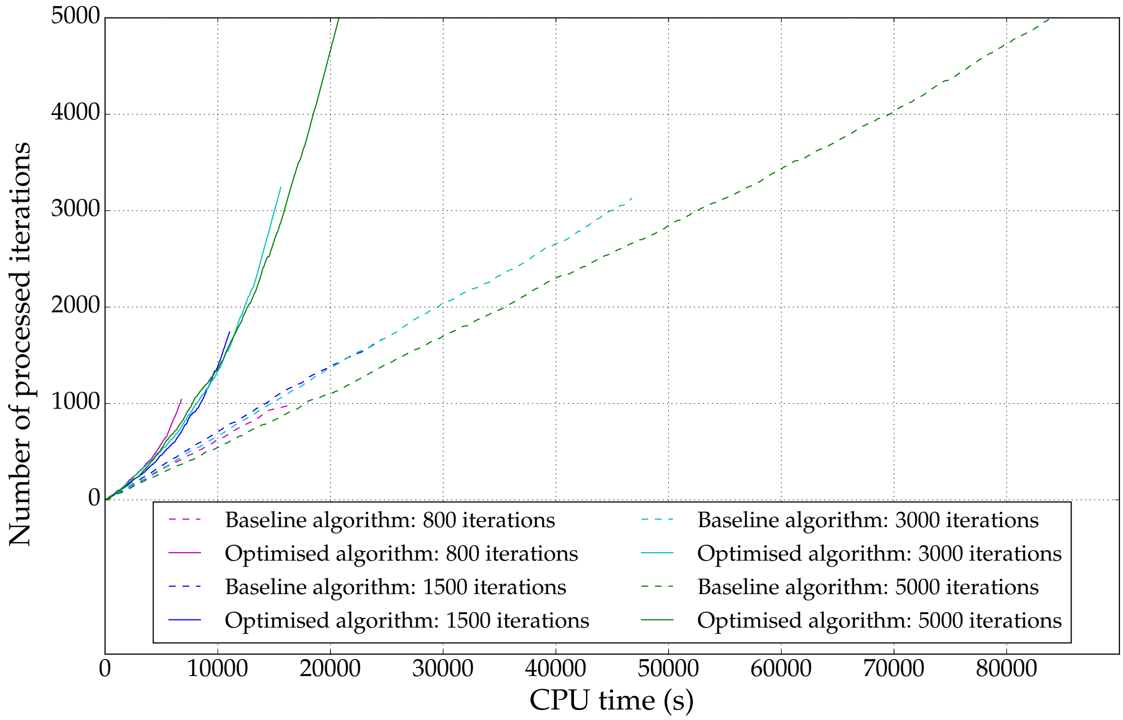


Figure 6.7: Comparison between processing speeds of the optimised and baseline algorithm when solving the 6-stage problem with 20 routes and 3 aircraft types for different number of iterations

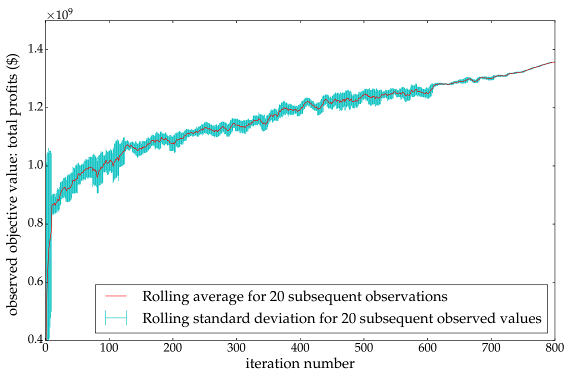


Figure 6.8: Non-convergent value function for 6-stage multi-period fleet planning problem with N=800

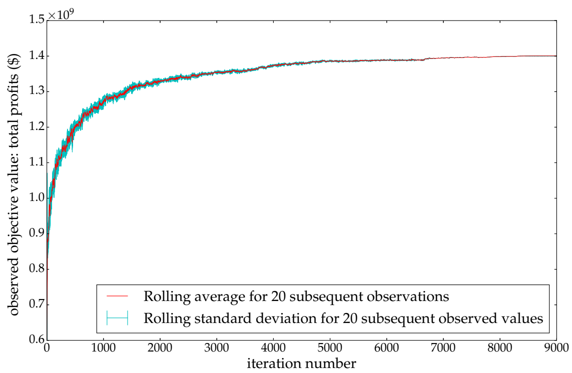


Figure 6.9: Convergent value function for 6-stage multi-period fleet planning problem with N=9000

6.3. Conclusions

The results obtained with the enhancement of the algorithm efficiency bring forth the following conclusions:

- The decision tree analysis has led to define 5 major levers to improve the algorithm performance. These consist in (1) providing a MIP start to Gurobi, (2) relaxing the optimality gaps when generating the training dataset and (3) solving part A of the 1-FPP problem, (4) avoiding unnecessary recalculations with Gurobi in part A of the 1-FPP problem as well as (5) reducing the state space explored to accelerate the subroutine in part B.
- When initialising the value function approximation of the stochastic baseline case, the relaxation of the optimality gap to 2% allows to reduce the initial 2 hours required to 20 minutes. This translates into an 80% reduction of CPU time and thus, enhances the possibility of dealing with more aircraft types and larger fleet sizes. However, this presents certain limits, given that the required calculation times increase exponentially with the number of aircraft types considered. On the other hand, this relaxation does not entail a significant loss in the accuracy of the adaptive policy.
- Regarding the ADP loop enhancement, the fact of avoiding recalculations with Gurobi is the lever that has the greatest impact. Indeed, it can be appreciated that the speed at which nodes are processed is not constant and accelerates throughout the iterations: during the exploration phase the algorithm spends more time calculating the value of states visited for the first time; however, the number of processed iterations increases when the algorithm starts to exploit states already visited and calculated. As far as the stochastic baseline case is concerned, the algorithm enhancement allows for a great computational saving of 74%, thereby reducing the computational time from 5h to 1h.
- The improvement of the algorithm performance was completely necessary tackle problems of more realistic dimensions. For instance, running 5000 iterations of the 6-stage fleet planning problem with the baseline algorithm would lead to a computational time of 22h, whereas the optimised algorithm would reduce it to 5h. Still, more complex problems are expected to require more than 5000 iterations.

7
Kenya Airways Case Study

Until now, several reduced problems have been used as a experimental setup to verify the correct performance of the ADP algorithm as well as to identify and mitigate its possible limitations. In these assessments, the ADP algorithm has proven capable of providing effective adaptive policies, which can definitely excel the profitability results expected from a deterministic fleet plan forecasted with a most-likely scenario. Despite the positive results obtained so far and the code optimisation made in Chapter 6, the key remaining question is whether the optimised ADP algorithm can tackle successfully a problem of realistic dimensions while still being capable of providing an effective adaptive policy.

In light of this, the objective of Chapter 7 is to perform a realistic case study based on a reference airline so that the potential of the developed ADP support tool is completely assessed. The general context and purpose of this case study will be described in Section 7.1, while Section 7.2 will present the model parameters used. Next, the ADP algorithm behaviour and adaptive fleet policy will be discussed in Section 7.3, and Section 7.4 will show how the obtained adaptive policy can be employed to extract useful signposts for fleet planers. Finally, a validation of the ADP results will be carried out in Section 7.5 by performing an expert survey and comparing the ADP results to real data from the reference airline.

7.1. Case study context

The proposed case study revolves around a former 5-year plan developed by the international carrier Kenya Airways (KQ) as part of its network expansion strategy dated from 2015. Within KQ strategy, a rollout plan for the opening of new routes was expected to be followed for the next 5 years. According to this plan, by the ends of 2014 Kenya Airways was considering to enlarge its passenger fleet to meet increases in future demand as well as to standardize it into 3 main aircraft types. In that regard, KQ corporate information described the following fleet development plan (Kenya Airways, 2015, 2016, 2017):

Table 7.1: Passenger fleet evolution from 2015 to 2017 and expected fleet plan for the upcoming years

Category	Operated Aircraft	2015	2016	2017	2018	2019
Wide body	Boeing 777-200	4	2	0	0	0
	Boeing 777-300	3	3	0	0	0
	Boeing 787-800	6	9	7	≥ 5	≥ 5
Narrow body	Boeing 737-700	4	2	2	0	0
	Boeing 737-800	6	8	8	≥ 5	≥ 5
	Embraer 170	3	2	0	0	0
	Embraer 190	15	15	15	≈ 18	≈ 18

In 2014 Kenya Airways publicly stated that within the next 5 years its fleet was expected to be more homogeneous by disposing progressively of all aircraft older than 8 years old (B737-700) and reducing fleet types from 7 to 3. Indeed, the fleet plan for 2019 was defined on the basis of the most-likely 5-year demand forecast dated from 2015. As for the B777s recently acquired, Kenya Airways determined that they were too big to be suitable for its network and thus, they were expected to be subleased to other airlines.

Given this background, the proposed case study consists in moving backwards in time to year 2015 in order to reproduce the former KQ 5-year plan and obtain from the ADP-based tool its corresponding adaptive fleet policy. Then, the recommended adaptive fleet policy will be assessed for the current conditions of year 2017 and compared to KQ actual plan. To this end, a 5-period fleet planning problem under demand uncertainty is modelled based on the 2015 available KQ forecasts and cost data. In light of this, the reader should notice that this case study is intended to get as close as possible to a validation analysis.

7.2. Model parameters

For the KQ case study, the model parameters can be classified following the same structure presented in the proof of concept: global, aircraft-related, route-related and aircraft/route-related parameters. However, as will be discussed shortly some new parameters have been introduced to simulate the dynamic opening of new routes. Furthermore, it must be reminded that the data presented hereafter has been extrapolated from the 2015 KQ 5-year plan and 2014 relevant cost data. Apart from this, all routes analysed are named with an ID code for the sake of ensuring confidentiality.

To reproduce as close as possible the 2015 KQ 5-year plan, the multi-period adaptive fleet planning problem consists of 5 periods ($T = 5$), where each period represents a year composed of 52 weeks ($n_t = 52$). In this way, year 2015 corresponds to period 0 and year 2019 to period 4. Furthermore, the discount factor γ_{t+1} is set to 1 for the same reasons mentioned in the proof of concept. Figure D.1 in Appendix D illustrates a scheme of the scenario tree used for the KQ case study.

Taking into account the major aircraft families operated by Kenya Airways in 2015 as well the possible time constraints presented by the ADP-based tool, a representative aircraft type selection is carried out for the case study. As inferred from Table 7.2, the existent B737-700, which all have around 12 years, are merged with the initial amount of B737-800 (from 6 to 10 B738s). This is based on the mentioned KQ disposal policy, the model inability to account for aircraft age as well as the similarities existent between both configurations. Apart from that, B777-300 is neither considered into the model since its acquisition was determined based on an overestimated forecast and thus, was planned to be subleased for the next years. The same conditions applied to B777-200s but in this particular case, they have been included into the model so as to provide more realistic dimensions and assess the model performance when having two competing aircraft types in long-range (B772 and B788). As for the short-medium haul routes, both Embraer 170 and 190 are modelled together with Boeing 737-800. All in all, the ADP-based tool will deal with a total of 5 aircraft types, which are thought to be representative enough of the KQ fleet status back in 2015.

Table 7.2: Aircraft-related parameters

Aircraft	c_i^{fix} [$/week]	cap_i [-]	BT_i [h/week]	TAT_i [h/flight]	c_i^{disp} [$/weekly based]	IF_a [-]
E70	50400	69	110	0.75	10000	3
E90	58000	107	110	0.75	15000	15
B738	102000	150	110	0.75	20000	10
B788	220000	234	96	1.00	25000	6
B772	200000	322	96	1.50	25000	4

In terms of route related parameters, the case study accounts for the entire network of Kenya Airways as well as the potential opening of new routes targeted by the KQ 5-year plan. In figures, this means a total of 64 routes amongst which 16 are planned to be opened in the next 5 years. These routes are assigned an identification number (from 1 to 64) to protect KQ confidentiality. Table D.1 in Appendix D shows the operating time per flight leg OT_r, the average fare f_r assumed to be constant through the years, the average demand existing at the first operating period for each of the flight legs of a route D^*_{0r}, the load factor LF_r and the minimum frequency per route Y^{min}_{tr}, which varies according to the minimum market shares targeted by Kenya Airways. More particularly, potential new routes are not assigned a minimum frequency. In this way, the ADP algorithm is encouraged to provide advice on whether entering a specific market or not. On the other hand, Table D.2 show the aircraft-route variable costs corresponding to the variable costs of operating a route with an aircraft type. It should be noticed that for those routes exceeding the maximum allowed range of a certain aircraft, a big M cost is assigned to prevent the algorithm from choosing infeasible aircraft-route combinations.

With the aim of keeping the model as much realistic as possible, the case study also differs from the proof of concept in the fact that the 64 routes are aggregated into segmented market regions characterized by different traffic growth forecasts. Indeed, it is not longer possible to consider a global traffic growth ω_t since Kenya Airways serves very different markets. For instance, 2015 KQ annual reports stated that more than half of the growth in passenger travel was due to emerging markets such as Asia-Pacific and the Middle East: while Middle East was growing at a double digit, some African regions were barely growing. Based on these statements and KQ region forecasts, the random variable ω_t is transformed from a scalar into a random vector $\omega_t := [\omega_{tg}]_{g\in\mathscr{G}}$, where $\mathscr{G}$ represents the set of aggregated regions. Therefore, ω_t is expected to take the values of three finite outcome vectors $(\omega^0_t, \omega^1_t, \omega^2_t)$, which correspond to the best case (H), most likely (M) and worst case (L) scenarios respectively.

$$\omega_t \in \left\{\omega^0_t, \omega^1_t, \omega^2_t\right\} = \{\mathrm{H}, \mathrm{M}, \mathrm{L}\}$$

The annual traffic growth values forecasted for each of these scenario vectors are found in Table 7.3, whereas a more detailed table showing the aggregation of routes can be found in Table D.3 of Appendix D. As done for the routes, each region is denoted with a letter from A to J for confidentiality reasons.

Table 7.3: Demand forecasts per aggregated region of routes

. Region	Number of routes	Best case	Most-likely case	Worst case
A	9	0.00%	0.00%	-8.00%
B	3	5.00%	2.00%	-1.00%
C	11	8.00%	4.45%	-3.00%
D	5	5.00%	0.69%	-4.00%
E	4	5.00%	0.98%	-1.00%
F	4	9.00%	5.00%	-3.00%
G	6	5.00%	2.00%	-6.00%
H	14	10.00%	5.00%	-3.00%
I	7	8.00%	3.00%	-5.00%
J	1	5.00%	1.00%	0.00%

For the case study, these annual growth values are assumed to be constant year over year. However, the ADP support tool provides the possibility to change them dynamically to simulate a possible stagnation in future market growth. Furthermore, their corresponding discrete probability distribution is slightly varied with respect to the proof of concept. Taking into account that Kenya Airways faced certain market instabilities in the recent years due to terrorist attacks and epidemic diseases, the potential decrease in demand will be assigned a higher probability of occurrence.

$$p^n_t(\omega^s_t) = p^s_t = \begin{cases} 25\%, & \text{if} \quad \omega^s_t = \omega^0_t \\ 50\%, & \text{if} \quad \omega^s_t = \omega^1_t \\ 25\%, & \text{if} \quad \omega^s_t = \omega^2_t \end{cases}$$

Finally, it must be reminded that by 2014 Kenya Airways was planning to open 16 new routes as part of its expansion strategy. This was reflected in a 5-year rollout plan as illustrated in Table 7.4, where each route is identified with its ID number. Thus, the ID numbers that appear in the table refer to either newly opened or closed routes, whereas absent ID numbers represent those routes already operated by 2014. According to this plan, 6 routes were targeted in year 2016, 1 exit and 3 entries in year 2017 and 7 new routes for 2018.

Table 7.4: KQ rollout plan where routes are identified by their ID number from 1 to 64

Rollout years	2015	2016	2017	2018	2019
New route exits			2		
New route entries		7	1	8	
		21	6	18	
		22	64	26	
		41		28	
		55		32	
		56		36	
				38	

To simulate the dynamic expansion of Kenya Airways a new model parameter called entry-into-market indicator δ_{tr} is introduced:

$$\delta_{tr} = \begin{cases} 1, & \text{if} \quad \text{route } r \text{ is already operated or planned to be opened in period t} \\ 0, & \text{if} \quad \text{route } r \text{ is not planned to be opened in period t} \end{cases} \tag{7.1}$$

In this way, the route demand for period 0 is

$$D_{0r} = \delta_{0r} D^*_{t0}, \tag{7.2}$$

while for periods other than the initial one the route demand is deduced as follows:

$$D_{t+1r} = \left(\widehat{D}_{t+1r} + D_{tr}\right)\delta_{t+1r} = \Delta_{t+1r} D^*_{t0} \delta_{t+1r} \qquad \forall t \in \{1, ..., \mathbb{T} - 1\} \tag{7.3}$$

where Δ_{t+1r} denotes the total demand growth experienced from the first operating period of that route, whose demand is D^*_{0r}. In this context, Δ_{t+1r} is reformulated as:

$$\Delta_{t+1r} = \begin{cases} 1, & \text{if} \quad \delta_{tr}\delta_{t+1r} = 0 \\ (1+\omega_{t+1g})\Delta_{tr}, & \text{if} \quad \delta_{tr}\delta_{t+1r} = 1 \end{cases} \tag{7.4}$$

Equation 7.4 can be understood as follows: If a route is not opened yet ($\delta_{t+1r} = 0$) the total demand growth Δ_{t+1r} is set to 1, implying that the demand has not varied from period 0, thereby being kept as zero by Equation 7.3. Likewise, if it is the first period in which that route is open ($\delta_{t+1r} = 1$ and $\delta_{tr} = 0$), then the total demand growth is also set to 1 to prevent the demand discontinuity entailed by opening a new route. In this situation, it is concluded from equation 7.3 that $D_{t+1r} = D^*_{t0}$. Finally, if a route is already being operated for at least 1 period ($\delta_{t+1r} = 1$ and $\delta_{tr} = 1$) , then the total demand growth will be equivalent to the sum of last period total demand growth and the stochastic growth variation for the present year:

$$\Delta_{t+1r} = \Delta_{tr} + \widehat{\Delta}_{t+1r} = \Delta_{tr} + \omega_{t+1g}\Delta_{tr} = (1+\omega_{t+1g})\Delta_{tr} \tag{7.5}$$

In light of the above definitions, it must be considered the fact that a new route will most likely not share the same total demand growth of the region to which it belongs:

$$\Delta_{t+1r} \neq \Delta_{t+1g} \tag{7.6}$$

For this reason, if there is any recently opened route, then its corresponding growth Δ_{tr} must be accounted independently from the overall region growth Δ_{tg}. Thus, all new growth parameters resulting from recently opened routes, $\Delta_t^* := [\Delta_{tr}]_{r' \in \mathscr{R}^{new}}$, will be concatenated to the original vector, $\Delta_t^g := [\Delta_{tg}]_{g \in \mathscr{G}}$, which tracks total demand growth for each region:

$$\Delta_t := \begin{pmatrix} \Delta_t^g & \Delta_t^* \end{pmatrix} \tag{7.7}$$

$$S_t = (R_t \quad D_t) \quad \rightarrow \quad S_t^g = (R_t \quad \Delta_t) \tag{4.43}$$

To conclude, the ADP stepsize parameter α_{n-1} is regulated with the already presented staircase rule:

$$\alpha_{n-1} = \begin{cases} 1, & \text{if} \quad n < N_1 \\ \overline{\alpha}_1, & \text{if} \quad N_1 \le n < N_2 \\ \overline{\alpha}_2, & \text{if} \quad N_2 \le n < N_3 \\ \overline{\alpha}_3, & \text{if} \quad N_3 \le n < N_4 \\ \overline{\alpha}_4, & \text{if} \quad N_4 \le n < N_{stop} \end{cases} \tag{7.8}$$

where values follow $\overline{\alpha}_1 \ge \overline{\alpha}_2 \ge \overline{\alpha}_3 \ge \overline{\alpha}_4$ to meet the three previous conditions for convergence. For the present case study, $\overline{\alpha}_n$ takes the same sequence of values as in the proof of concept: $\{1, 1, 0.5, 0.4, 0.4\}$. On the other hand, the control parameter λ is calibrated to 0.6 for positive values of $\widetilde{C}_t^*$ (profits) and 1.4 for negative values (losses). These selections have been based on the results obtained from their sensitivity analysis in the proof of concept. In terms of iteration number, it is decided to carry out a long simulation of 15000 iterations to ensure the highest possible quality of results. Maintaining the exploration/exploitation proportion recommended in Chapter 4, the iteration parameters are:

$$\begin{pmatrix} N_1 & N_2 & N_3 & N_4 & N_{stop} \end{pmatrix} = (6450 \quad 11100 \quad 12900 \quad 13800 \quad 15000)$$

7.3. Results

In this section the results for the case study are presented and structured into two parts. The first part focuses on the algorithm learning behavior, whereas the second part analyzes the adaptive fleet policy obtained with the ADP algorithm.

7.3.1. Learning behavior of the enhanced ADP algorithm

The enhanced ADP algorithm presents an iterative learning behavior which shares many similarities to the one seen in the proof of concept.

Figure 5.8 illustrates the behavior of the ADP algorithm when solving the stochastic problem for 15000 iterations. As in the other experiments, the green dots correspond to the observations of the total operating profit expected to be earned during the 5 periods: $V_0(S_0)$. Likewise, the red line shows the rolling average for the 20 subsequent observations obtained during the last 20 iterations, which allows to identify the general trend throughout the iterations.

In comparison to the stochastic problem solved for the proof of concept, this case study requires longer exploration times to start converging: while the previous small test case started to stabilize by the first 500 iterations, the KQ case study does it for the first 2000 iterations. This is explained by the greater dimensions of the present problem. Furthermore, the increased complexity of this case study leads to higher levels of noise throughout the iterations and thus, higher margins for the standard deviation between value function observations. Taking this behaviour into account, the number of iterations must be increased significantly to learn coherent policies.

Over the iterations, the standard deviation decreases steadily since the algorithm is learning better approximations of the value function, fact which reduces the noise amongst observations. Figures 7.2 and 7.3 illustrate this trend by plotting together the rolling mean and standard deviation for the iterations intervals

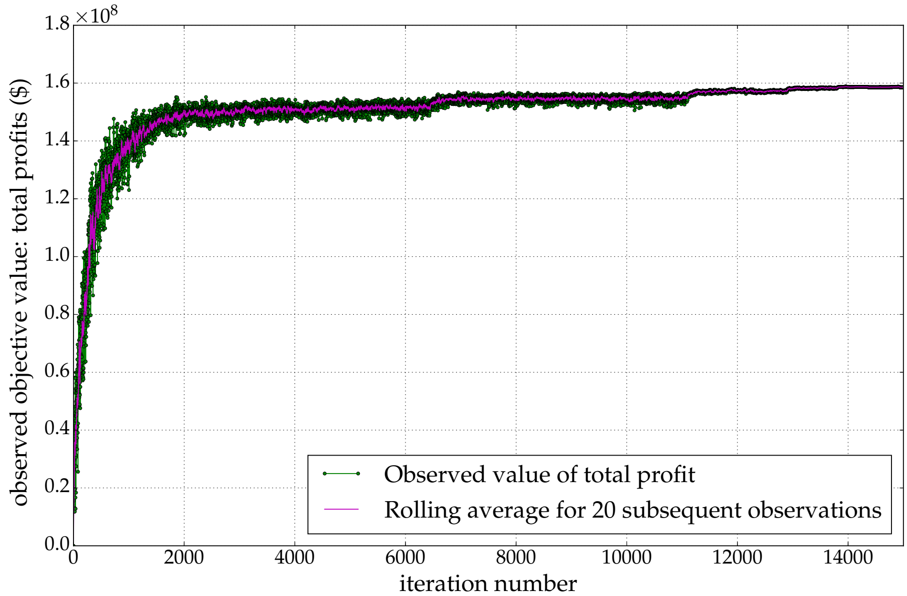


Figure 7.1: 5-period scenario tree applicable to the KQ case study.

[0,6000] and [6000, 15000]. By analyzing both figures, it becomes apparent the importance of balancing the exploration and exploitation phases: exploration allows for faster learning rates and provides a solid value function approximation, upon which the exploitation phase can build an accurate adaptive policy. Indeed, exploitation can increase the objective function by 5% more, which implies an average increase in annual operating profits of $ 2 million.

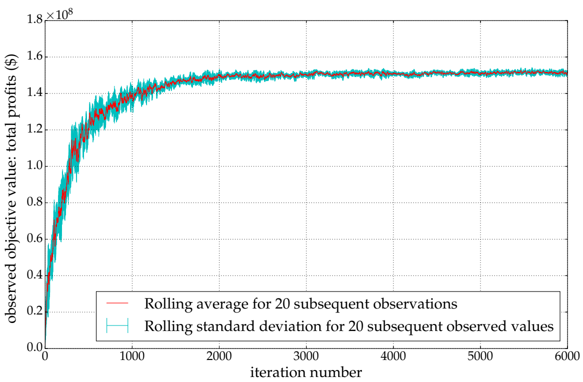


Figure 7.2: Rolling mean and standard deviation during exploration phase

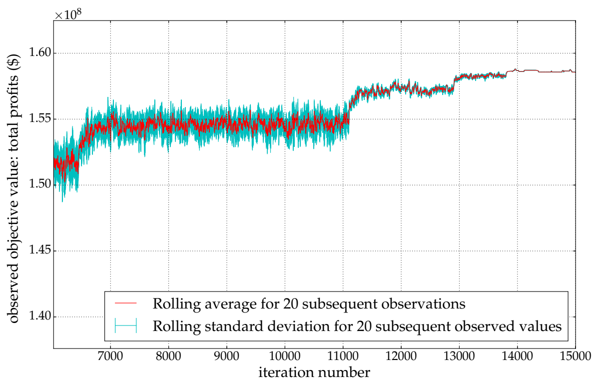


Figure 7.3: Rolling mean and standard deviation during exploitation phase

Apart from this, Figure 5.11 shows in particular that the standard deviation decreases exponentially during the exploration phase. However, it decreases at a lower pace than in the proof of concept. This fact is a direct consequence of the longer exploration times. On the other hand, during the exploitation phase shown in Figure 5.12 the exponential behavior is less noticeable and thus, it takes more time to mitigate the standard deviation.

In terms of computational time, the entire simulation with the enhanced ADP algorithm lasted 10h. Nevertheless, it should be noted that initialization tasks took around 60% of the total time when compared to the previous 30% estimated during the proof of concept. This increase is mainly due to the fact of dealing with a more complex case study of higher dimensions: 5 aircraft types and 64 routes. Indeed, the algorithm needs to

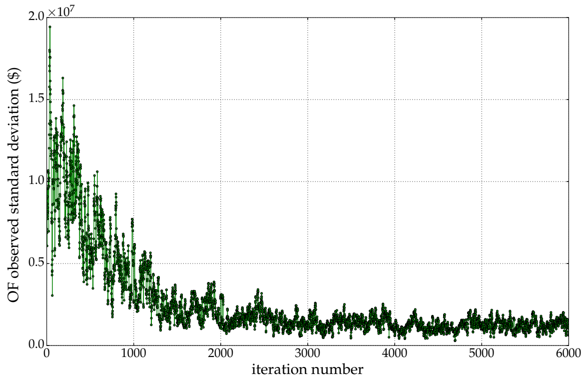


Figure 7.4: Exponential reduction of standard deviation in exploration phase

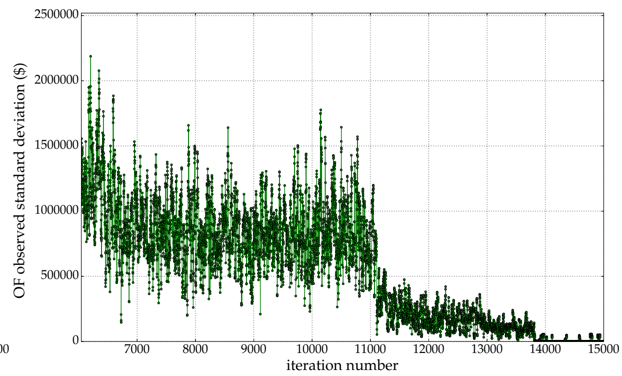


Figure 7.5: Reduction of standard deviation in exploitation phase

browse a wider state space to obtain a useful initial set of observations to train the kernel regression and start the ADP pure loop. Fortunately, the enhanced algorithm performance makes up for the longer initialization tasks by reducing sharply the simulation time of the pure ADP loop down to 2 hours. The CPU performance for the pure ADP algorithm without the initialization tasks can be appreciated in Figure D.13 in the Appendix D.

In any case, it should be reminded that the kernel training only needs to be solved once, thereby allowing independent runs of different ADP loops. Apart from this, it must be reminded from previous sensitivity and verification analysis that observations are obtained from a baseline case. This means that the more the assessed case differs from the baseline, the less confidence the ADP support tool will perform at its best. All in all, the algorithm proved to be very robust in previous analysis, where wide confidence intervals were determined as well. Should any desired parameter exceed the confidence interval, then it is necessary to repeat the initialization tasks for that specific case.

7.3.2. Adaptive fleet policy

Given the size of the case study, the obtained fleet policy appears to be more complex than the one presented in the proof of concept. Figure 7.6 illustrates how the recommended fleet adapts to each of the 81 scenarios composing the 5-stage scenario tree. All fleet data related to each aircraft type is presented in format (E70, E90, B737, B788, B772). As a reminder, blue vectors express the number of aircraft of each type R_{ti} that KQ is advised to possess at a certain state S_t. On the other hand, yellow vectors correspond to the fleet related decisions (x_t^{buy}, x_t^{disp}) recommended to be made during period t given the state S_t: positive values imply new aircraft acquisitions for next year whereas negative values, disposals for next year.

By analyzing the evolution of the initial fleet status, the first fact to be noticed is that KQ is considered to have too many B772s for the existent demand across its network. From the adaptive policy, it is concluded that it is more profitable to assume the disposal costs of all 4 aircraft rather than still operating them. Indeed, B772 type is more suitable for those routes with high density of passengers and lower levels of frequency. In turn, it is preferable to invest in 3 more Dreamliners: despite its lower capacity, B788 type is more efficient when it comes to operate long-haul routes with the levels of demand expected by KQ. As for the short-medium range fleet, it can also be appreciated a shift in composition. The adaptive policy advices to dispose two E70s, one E90 and one B738 with the objective of adjusting more the fleet size to the existent demand of regional routes. In conclusion, KQ would have been advised to possess a fleet of 1 E90s, 14 E70s, 9 B737s and 9 B788s in year 2016.

Over the following periods, the adaptive policy provides different recommendations in function of how demand evolves. In doing so, it tries to respect as much as possible the optimal fleet plans obtained for each scenario solved independently. Indeed, the advised fleet plans tend to be really similar to the optimal fleet of each independent scenario, which are all shown in Table D.5 of Appendix D.

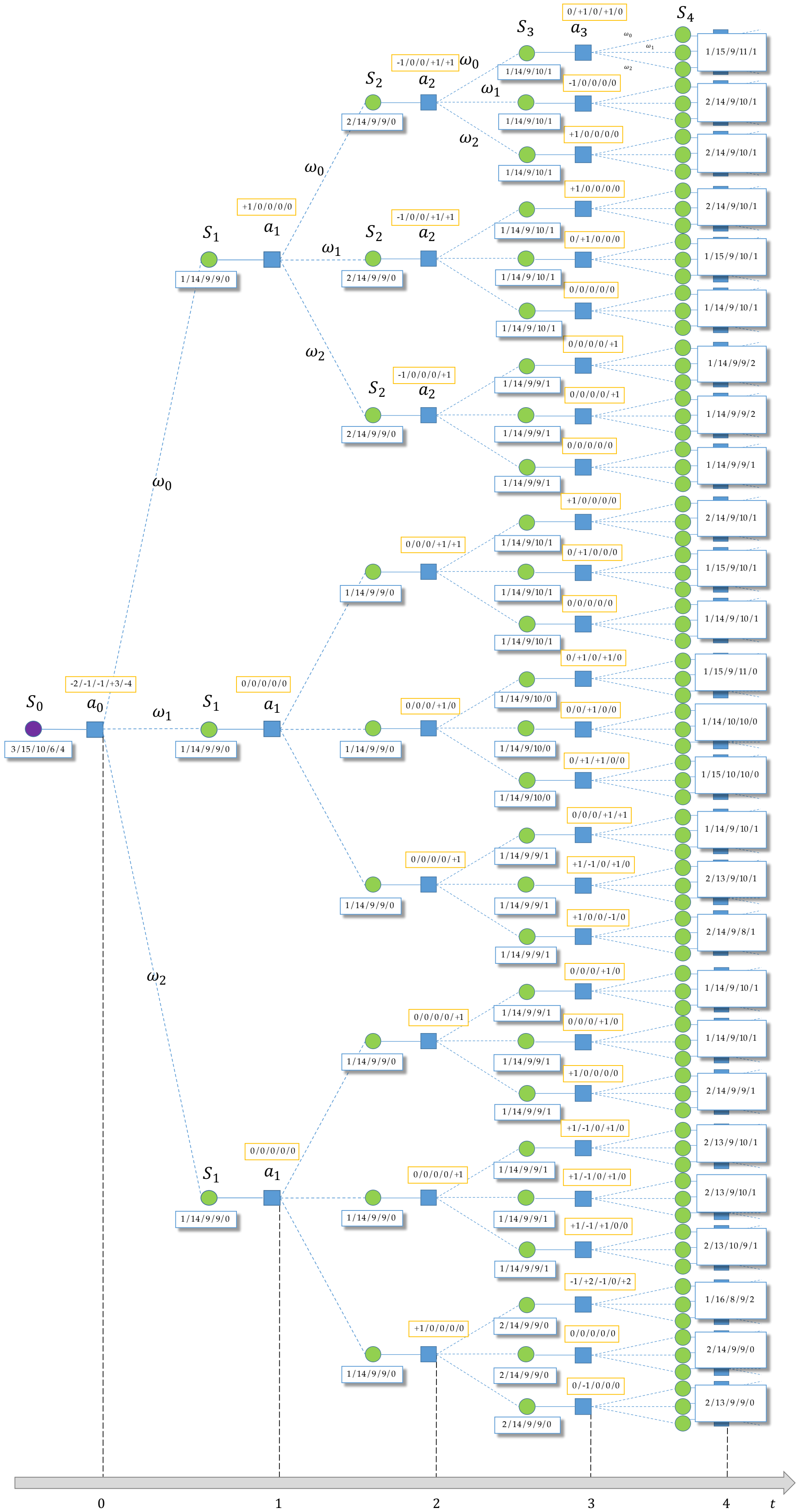


Figure 7.6: ADP recommended fleet policy for KQ 5-year plan.

In this context, some trends can be identified if these recommendations are examined in more detail. For instance, one could observe that the number of aircraft E70 advised eventually decreases when approaching scenarios with higher demand growth, meanwhile the number of recommended B737 and E90 increases. Indeed, for higher levels of demand, it is more profitable to slightly increase the capacity of the aircraft operated. Evidently, there is a limit to this behaviour since demand is highly influenced by frequency levels and consequently, a minimum frequency must be maintained so as not to lose market share. In any case, it must be highlighted that the number of B737 recommended does not experience major changes, ranging from 9 in the worst scenarios and 11 in the best cases. What is surprising is that in the last period the airline is recommended to rebuy a B772 in the majority of scenarios and independently of their demand growth evolution. Nevertheless, this is mainly due to two reasons. Firstly, for those scenarios experiencing the greatest decay in demand, the adaptive fleet policy does not advise to possess one B772 until the last period. This can be understood as a way to minimise costs by reducing frequency while still carrying the maximum amount possible of passengers. Therefore, this recommendation is a direct consequence of not considering demand elasticity with respect to frequency in the model formulation: the algorithm will tend to minimize frequencies to reduce costs. The second reason is applicable to those scenarios featuring higher demand growth in year 2018: once demand has grown steadily from year 2015 onwards, the adaptive policy considers this is the adequate time to operate long-haul routes with one B772. Nevertheless, the majority of high-density and long-haul routes are still operated with Dreamliners. In any case, the adaptive policy underpins the already known fact that the acquisition of 4 B772s back in 2014 was a result of an overestimated forecast.

Apart from this, path dependency is still present in the adaptive policy. As discussed in the proof of concept, adaptive policies capture the influence of reaching the same conditions from different histories of demand evolution. Table 7.5 provides a clear example by showing the recommended fleet evolution for three scenarios with same total demand variation but different stochastic processes. By comparing their fleets, it can be concluded that scenario IMHML and IMMHL face less financial risk to invest in more aircraft (36 in total), whereas the scenario IMLMH has invested in a cheaper fleet composition due to experiencing instabilities in demand growth (35 in total). Indeed, differences in fleet composition are also quite visible: the scenarios with less risks (IMHML and IMLMH) deal with more expensive aircraft in general, whereas the scenario that has priorly experienced decreases in demand (IMLMH) invests in cheaper aircraft such as E70.

Table 7.5: Example of path dependency amongst scenarios with same demand variation

				Recommended fleet plan				
ID	Scenario	Total probability	Demand variation	year 0	year 1	year 2	year 3	year 4
32	IMHML	1.5%	22%	3/15/10/6/4	1/14/9/9/0	1/14/9/9/0	1/14/9/10/1	1/15/9/10/1
48	IMLMH	1.5%	22%	3/15/10/6/4	1/14/9/9/0	1/14/9/9/0	1/14/9/9/1	2/13/9/10/1
66	IMMHL	1.5%	22%	3/15/10/6/4	1/14/9/9/0	1/14/9/9/0	1/14/9/10/0	1/15/10/10/0

In terms of economical impact, Table D.6 in Appendix D presents a comparison between the average weekly operating profits that would be earned, by either following the optimal fleet plan of the most-likely scenario or either following the recommendations of the adaptive fleet policy. As previously described in the proof of concept, both options are compared to the maximum operating profits that could be obtained if every scenario was deterministic and independent. This difference is expressed in the form of relative error with respect to the best value (Dif.Best). An overview of some of the results is provided in Table 7.6.

The first notable observation is that the order of magnitude of the Dif.Best values are much higher than those previously seen in the proof of concept. This is entailed by the fact of dealing with a much larger and complex network: any fleet modification as well as any suboptimal operational decision makes a higher impact across the network. In this context, the percentage difference between the objective values of each method shows that the adaptive policy clearly outperforms the most-likely fleet plan when applied to the most extreme scenarios. In turn, the most-likely solution performs better in those scenarios similar to the most-likely scenario. This is not surprising since the most-likely fleet plan has been calculated regardless of the possibility of other uneven scenarios happening. In contrast, the solution provided by the adaptive policy accounts for this uncertainty. However, it must be highlighted that the ADP worst scores do not differ

Table 7.6: Extract of Table D.6: performance analysis of recommended ADP policy against most-likely solution and independent-scenario solutions - operating profits expressed on a weekly basis in USD

				Independent scenarios		Most-likely solution		ADP adaptive policy	
ID	Scenario	Total probability	Demand variation	Optimality Gap	OF	Dif.Best	OF	Dif.Best	OF
0	IHHHH	0.39%	32.02%	0.82%	1289499	-5.30%	1221168	-0.87%	1278281
1	IHHHM	0.78%	26.76%	0.78%	1233933	-4.58%	1177390	-0.96%	1222100
2	IHHHL	0.39%	18.89%	0.52%	1121545	-3.66%	1080508	-2.11%	1097912
3	IHHMH	0.78%	27.19%	0.94%	1172776	-3.59%	1130704	-0.92%	1162025
4	IHHMM	1.56%	22.14%	0.66%	1114352	-3.23%	1078317	-0.83%	1105112
5	IHHML	0.78%	14.57%	0.60%	1009371	-2.86%	980456	-1.29%	996338
36	IMMHH	1.56%	23.08%	0.90%	901812	-1.56%	887708	-0.84%	894259
37	IMMHM	3.13%	18.21%	1.03%	835186	-0.69%	829404	-0.88%	827838
38	IMMHL	1.56%	10.90%	0.78%	740436	-1.13%	732101	-2.12%	724710
39	IMMMH	3.13%	18.64%	0.95%	781678	0.16%	782945	-0.59%	777054
40	IMMMM	6.25%	13.96%	1.29%	728496	0.00%	728496	-0.22%	726902
41	IMMML	3.13%	6.93%	0.83%	629653	-0.84%	624386	-0.96%	623584
78	ILLLH	0.39%	0.40%	1.50%	-124688	-37.33%	-171234	-20.52%	-150273
79	ILLLM	0.78%	-3.53%	1.50%	-185460	-30.90%	-242773	-11.11%	-206070
80	ILLLL	0.39%	-9.33%	1.50%	-298990	-28.94%	-385524	-7.66%	-321899
			Expected values		618438	-1.99%	606154	-1.59%	608596

significantly from the better ones presented by the most-likely solution. Furthermore, the worsening impact of the most-likely fleet plan becomes much more evident the more the scenarios differ from the most-likely case. As in the proof of concept, the worst scores are found in the neighborhood of the extreme scenarios: For the most pessimistic scenarios the most-likely solution can imply weekly operating losses with a Dif.Best ranging between -37% and -29%, whereas in the most optimistic scenarios losses remain between 6% and 4%. In contrast, the recommendations of the adaptive fleet policy can help reduce the Dif.Best by more than 50% with respect to the most-likely case.

Finally, the last row of Table 7.6 shows the expected weekly operating profits for the most-likely fleet plan and the ADP policy. By analyzing these last results, it can be concluded that the adaptive policy reduces by 20 % the profit losses of the most-likely solution. The reader may notice that this reduction is inferior to the 50% reduction presented in the proof of concept. The main reason for this result is the level of growth disparity existing between the extreme scenarios and the most-likely scenario. If these extreme scenarios happened to present a demand variation less different with respect to the most likely case, then the most-likely fleet would perform better than in a problem with more uneven scenarios. Indeed, scenarios would be more uniform and similar to the most-likely one, which would enhance the performance of the most-likely fleet and reduce the advantage presented by the adaptive fleet policy. All in all, this explains why the adaptive fleet policy presents slightly less advantage with respect to the most-likely solution: in the proof of concept the random variable ω_t happened to take more uneven values (15%, 5% and -5%) that in the case study. In any case, both proof of concept and case study prove the adaptive fleet policy as the most robust method to take decisions under uncertainty. Even more importantly, approximate dynamic programming has proven to be a very powerful method to solve efficiently a multi-period fleet planning problem of realistic dimensions.

7.4. From an adaptive policy to the definition of signposts

Adaptive fleet policies are meant to include uncertainty into the decision-making process of planning the future fleet of an airline. As previously seen in Figure 7.6, a policy consists in providing the best rule for making decisions given certain conditions. Since it does not provide a single solution, decision-makers might find challenging to draw clear conclusions from the scenario tree structure. Taking into account this fact, the results from the adaptive policy can be represented in a more general way where useful signposts for fleet planners could be detected more intuitively. In this context, the following subsections propose different methods to post-process the recommendations of the adaptive policy.

7.4.1. Fleet maps

Fleet maps consist in colored area plots displaying the probability of the adaptive policy recommending a certain fleet status given some demand conditions and time period. This type of plots are created by means of browsing the different fleet status recommended for every demand variation in each period. That is to say, a fleet map results from adding up all the probabilities of recommending in a certain period a certain fleet composition and size for different demand growth. Eventually, fleet maps are a way to represent the probability distribution of the recommendations provided by the adaptive policy.

When dealing with a discontinuous scenario tree such as in Figure 7.6, the probability distribution becomes discontinuous and can also be represented by a table like the one presented below.

Stage	3

Suma de Total probability / FILTER FLEET SIZE & CLICK BUTTON TO SEE BEST COMPOSITION	FILTER DEMAND -2,9	2,3	2,5	2,7	6,9	7,4	8	8,3	8,5	13	13,2	13,3	13,7	13,9	14,1	14,5	18,6	19,2	19,7	19,9	20,1	20,5	25,9	26,2	26,5	32,7	Total general
⊞ 35																	1,6%						3,1%			1,6%	6,3%
⊞ 34					1,6%	1,6%	6,3%			3,1%	3,1%	3,1%	3,1%			12,5%		1,6%	1,6%	6,3%	6,3%	6,3%		3,1%	3,1%		62,5%
⊞ 33	1,6%	3,1%	3,1%	3,1%			1,6%	6,3%	6,3%					3,1%	3,1%												31,3%
Total general	1,6%	3,1%	3,1%	3,1%	1,6%	1,6%	7,8%	6,3%	6,3%	3,1%	3,1%	3,1%	3,1%	3,1%	3,1%	12,5%	1,6%	1,6%	1,6%	6,3%	6,3%	6,3%	3,1%	3,1%	3,1%	1,6%	100,0%

Figure 7.7: Table of discrete probability for every fleet-demand variation combination

The colors scale goes from green to red intense, denoting the lowest and highest probability values respectively. Following this logic, it can be appreciated that a fleet size of 34 aircraft will be advised by the ADP tool with 62.5% probabilities. However, it has also been seen that demand variation plays an important role when it comes to advise fleet size. Indeed, it must be highlighted that a smaller fleet is more likely to be advised for scenarios with low demand growth, while a larger fleet is more probable for scenarios with high demand. Apart from this, Figure 7.8 shows that there are fleet compositions with higher probability to be recommended than others given the same fleet size and demand variation.

Stage	3

Suma de Total probability / FILTER FLEET SIZE & CLICK BUTTON TO SEE BEST COMPOSITION	FILTER DEMAND -2,9	2,3	2,5	2,7	6,9	7,4	8	8,3	8,5	13	13,2	13,3	13,7	13,9	14,1	14,5	18,6	19,2	19,7	19,9	20,1	20,5	25,9	26,2	26,5	32,7	Total general
⊞ 35																	1,6%						3,1%			1,6%	6,3%
⊟ 34					1,6%	1,6%	6,3%			3,1%	3,1%	3,1%	3,1%			12,5%		1,6%	1,6%	6,3%	6,3%	6,3%		3,1%	3,1%		62,5%
1/13/10/9/1										3,1%										6,3%				3,1%			12,5%
1/13/10/10/0					1,6%						3,1%	3,1%						1,6%			6,3%				3,1%		18,8%
2/13/9/9/1							6,3%									12,5%						6,3%					25,0%
2/12/10/9/1						1,6%							3,1%						1,6%								6,3%
⊞ 33	1,6%	3,1%	3,1%	3,1%			1,6%	6,3%	6,3%					3,1%	3,1%												31,3%
Total general	1,6%	3,1%	3,1%	3,1%	1,6%	1,6%	7,8%	6,3%	6,3%	3,1%	3,1%	3,1%	3,1%	3,1%	3,1%	12,5%	1,6%	1,6%	1,6%	6,3%	6,3%	6,3%	3,1%	3,1%	3,1%	1,6%	100,0%

Figure 7.8: Discrete probability assigned to different fleet composition in scenarios with low demand variation

Nevertheless, the utility of these tables can be questioned easily since they provide a discrete representation of reality. Even though this is a direct consequence of assuming a discrete scenario tree, extrapolation of results can mitigate this arguable weakness. This is done by means of applying a kernel density estimator, with the objective of approximating the continuous density probability function based on the finite set of adaptive recommendations. An example of the corresponding results for the total fleet size and an aircraft type are shown in Figures 7.9 and 7.10. The reader is referred to Appendix D for the analysis of the rest of fleet maps obtained.

The analysis of the fleet maps allows to identify the following general trends:

- **Total fleet size:** Starting from a total fleet size of 38 aircraft in period 0, a reduction to 33 aircraft is recommended uniformly across the different demand variations in period 1. This is due to the tree structure adopted. For later periods 2, 3 and 4 the area of probability spreads considerably as a consequence of the higher levels of uncertainty faced and thus, lowers down the overall probabilities amongst fleet-demand combinations recommended. In any case, a clear trend is visible: since demand is forecasted with a slight tendency to grow, the best performing fleet sizes will tend to grow as well with the periods. This can be identified by the most intense colors in the fleet map of each period. In fact, grey and reddish colors represent those combinations of fleet size-demand variation most likely to occur. Generally speaking, the best performing fleet sizes will be those covering a wider demand variation interval. The wider the demand variation is, the more robust the fleet size will be. Likewise, the fleet map can also provide information in the opposite direction: Given a demand growth variation interval, the fleet map illustrates the range of aircraft sizes more suitable for those assessed conditions. This allows to adopt different ranges of confidence when analyzing the suitability of a fleet: either pessimistic, most probable or optimistic studies. In light of this, fleet planners could identify the boundaries of each desired envelope as useful signposts to make decisions.

The same type of analysis can be performed for each type of aircraft so as to draw the most suitable fleet composition under different conditions.

- **E70:** From the fleet map of aircraft E70, it can be inferred that the ADP tool tends to recommend more disposals when higher levels of demand growth are expected. In the same way, a higher quantity of two E70s is recommended for the worst case scenarios. As commented previously, this is explained by its low fix and operating costs as well as its reduced capacity: the adaptive policy captures a lower demand and consequently, saves costs by reducing the number of available seats provided.

- **E90:** Embraer-190 type shows a higher degree of robustness across the different demand growth scenarios. The number of aircraft of this type ranges between 13 and 16. There is slight trend to recommend more E90s when demand grows; however, it can also be appreciated that a lower quantity of 14 aircraft is recommended in very similar scenarios. Either choosing one quantity of E90 or another will depend on the resulting fleet composition of the airline. For instance, if demand has not grown in period 4 (demand variation is equal to zero) and the airline already has 2 E70s, then the most probable recommendation will be to have 13 E90s rather than 15. This is due to the fact that E70 and E90 types are mainly substitutes.

- **B738:** The quantity of B738s increases for higher demand growth expected. However, by analyzing its fleet map evolution, it can be stated that the B738 type is stabilized in 9 aircraft across the majority of scenarios, thus proving to be robust under uncertainty. More specifically, the most recommended fleet size is 9 in period 1 and 2 and then turns into 10 for period 4. Apart from this, it can also be noticed that the type B738 is very polyvalent, thereby becoming competitor of both short-haul aircraft (E70 and E90) and medium-long haul aircraft (B788).

- **B788:** For the case of the Dreamliner, it is clear that the probability of recommending to increase its quantity is proportional to the levels of demand growth expected. Since demand forecast tend to present a slight tendency towards growth, it is not surprising that the most probable recommendations pass from possessing 6 B788s to 9 or even 11. Again, the quantity of Dreamliners is highly influenced by future demand variation and the traffic growth estimated for long-haul routes. Indeed, wide-body aircraft are more profitable for those routes featuring high density.

- **B772:** The fleet map for the B772 backs up the fact that the initial four B772s resulted from an overestimated demand growth. Indeed, the adaptive policy recommends to dispose them all for periods 1 and 2. As explained earlier, in the last period the airline is recommended to rebuy a B772 in the majority of scenarios and independently of their demand growth evolution. Nevertheless, this was mainly due to two reasons. On the one hand, for those scenarios experiencing the greatest decay in demand, the acquisition of one B772 can be understood as a way to minimise costs by reducing frequency while still carrying the maximum amount possible of passengers. It must be reminded that this recommendation is a direct consequence of not considering demand elasticity with respect to frequency in the model formulation: the algorithm will tend to minimize frequencies to reduce costs. On the other hand, for those scenarios featuring higher demand growth by period 3, the adaptive policy considers that it is the right moment to operate long-haul routes with one B772. Nevertheless, the majority of high-density and long-haul routes are still operated with Dreamliners.

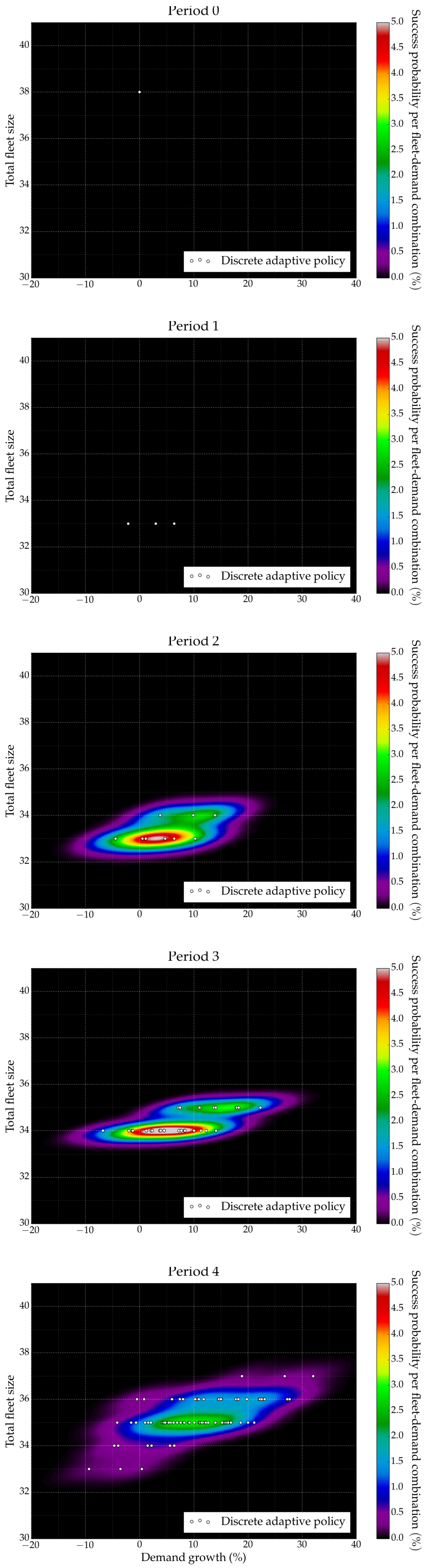


Figure 7.9: Evolution of total fleet size map

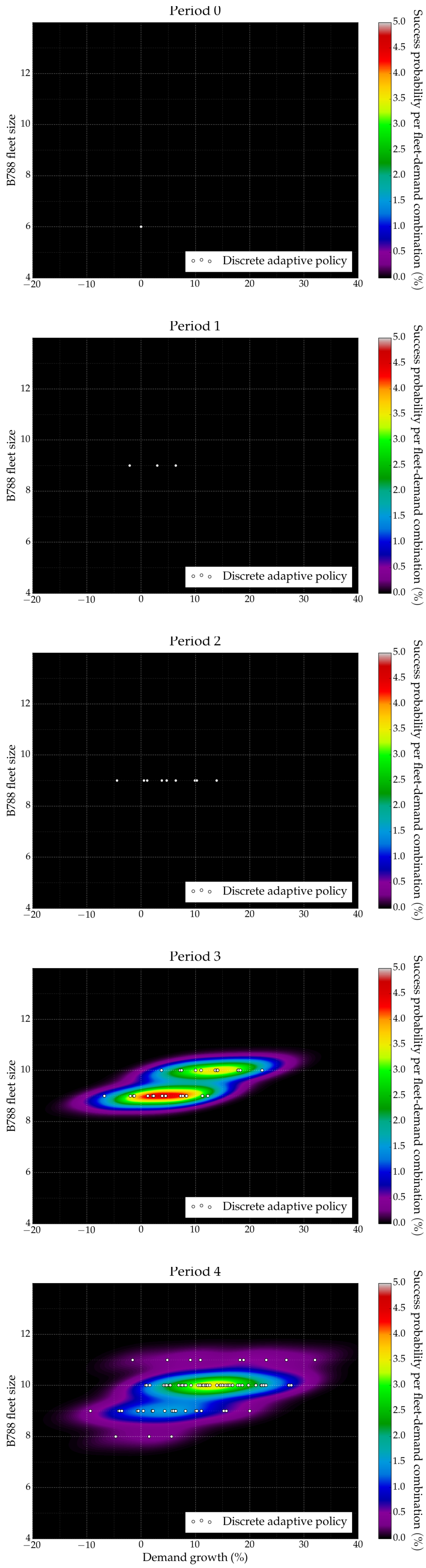


Figure 7.10: Evolution of B788 fleet map

7.4.2. Operational maps

Operational maps are meant to provide an overview of the operating-related decisions advised by the adaptive policy. Although the main focus has been so far on fleet planning, it should be remembered that the adaptive policy can also provide an operational perspective to the problem. Nevertheless, the analysis of operating decisions is significantly hindered by the complexities of browsing wide scenario trees with multiple stages. In light of these constraints, the goal of operational maps is to provide a more intuitive way to analyze the results in terms of weekly operating profits, weekly route frequency and passengers carried per week. In doing so, they illustrate the accumulated impact of each of the routes in the network, which are presented in a descendant order according to their averaged contributions. Furthermore, this characteristic of operational maps allows to obtain Pareto representations of the most important routes concerning either profit, frequency or volume of passengers carried. In other words, it helps fleet planners to identify the crucial routes within KQ network: those routes providing the highest/lowest operating profit, those routes taking the major/minor number of slots as well as those ones with high/low density of passengers.

Figure 7.11 illustrates the operational maps extracted from the adaptive fleet policy applied to the KQ case study in period 4. The full evolution of the operational maps throughout the periods can be found in section D.6 of Appendix D. They represent the profits, route frequencies and volume of passengers carried throughout the different stages of the scenario tree. As already seen in the fleet maps, the range of possible demand variations spreads considerably with the number of periods as a result of dealing with higher levels of uncertainty. Dashed blue lines between the bars represent the interpolation between the different discrete values so as to provide continuity. Apart from this, it must be stated that every colored bar represents the averaged aggregation of different possible scenarios facing the same demand variation. In fact, since the adaptive policy captures the path dependency across scenarios with same demand variation, an average has to be made amongst all the accumulated values. This is the reason for which there are some errors bars plotted, which show the existent standard deviation between values across scenarios with similar demand variationmulti-period. However, the impact of this error is minor when it comes to drawing the main operational trends. Color legend applies grey/reddish colors to the most impactful routes and dark blue black to the ones barely operated or even not opened.

By analyzing the graphs, it is appreciated how operating profits are expected to be higher when demand growth increases. Indeed, these are more sensitive to demand variation than the level of frequencies operated, whose response is more uniform and stable. Evidently, the number of passengers transported highly depends on demand growth, thereby behaving similarly to the response of the operating profits. Finally, it should be noticed that it is also possible to analyze the assignment of route frequency for each aircraft type. This study could be helpful for analyzing the utilization rates of each aircraft type.

7.5. Validation

To the greatest extent possible, this section attempts to validate the results of this case study by benchmarking them against the actual KQ historical data dating from 2015. This set of validation experiments have been carried out right on the network planning department of Kenya Airways, fact which allows for an unbiased review of the validation results.

First of all, the historical demand evolution experienced by Kenya Airways will be reproduced across the scenario tree. In this way, it is possible to determine which fleet plan would have been recommended by the adaptive policy starting from 2015 onwards. Then, this fleet plan will be compared to the actual development of the KQ fleet plan over the years. This allows to evaluate the profitability performance of the ADP recommended fleet plan and the one actually followed by Kenya Airways.

For the sake of completeness, a comparison is also carried out between the ADP network related decisions and the current situation of Kenya Airways. The fact of contrasting both networks enables a better understanding of the fleet related decisions and thus, it helps draw the full picture of the problem.

Finally, the validation is concluded by an expert survey which gathers realistic feedback on the tool. Transavia's network department must also be acknowledged for its collaboration with the survey.

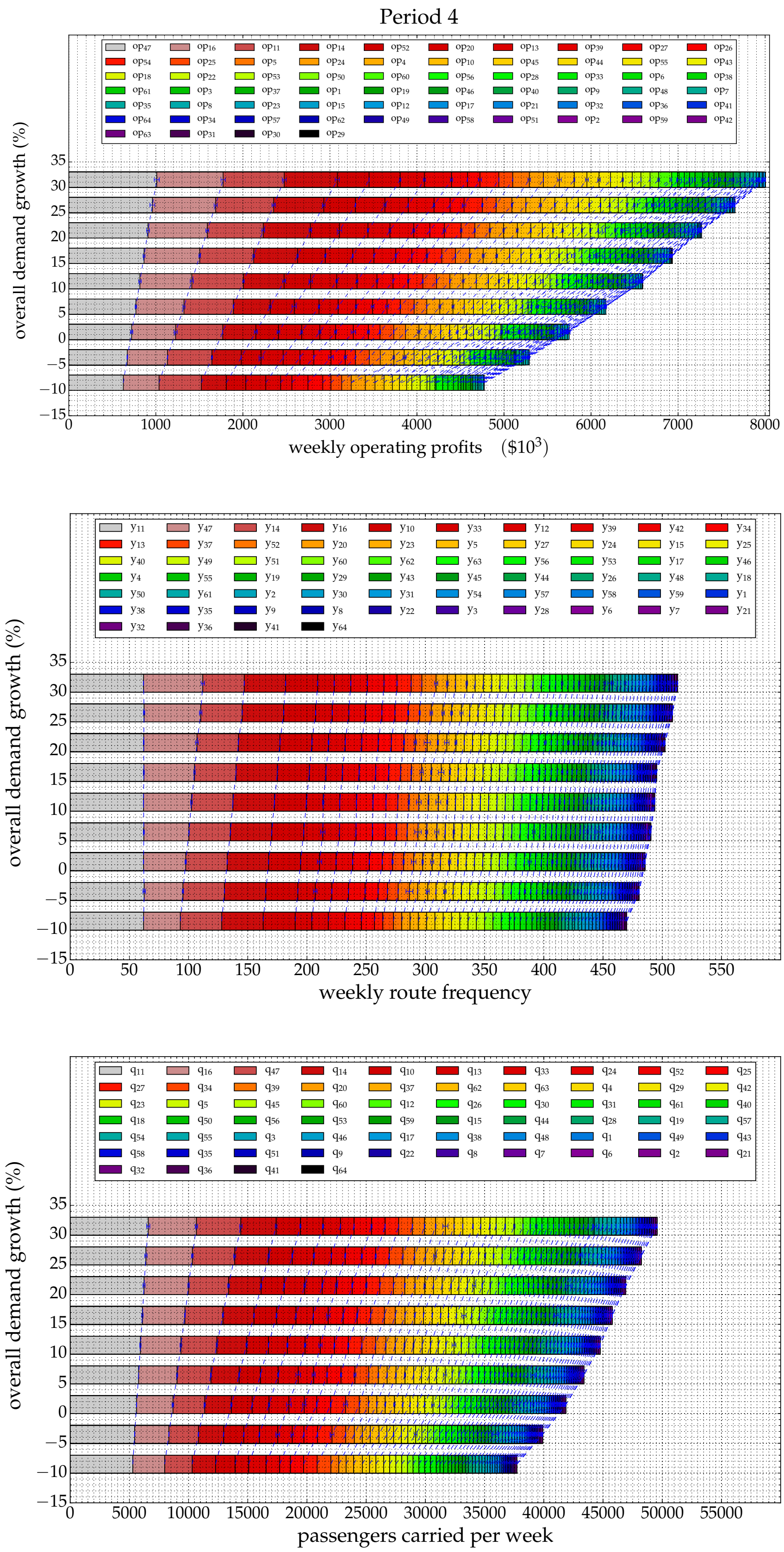


Figure 7.11: Average weekly operating profits, route frequency and passengers carried per scenario in year 4

7.5.1. Fleet evolution and their profitability performance

By analysing the actual number of passengers carried from 2015 onwards, it can be estimated that the KQ history of captured demand is found in between scenarios IMM and IML. The reader should notice that only the first 3 periods of the scenario tree can be visualized. This is due to the fact that periods 4 and 5 refer to future years 2018 and 2019, whose related information is still unavailable except for their demand forecasts.

Table 7.7: Actual KQ fleet evolution

		FY 0	FY 1	FY 2
	Operated Aircraft	2015	2016	2017
ADP	Boeing 777-200	4	0	0
	Boeing 787-800	6	9	9
	Boeing 737-800	10	9	9
	Embraer 170	3	1	1
	Embraer 190	15	14	14

Table 7.8: Recommended fleet evolution by ADP policy

		FY 0	FY 1	FY 2
	Operated Aircraft	2015	2016	2017
KQ	Boeing 777-200	4	2	0
	Boeing 787-800	6	9	7
	Boeing 737-800	10	10	10
	Embraer 170	3	2	0
	Embraer 190	15	15	15

Considering this demand evolution, Table 7.7 shows the fleet plan that the ADP adaptive policy would recommend under those conditions of growth. The advised fleet plan can then be compared to the actual fleet changes made by Kenya Airways. At first sight, it is appreciated that both fleet sizes and composition present a similar order of magnitude. In particular, the following conclusions can be drawn:

- **By 2015, the recommended fleet policy would have captured the need to dispose the B772s and invest in more Dreamliners, since these ones have proven to adjust better to the size of KQ network.** It is pointed out, however, that two B788s have been subleased to other airlines in 2017. This decision was imposed with the company turnaround strategy to reduce costs after several years of financial losses. In fact, those disposals turned into a subleasing contract with another African airline. This explains the reason for which Kenya Airways finds more profitable to dispose of two B788s rather than operating them. Evidently, this opportunities cannot be captured my the multi-period fleet planning model.

- **Initially, the ADP policy would have also advised to reduce the number of aircraft operating short-medium haul routes.** As can be observed, the ADP fleet plan recommends to have one aircraft less for all B788, E90 and E70 types. Interestingly, experts of Kenya Airways commented that they always tend to keep one aircraft of each type on the ground, thus supporting the validation of the ADP results. What is interesting to highlight is the different fleet composition existent between E70 and E90. While Kenya Airways eventually disposes of all E70s, the adaptive policy recommends to keep at least one for operating short and thin routes. This advice will be discussed in more detail shortly.

Once both fleet plans are identified, their corresponding impact on KQ operating profits is assessed. Table 7.9 shows the annual profits earned by operating the network with each of the fleet plans under the conditions of all possible scenarios. It must be highlighted that the bolded rows refer to the scenarios IMM and IML defining the boundaries of the actual historical demand growth experienced by KQ.

The first noticeable point is that in 2015 the ADP policy presents higher losses than the KQ fleet plan. This is because of the higher disposal costs entailed by the decision of changing the fleet for next period. In this sense, the KQ fleet undergoes less changes from 2015 to 2016. Nevertheless, the losses from the ADP policy can be understood as an initial investment, since its value is clearly offset by the profits earned within the next years. This is appreciated in the column of aggregated results, where the total performance of the KQ fleet plan appears as well. Indeed, it is very visible that for the following years the performance of Kenya Airways would have significantly improved if the recommendations of the ADP policy had been followed. In figures, the 3-year aggregated profits would have been bounded between \$ 41M and \$15M for the ADP policy, whereas the actual aggregated profits of KQ will remain between \$10M and \$-18M. Indeed, the ADP policy could achieve a profit gain between 300% and 180% with respect to the profit obtained with the real KQ plan.

Finally, It is interesting to highlight how well the model captures the reality: KQ 2017 press statements have accounted for a total operating profit of \$8M in 2016, while the model indicates \$7.6M for the same year under very similar demand conditions. Once again, this results prove the ADP algorithm as an effective

Table 7.9: Comparative analysis of the profitability of KQ fleet plan and ADP policy for all scenarios from year 2015 to 2017

		Annual operating,profits ($)			
	Scenario	FY 0	FY 1	FY 2	Aggregated
		2015	2016	2017	operating profits ($)
ADP policy	IHH	-32515267	46813513	67583740	81881986
	IHM	-32515267	46813513	54579935	68878181
	IHL	-32515267	46813513	28360863	42659109
	IMH	-32515267	31891386	54268219	53644338
	IMM	**-32515267**	**31891386**	**41656523**	**41032641**
	IML	**-32515267**	**31891386**	**15561791**	**14937909**
	ILH	-32515267	3054794	31758341	2297868
	ILM	-32515267	3054794	16427055	-13033418
	ILL	-32515267	3054794	-14246537	-43707010
KQ fleet plan	IHH	-27315267	23980902	62402477	59068112
	IHM	-27315267	23980902	46239366	42905000
	IHL	-27315267	23980902	17795945	14461580
	IMH	-27315267	7610701	46239366	26534800
	IMM	**-27315267**	**7610701**	**30077222**	**10372656**
	IML	**-27315267**	**7610701**	**1341116**	**-18363449**
	ILH	-27315267	-25552496	18661209	-34206553
	ILM	-27315267	-25552496	2206381	-50661382
	ILL	-27315267	-25552496	-32293261	-85161024

Table 7.10: Potential profit gains obtained with the ADP policy with respect to KQ fleet plan

		Annual operating profit gain (%)			
	Scenario	FY 0	FY 1	FY 2	Aggregated
		2015	2016	2017	profit gain (%)
ADP/KQ	IHH	-19.04%	95.21%	8.30%	38.62%
	IHM	-19.04%	95.21%	18.04%	60.54%
	IHL	-19.04%	95.21%	59.37%	194.98%
	IMH	-19.04%	319.03%	17.36%	102.17%
	IMM	**-19.04%**	**319.03%**	**38.50%**	**295.58%**
	IML	**-19.04%**	**319.03%**	**1060.36%**	**181.35%**
	ILH	-19.04%	111.95%	70.18%	106.72%
	ILM	-19.04%	111.95%	644.52%	74.27%
	ILL	-19.04%	111.95%	55.88%	48.68%

method capable of providing meaningful insights to fleet planning.

7.5.2. Frequency-network analysis

In order to understand the main differences between the ADP and actual fleet plans presented, it is very insightful to analyse how the KQ network is operated at a frequency level. This helps us understand the reasons upon which the ADP algorithm bases its fleet decisions.

The first key aspect to examine is whether the ADP-based tool opens the same routes that Kenya Airways wanted to open according to its 5-year expansion plan. Table 7.11 summarizes the network actions made by both KQ and the ADP algorithm until the present year 2017:

Table 7.11: Comparison between network expansion decisions made by the ADP algorithm and Kenya Airways

ID Route	ADP	ADP decisions	KQ decisions	Distance (km)
2016	7	**Open**	**Not operated**	**3064**
	21	Not operated	Not operated	2400
	22	Open	Open	3480
	41	Not operated	Not operated	3200
	55	Open	Open	3440
	56	Open	Open	5720
2017	1	Open	Open	2811
	6	**Open**	**Not operated**	**858**
	64	Not operated	Not operated	12002

From the comparative analysis, it is inferred that the ADP algorithm agrees with the majority of the expansions decisions made by the KQ network planning department. It can also be appreciated that the expansion strategy of Kenya Airways evolved slightly differently as initially planned in 2014. Over the years, some routes were not considered profitable enough and were not operated. The ADP policy draws mainly the same conclusions except for two very few exceptions: routes with ID 7 and 6, which correspond to a medium and short haul routes respectively. Yet, these differences can be explained with Figure 7.12, where ADP route frequencies are benchmarked against KQ frequencies currently operated.

In Figure 7.12, the bars represent the flight frequencies that the ADP policy assigns to each route per aircraft type distinguished by color. Likewise, dark blue dots refer to the actual frequency levels that Kenya Airways is providing for each route, whereas orange dots represent the minimum flight frequencies required to maintain the targeted market share for each route.

Overall, it is concluded that the correlation between ADP and KQ actual route frequencies is satisfactory: the ADP policy is able to provide realistic levels of route frequencies and even better, assign aircraft types in a way that makes sense. As expected beforehand, the ADP model tends to slightly reduce the amount of flights per each route. This general tendency results from the assumption previously made: the amount of captured demand is supposed to be independent of the level of frequency offered. Evidently, this hypothesis is far from reality and sometimes makes the model assign bigger aircraft to certain routes, thus carrying more passengers in less flights. Fortunately, this trend is not really accentuated so the aircraft type assignments carried out by the ADP policy are very similar to the ones made by Kenya Airways. Figure D.12 in Appendix D shows how wide-body aircraft are assigned to long-haul routes, while single-aisle aircraft cover short-medium haul routes.

In this situation, the frequency-network analysis helps identify as well the main reasons by which some fleet decisions and route openings are made. For instance, it explains why the ADP policy opens routes 6 and 7 and KQ has not. By analysing in detail these routes in Figure 7.12, it can be concluded that the ADP algorithm decides to operate these routes with bigger aircraft such as B738 or even a B788. From a mathematical perspective this makes sense since with one single flight, Kenya Airways would be able to earn more profits from those routes. Nevertheless, this is hardly possible in real life, since demand would most likely decrease dramatically provided that one single frequency was offered wit a big aircraft. This fact enhances the need to adopt a critical perspective regarding the fleet advised by the ADP policy. Given this general preference for bigger aircraft, it should be assessed in more detail whether the quantity of wide-body aircraft is slightly overestimated.

Furthermore, other fleet-related decisions can also be explained. In fact, the ADP policy decides to maintain at least one E70 so as to assign it to short haul routes with very low demand. These type of routes are considered as *thin* routes by Kenya Airways and they are actually fully operated by E90s. Despite this decision, the ADP algorithm concludes that is more profitable to invest in smaller aircraft to reduce operating costs and increase the load factors. Indeed, the load factor characteristic of this type of routes is around 40%.

7.5.3. Expert survey

As a complementary activity to validate the practicability of the ADP support tool within the airline industry, several experts in network planning were surveyed. Based on various interviews carried out on-site at Kenya Airways as well as Transavia, the following conclusions can be drawn:

Regarding the application of adaptive policies to strategic fleet planning,

- Both Kenya Airways and Transavia are conscious of the fact that demand uncertainty plays an essential role in their profitability performance. However, their best practice to hedge against uncertainty is to plan their future fleet according to deterministic forecasts and certain sensitivity analysis of demand. Spreadsheets are the preferred technique to carry out this type of analysis. Therefore, it consists more in a trial-and-error process rather than a sophisticated mathematical optimisation.

- Despite being conscious of the importance of accounting for demand uncertainty, the truth is that nobody is studying new techniques to improve their current best practices. Indeed, uncertainty becomes an unpleasant topic for discussion since it challenges one of the most entrenched traditional methods within the company. In this context, the concept of adaptive policy is not very familiar; however, the scenario tree format is considered an intuitive way to model demand uncertainty and represent future possible scenarios.

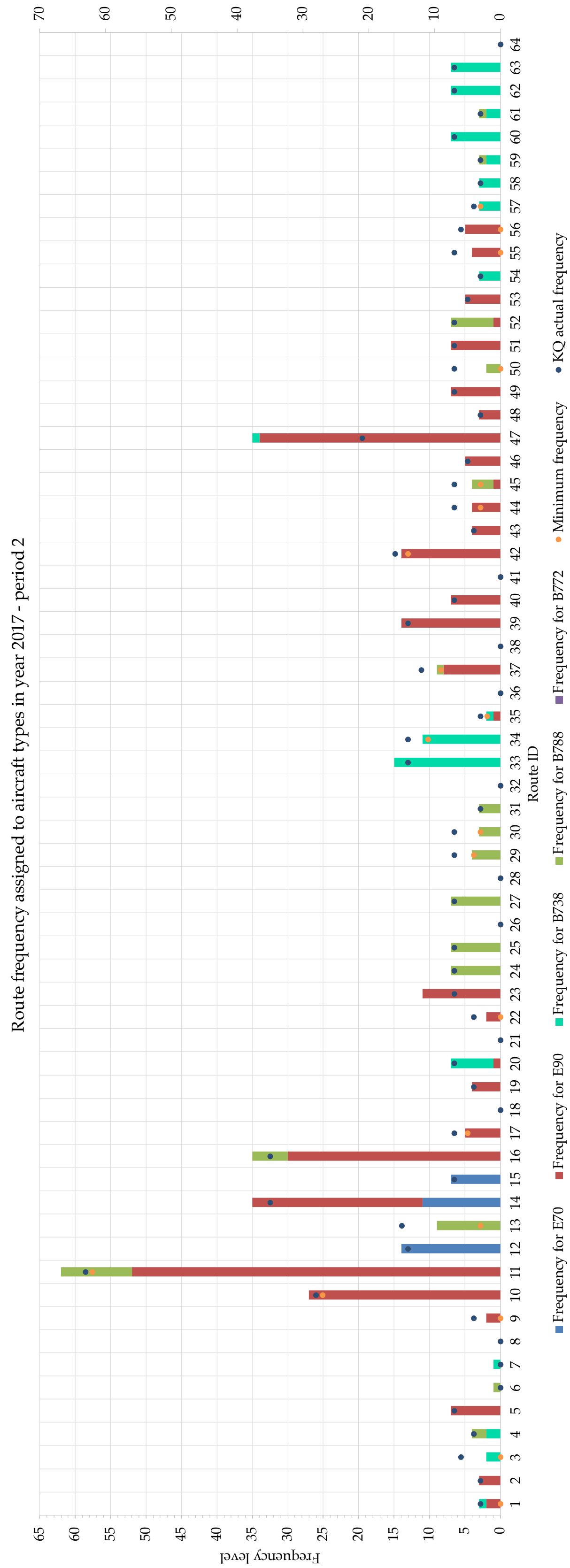


Figure 7.12: Route frequency assigned to each aircraft type in period 2

Regarding the multi-period adaptive fleet planning model:

- Experts generally agree with the mathematical formulation of the multi-period adaptive fleet planning model. However, expert feedback varies depending on the conditions faced by each airline.

- Kenya Airways experts pointed out as important weakness the fact that the model does not consider fuel price uncertainty, which has sometimes amounted to the 40% of their operating costs. In fact, the impact of stochastic fuel prices was assigned the same level of importance as demand uncertainty. Surprisingly, the concept of minimum frequency was deemed appropriate to model market interactions.

- Transavia highlighted the need to include a new model constraint to take into account the maximum number of slots that an airline owns in a saturated airport. Indeed, this cannot be dismissed in those situations in which an airline operates at a saturated airport (e.g. Heathrow and Schiphol) where its flight frequencies are constrained. Furthermore, the planning horizon at Transavia is much shorter than the one of Kenya Airways since the former deal with a more seasonal demand. Given this fact, periods were recommended to be shortened in six months and average prices were required to be dynamic throughout the periods.

Regarding the practicability of the ADP support tool:

- All fleet planners interviewed highlighted the benefits of having an interface that allowed the introduction of model parameters via Excel. As for the ADP results, fleet maps and operational maps were considered a quick method to pinpoint the general trends in fleet planning.

- In particular, Transavia fleet planners acknowledged the potential of the ADP results. Nevertheless, it was questioned whether the benefits of applying an adaptive policy would paid off the time invested in mastering such a complex model. Indeed, the model characteristics were seen as too complex in comparison to the simplicity of Transavia business model: Transavia has a stagnated market due to the current slot saturation of its main airports, and yet its fleet is mainly homogeneous.

- The feedback of Kenya Airways contrasts sharply with the vision of the low-cost carrier. The possibility of having a general overview of the impact of their fleet decisions was clearly welcomed by the Head of Network and the Head of Strategy and Performance. Indeed, fleet maps and operational maps proved to be very suitable for executives. In a very short time, they were able to identify the main trends in fleet and network planning. Even better, they agreed with many of the recommendations inferred from the adaptive policy. For instance, they recognized that they were analysing the possibility to acquire new smaller aircraft such as the E70 to cover thinner routes more efficiently. Apart from this, they also agreed with the recommendation of operating new B737 MAX in certain long-haul routes, where demand is not sufficient enough to afford wide-body aircraft flights. Finally, KQ fleet planners were also thinking about increasing the frequencies of some specific routes where the ADP policy was already advising to increase the frequency. All in all, it was concluded that the ADP support tool was very capable to provide valuable information to support strategic decisions.

7.6. Conclusions

The present case study has provided valuable insights upon which the following conclusions can be based:

- The computational times of this case study make clear that a problem of such dimensions could not have been solved without the previous enhancement of the algorithm performance. In fact, a total of 15000 iterations were needed to reach convergence and coherent results. In contrast to the previous results obtained with the baseline algorithm, the enhanced algorithm allowed to solve this problem within a total CPU time of 10 hours: 4 hours to run the ADP loop and 6 hours to generate an sufficient number of meaningful observations to train the kernel regression.

- The enhanced ADP algorithm presents an iterative learning behavior which is comparable to the proof of concept except for the higher levels of noise experienced. Therefore, this case study requires longer exploration times to start converging: while the previous small test case started to stabilize by the first 500 iterations, the KQ case study does it for the first 2000 iterations. This is explained by the larger dimensions of the problem itself.

- The case study has also identified the adaptive fleet policy as the most robust method to take decisions under uncertainty, thereby outperforming the profitability performance of the optimal fleet plan for the most-likely scenario. Indeed, the worsening impact of the most-likely fleet plan becomes much more evident the more the scenarios differ from the most-likely scenario. This is particularly true in those extreme scenarios where the most-likely solution can entail weekly operating losses by more of 30% with respect to the deterministic upper bound. In contrast, the recommendations of the adaptive fleet policy can help reduce these losses by more than 50% with respect to the most-likely case.

- Nevertheless, the expected profit gain achieved with the adaptive policy (20%) is lower than the one presented by the stochastic case of the proof of concept (50%). This advantage reduction can be explained by the level of growth disparity existing between the extreme scenarios and the most-likely scenario. The greater the variations are in terms of demand outcomes, the better the adaptive policy will perform against the most-likely fleet. Indeed, this case study featured demand growth outcomes whose difference was less accentuated when compared to the ones of the proof of concept.

- A 5-stage scenario tree leads to adaptive fleet policies which are more difficult to interpret. However, it is still possible to identify the path dependency by just looking at the adaptive fleet policy recommended. In this context, fleet and operational maps offer the possibility to extract general trends at first sight, thus being well received by decision makers.

- Finally, the results obtained from the validation analysis highlight the capability of the ADP algorithm to capture realistic and meaningful trends back in 2015. Despite not considering the interactions between captured demand and frequency offered nor fuel or competition uncertainty, the ADP support tool was able to provide a future fleet policy whose profitability performance proved to be notably better than the actual fleet plan followed by Kenya Airways. Even better, the ADP tool forecasted extremely well the actual operating profits that KQ achieved in 2016. Apart from this, the network-frequency analysis presented a very reasonable correlation between the recommended and real frequency levels currently operated by KQ. Finally, it is very encouraging to see how many of the recommendations provided by the ADP support tool agree with the airline future strategy.

All in all, this case study has become a proof that the ADP support tool can adjust well to reality while providing meaningful adaptive policies to hedge against uncertainty, which indeed could make a difference in airline fleet planning.

8

Conclusions and recommendations

In this chapter, the main conclusions drawn from this MSc thesis are presented in Section 8.1 , followed by a set of recommendations in Section 8.2, which define the guidelines for future research.

8.1. Final conclusions

The inherent uncertainty of the fleet planning problem has hindered the emergence of sophisticated models to support airlines with strategic decision-making. For many decades, airlines have been applying similar top-down deterministic approaches when planning their fleet development. In this context, the objective of this MSc thesis was to contribute to the development of adaptive policies in the context of airline fleet planning under demand uncertainty by (a) modelling and solving with Approximate Dynamic Programming a multi-period adaptive fleet planning problem that integrates stochastic demand and by (b) detecting useful signposts for fleet planners.

With the objective of solving the multi-period adaptive fleet planning problem by means of approximate dynamic programming, the developed methodology was split into two parts. Firstly, the problem was modelled as a dynamic program thanks to the suitable application of a state-space modelling framework. This consisted of a state vector, decision vector, random variable, transition function, contribution function and value function together with a set of constraints defining the feasible space of actions. Next, the objective function representing the maximisation of the airline operating profits was reformulated in the form of Bellman equation. Taking advantage of the dynamic formulation of the problem, an algorithm based on approximate dynamic programming was chosen and implemented following its main building blocks.

Amongst all the parts of the ADP algorithm, the development of an effective value function approximation strategy became crucial for the successful implementation of a decision support tool. To this end, several approximation methods were benchmarked, leading to the conclusion that nonparametric regressions are the most suitable strategy to approximate the value function of the multi-period adaptive fleet planning problem. In particular, a separable Gaussian kernel regression was applied to obtain smooth approximations across all non observed scenarios. As for its initialisation, an initial set of approximated observations had to be computed to train the kernel regression with meaningful problem information.

The full multi-period adaptive fleet planning problem was optimised by decomposing it into subproblems. With the aim of taking full advantage of the computational power of commercial optimisers as well as the two-stage overlapping structure existing between subsequent subproblems, each maximisation subproblem was simultaneously decomposed into two independent parts A and B. Part A was solved directly with Gurobi and referred to the calculation of operations-related actions once previous fleet decisions had been taken. Likewise, part B corresponded to the problem of selecting the fleet-related decisions that would impact future operations. In this case, an epsilon-greedy subroutine based on simulated annealing was developed to let the algorithm explore the state space and choose other states with high potential.

Apart from this, the ADP algorithm required several iterations and sample realizations of the random demand growth to learn iteratively better policies for the scenario tree. This was carried out by means of Monte Carlo sampling, which helped extract random values from a known distribution of demand growth. Depending on the discrete or continuous nature of the known distribution, we could either have a finite or infinite number of possible random outcomes, thus having a finite or infinite number of possible scenario tree branches. While a continuous distribution could match better the actual demand uncertainty by slowing down the algorithm convergence, a finite distribution allowed for faster learning rates of the ADP algorithm at the expense of less modelling accuracy. Therefore, a trade-off between these two aspects was made, leading to the selection of a discrete random variable so as to facilitate the convergence of the algorithm.

The developed ADP algorithm was denoted as the *baseline algorithm* that would be verified in both deterministic and stochastic experimental settings. For the deterministic version of the multi-period fleet planning problem, the ADP results were compared to the optimal solutions obtained from solving the equivalent MILP problem with Gurobi. This comparative analysis led to the conclusion that the ADP results provided an accurate fit to the optimal values provided by Gurobi. Overall, the average optimality gap presented by ADP was of 0.2%, a magnitude which is highly comparable to the optimality gaps currently seen in ADP literature. This fact encouraged to further investigate kernel regression as an effective value function approximation strategy. In terms of CPU performance, Gurobi stood as the best technique to solve these small test cases: it took an average of 35s to provide optimality gaps of the same order of magnitude as the ones presented by the ADP algorithm within 120s. These results are coherent since the potential benefits of the ADP algorithm are expected to be noticeable in those problems with a vast number of decision variables and multiple stages. Indeed, it is in that context that the initial 2h-calculation of the kernel training dataset starts to make sense.

As for the stochastic version, the introduction of demand uncertainty as a stochastic process entailed several complexities. Indeed, the coupling of scenarios with common tree nodes made the update of the value function very unpredictable and difficult to track. In turn, this led to an increase in noise levels, thereby complicating the convergence of the ADP algorithm. Whereas only 80 iterations were required to achieve optimal or near-optimal solutions for the deterministic case, the stochastic version of the problem needed around 3000 iterations to reach convergence and provide a coherent policy. In terms of computational performance, running these 3000 iterations translated in a CPU time cost of 5h, without considering the 2h punctually invested in creating an initial set of observations to train the kernel regression. Thus, the sharp increase in computational time with respect to the deterministic case was evident and it was concluded that it had to be definitely improved in case of tackling more realistic problems.

Furthermore, solving the stochastic version of the multi-period fleet planning problem did not provide a single result, but an adaptive fleet policy: the best rule for making fleet and operational decisions in function of a certain demand and financial conditions happening in a given time period. In this context, it was concluded that the resulting fleet policy adjusted as much as possible to the optimal fleet plans corresponding to each deterministic scenario solved independently. Moreover, a very unique feature of the adaptive fleet policy was its capability of capturing the path dependency effect amongst different scenarios. Indeed, the adaptive policy assigned different fleets to states with the same level of demand, which were coherent with their previous demand history experienced. Therefore, the adaptive policy was able to simulate in a realistic way the decision-making process of airline fleet planning.

As for profitability performance, the adaptive policy was the most robust method for fleet planning: across the majority of scenarios it clearly excelled the profits resulting from the optimal fleet plan for the most-likely scenario. Evidently, the most-likely solution performed better in those scenarios similar to the most-likely scenario. Nevertheless, its performance started to stall towards differing scenarios. Indeed, its worst scores were found in the neighborhood of the extreme scenarios: For the most pessimistic scenarios the most-likely solution could imply weekly operating losses ranging between 3% and 6%, whereas in the most optimistic scenarios losses remained between 2% and 1%. In contrast, the adaptive fleet policy mitigated losses in all extreme scenarios, which never surpass the 1.1%. By analyzing the general expected performance, it was concluded that the adaptive policy could reduce by 50 % the losses entailed by the most-likely solution.

Apart from this, the verification analysis proved the stable behaviour of the ADP algorithm. By modifying the parameters of the baseline case within different intervals, it was confirmed that the kernel approximation strategy could still be trained reasonable well using the same dataset of observations. Furthermore, a sensitivity analysis helped understand better the influence of the most relevant parameters within ADP algorithm. On the one hand, higher stepsize values ($\alpha_{n-1} \approx 1$) provided the best results for the value iteration algorithm. On the other hand, the correct calibration of the control parameter λ became essential to guarantee the convergence of the ADP algorithm as well as the coherence of the adaptive policy. Indeed, the problem of value function overestimations represents a major challenge to tackle in follow-up research.

Likewise, the application of a continuous random variable normally distributed was experimented. This analysis reaffirmed the hypothesis that the continuously distributed Monte Carlo sampling would provoke slow convergence and higher instability for the algorithm. In any case, it should not be overlooked that sampling a continuous distribution provides much more realistic scenarios, thus embracing a wide range of possible outcomes. Nevertheless, the size of the scenario tree became unmanageable to assess it thoroughly.

The conclusions obtained with the proof of concept unveiled the advantage of applying adaptive fleet policies and most importantly, proved approximate dynamic programming as an effective method to calculate them. Nevertheless, the corresponding computational times warned about the need to improve the algorithm efficiency. To do so, a decision tree analysis was carried out to pinpoint the major sources of CPU waste and define five major levels to improve the algorithm accordingly. These consisted in (1) providing a MIP start to Gurobi, (2) relaxing the optimality gaps when generating the training dataset and (3) solving part A of the 1-FPP problem, (4) avoiding unnecessary recalculations with Gurobi in part A of the 1-FPP problem as well as (5) reducing the state space explored to accelerate the subroutine in part B. In this sense, the computational times of the subsequent case study carried at Kenya Airways made clear that a problem of such dimensions could not have been solved without the previous enhancement of the algorithm performance. In fact, a total of 15000 iterations were required to reach convergence and coherent results. In contrast to the previous results obtained with the baseline algorithm, the enhanced algorithm allowed to solve this problem of realistic dimensions within a total CPU time of 10 hours: 4 hours to run the ADP loop and 6 hours to generate a sufficient number of meaningful observations to train the kernel regression.

The extended case study carried out at Kenya Airways provided valuable insights to validate the ADP support tool in the air transport industry. The ADP tool was capable of tackling successfully a 5-period fleet planning problem with 5 aircraft types and 64 routes aggregated per market growth regions. The proposed case study consisted in moving backwards in time to year 2015 with the objective of reproducing a 5-year expansion plan formerly defined by Kenya Airways.

First of all, the adaptive fleet policy was once again identified as the most robust method to take decisions under uncertainty. Indeed, the worsening impact of the most-likely fleet plan became much more evident in extreme scenarios, where it entailed weekly operating losses by more than 30% with respect to the deterministic upper bound. In contrast, the recommendations of the adaptive fleet policy could help reduce profit losses by more than 50% with respect to the most-likely case. Moreover, a 5-stage scenario tree provided adaptive fleet policies which were more difficult to interpret. However, it was still possible to identify the path dependency by just looking at the adaptive fleet policy recommended. In that context, fleet and operational maps offered the possibility to extract general trends at first sight, thus being well understood and received by decision makers.

To conclude, the results obtained from the validation analysis enhanced the capability of the ADP algorithm to capture realistic and meaningful trends back in 2015. Despite not considering the interactions between captured demand and frequency offered nor fuel or competition uncertainty, the ADP support tool was able to provide a future fleet policy whose profitability performance proved to be notably better than the actual fleet plan followed by Kenya Airways from 2015 onwards. Even better, the ADP tool forecasted extremely well the actual operating profits that KQ achieved in 2016. Apart from this, the network-frequency analysis presented a very reasonable correlation between the recommended and real frequency levels currently operated by KQ. Finally, it was confirmed that many of the recommendations provided by the ADP support tool back in 2015 agreed with the current airline strategy. All in all, the case study became a proof that the ADP support tool can adjust well to reality while providing meaningful adaptive policies to hedge against uncertainty, which indeed could make a difference in airline fleet planning.
For all the above mentioned conclusions, it can finally be stated that the research objective of this MSc thesis has been satisfactorily met.

8.2. Final recommendations

While carrying out this MSc thesis, several recommendations could be identified to follow up with this work. These are presented according to different areas of research.

8.2.1. Modelling

When it comes to modelling, many proposals can be made. Nevertheless, it should be taken into account that any extension to the multi-period fleet planning problem will most likely entail additional computational performance. It is for this reason that the combination of both modelling and algorithm development is highly encouraged.

With this in mind, it seems quite fast to implement some of the suggestions put forward by the surveyed airlines. For instance, Transavia highlighted the need to include a new model constraint to take into account the maximum number of slots that an airline owns in a saturated airport. Indeed, this cannot be dismissed in those situations in which an airline operates at a saturated airport (e.g. Heathrow and Schiphol) where its flight frequencies are constrained. This requirement could be easily formulated as

$$\sum_{r \in \mathcal{R}} \sum_{i \in \mathcal{I}} y_{tir} \quad \leq \quad Y_t^{Max} \quad \forall t \in \mathcal{T} \tag{8.1}$$

where Y_t^{Max} refers to the maximum number of slots available in a given time period. Again, the impact of this constraint on the optimisation of the problem should be assessed. Nevertheless, it is not expected to introduce major computational challenges.

Furthermore, it may happen that demand is so seasonal for certain airlines that it could not be dismissed. In this case, it is recommended to shorten the periods into at least six months. Likewise, average prices could be modelled dynamically throughout different time intervals with the objective of increasing the model accuracy. Another interesting option to explore would be the introduction of a cost factor penalising the heterogeneity of the fleet: the more fleet types owned, the higher its value would be so as to take into account costs associated to maintenance and crew trainings. Indeed, a current best practice amongst airlines is to move towards leaner fleets with less aircraft types so as to minimise costs.

Apart from this, it must be reminded that Kenya Airways experts pointed out as important weakness the fact that the model does not consider fuel price uncertainty, which has sometimes amounted to the 40% of their operating costs. In fact, they considered the impact of stochastic fuel prices as much important as the effects of demand uncertainty. Therefore, a very interesting research field would be to model the volatility of fuel prices, while providing recommendations on hedging measures to minimise the costs entailed by this uncertainty. One reasonable way to initially tackle this problem would be to study the independent effects of fuel price uncertainty. Having then understood the main essence of this problem, it would be ideal to combine both demand and fuel price uncertainties. However, this would challenge the way of representing the global scenario tree as well as the approximation strategy for the corresponding value functions.

Another important factor to study would be the effect of competition uncertainty, which has mostly been modelled with game theory in literature. Nevertheless, competition could be modelled from the perspective of a single airline in a more simple and useful way, which would allow the integration of competitive elements into fleet planning models. Instead of using game theory, the impact of competition on fleet plans could be modelled as the uncertainty entailed by the probability of other airline entering into a market. Indeed, an airline may invest in its fleet expansion so as to exploit a promising market with high growth rates. Nevertheless, any optimistic forecasted results may increase the probability of other airlines entering into the same market, thereby leading to a fleet surplus and its corresponding losses. This new idea would still capture the relevant effects of competition in fleet planning while keeping the model dimensions within acceptable dimensional limits. Yet surprisingly, the majority of surveyed experts considered that the concept of minimum frequency was deemed appropriate to model market interactions.

In this line, another important recommendation could be given: it would be very meaningful to model the elasticity existing between the demand captured by an airline and the level of frequencies provided. Indeed, the absence of this trait becomes a notable weakness when it comes to plan network frequencies: by not considering demand elasticity with respect to frequency, the ADP algorithm tends to select larger aircraft to carry more passengers while minimising frequencies and operating costs. From a mathematical perspective this seems logical, but it does not clearly adjust to reality.

8.2.2. Approximate Dynamic Programming algorithm

Approximate Dynamic Programming consists in a vast emerging field offering plenty of opportunities for research. Even though kernel regressions have proven suitable to approximate the value function of the multi-period fleet planning problem, the truth is that neural network presents a potential impossible to overlook: some applications have already benefited from their highly flexible framework capable of estimating completely unknown functions. Encouraged by the good results initially obtained with approximate dynamic programming, a natural choice would be to gain more expertise in the domain of statistical learning and more particularly, neural networks applied to the multi-period fleet planning problem.

Nevertheless, the field of kernel regressions also faces many challenges ahead. While the applied dynamic control of kernel bandwidth has proven effective for a maximum number of six periods, it is evident that it starts to present some limitations for a higher number of stages: observations might be so dispersed from the queried state that it is useless to incorporate them into the kernel weighted sum. A solution to this limitation would be to previously estimate more approximated observations in other non-observed scenarios so as to reduce point dispersion. Indeed, this is an area for further research since it would be ideal to determine a correlation between the number and position of required observations and the dimensions of the scenario tree. In any case, follow-up research could revolve around the introduction of a dispersion factor that would help determine an optimal training dataset in terms of number and location of observations. Apart from this, parallel kernel regression across observed scenarios could also be studied. Besides, another important point to be addressed would be to find a general rule to completely mitigate the adverse effect of value function overestimations. What is more, it would be very insightful to determine the maximum values of the optimality gaps for which the ADP algorithm could provide coherent results.

Finally, it is always beneficial to improve as much as possible the efficiency of the ADP algorithm so as to tackle even larger networks and fleets. For instance, the next step could be the introduction of dynamic control criteria to automatically stop the ADP loop once an acceptable level of stability has been reached. Another possibility could be to optimise the exploration versus exploitation problem by means of defining a general rule describing the best trade-off between exploration and exploitation phases. Finally, it would be very insightful to determine the limits of the optimality gap relaxation interval, for which the ADP algorithm provides stable and coherent adaptive policies. This knowledge could help speed up the process of creating a training dataset of observations.

8.2.3. Validation

One of the last proposed recommendations goes in line with the validation of the current ADP model but it is only applicable to 5 years from now. The case study solved in this MSc thesis was intended to reproduce past decisions until the present. Nevertheless, it would have been even better to work with older data, which has been very difficult to find for this MSc thesis. In light of this, it would be highly interesting to carry out a similar validation analysis in 2022 so as to compare the validity of the adaptive policy obtained now.

To conclude, it must be admitted that further work needs to be done in terms of signposts. If these were further researched, the solution of wide scenario trees could pave the way for the development of more advanced analytics tools capable of defining preventive, corrective and even predictive measures to steer strategic plans towards success.

A

Appendix for kernel regressions

This is an appendix for section 4.4.1 in chapter 4, where the performance of kernel regression will be discussed in more detail.

A.1. Verification of kernel regression performance

To verify the correct performance of the designed kernel regression tuned with $G = 3$, it is worth to analyse the correspondence between the estimated value function obtained with kernel regression, and the real approximated value that would be obtained if the value function was initially observed with $\lambda(T-t)\,\widetilde{C}_t^*(S_t)$. Figures 4.15 to 4.16 show the effective correlation obtained for nodes not belonging to the extremes or central scenarios in different periods. For the sake of clarity, it should be reminded that the interval notation is $\{0,1,...,T-1\}$. Therefore, even though we are referring to stage 5, the scenario tree considered has 6 periods and 0 refers to the initial state I. The blue line represents a 1:1 bisection while the green dots refer to the correspondence between the value function estimations obtained for different queried states S_t^*. In this way, it can be appreciated by how much the non-parametric observations differ from the observed value function using $\lambda(T-t)\,\widetilde{C}_t^*(S_t)$. While a good correspondence exists throughout all time stages, the impact of dispersed observations is noticeable for the later stages 4 and 5: there are kernel value function approximations differing significantly from correlation 1:1 since either they are overestimated or underestimated. In fact, maximum kernel relative errors can amount to 15% respect the observed value in stage 5. Nevertheless, in the same stage correlation errors are strongly mitigated up to 2% for the states with higher value functions, which are indeed the best candidates for exploration and exploitation during the ADP algorithm. In other words, there is a low probability that high correlation errors will affect significantly the ADP algorithm since they will seldom be selected for exploration. Furthermore, relative errors of 2% are considered admissible since it should be reminded that all observations feeding the kernel regressions in this verification test are initial approximated observations and not real observations of the value function. Indeed, these errors are expected to reduce along the subsequent ADP iterations. This is due to the fact that more algorithm iterations will provide more and better approximated observations, which will help improve the quality of kernel regression. What is important to infer from this correlation test is that kernel regression captures the problem structure effectively, thereby allowing the successful identification of the most promising states for the fleet planning problem. Lastly, Figures A.5 to A.6 show the great importance of kernel bandwidth: an unaware selection of $G = 1000$ may lead to wrong correlations unable to capture the essential information of the problem.

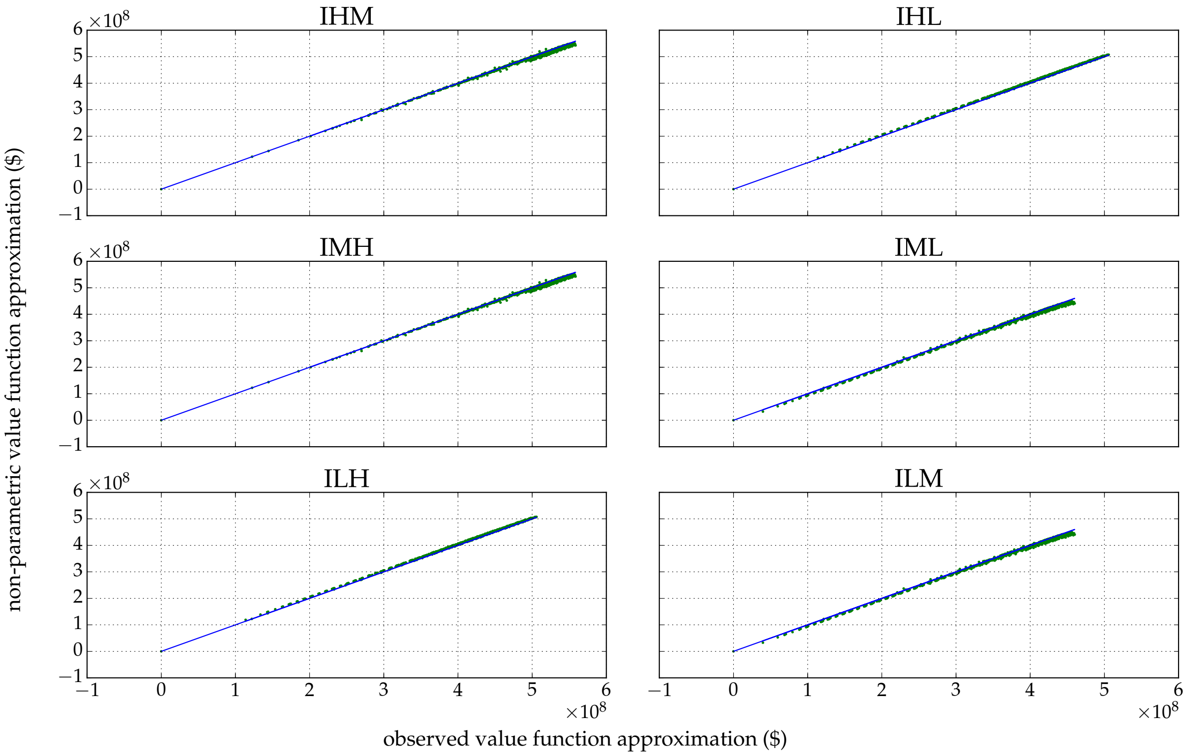


Figure A.1: Effective value function correspondence at stage 2 with $G = 3$

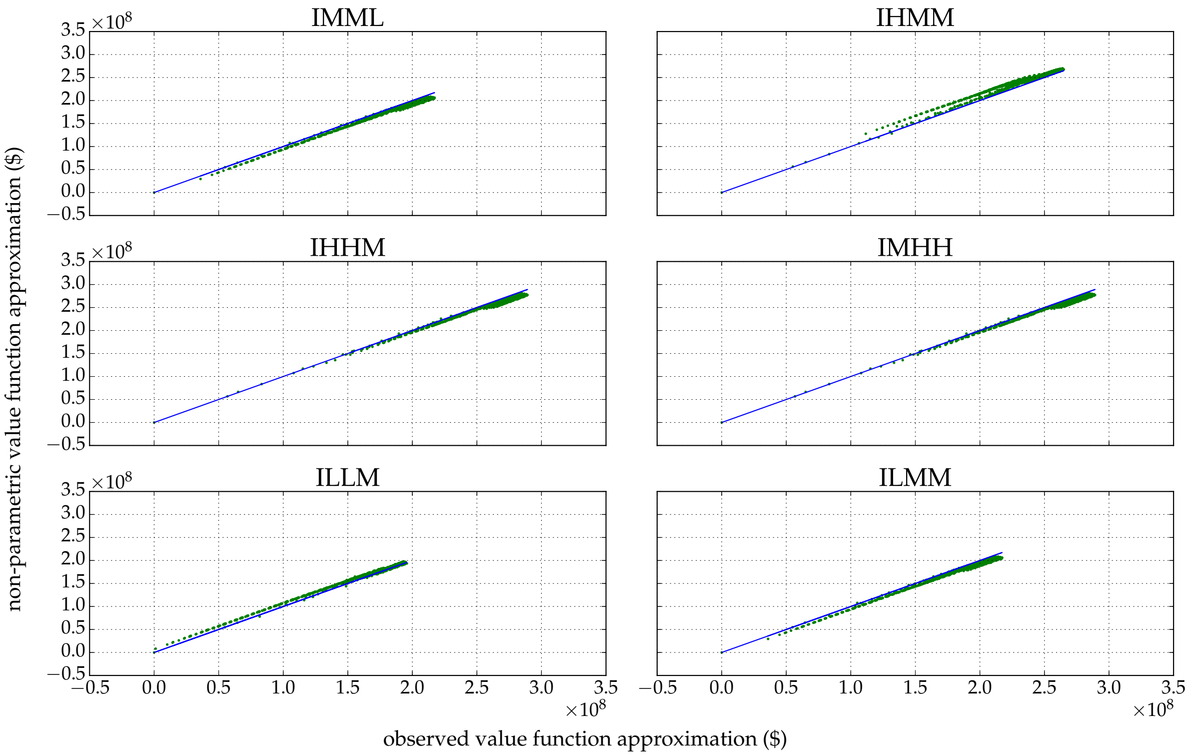


Figure A.2: Effective value function correspondence at stage 3 with $G = 3$

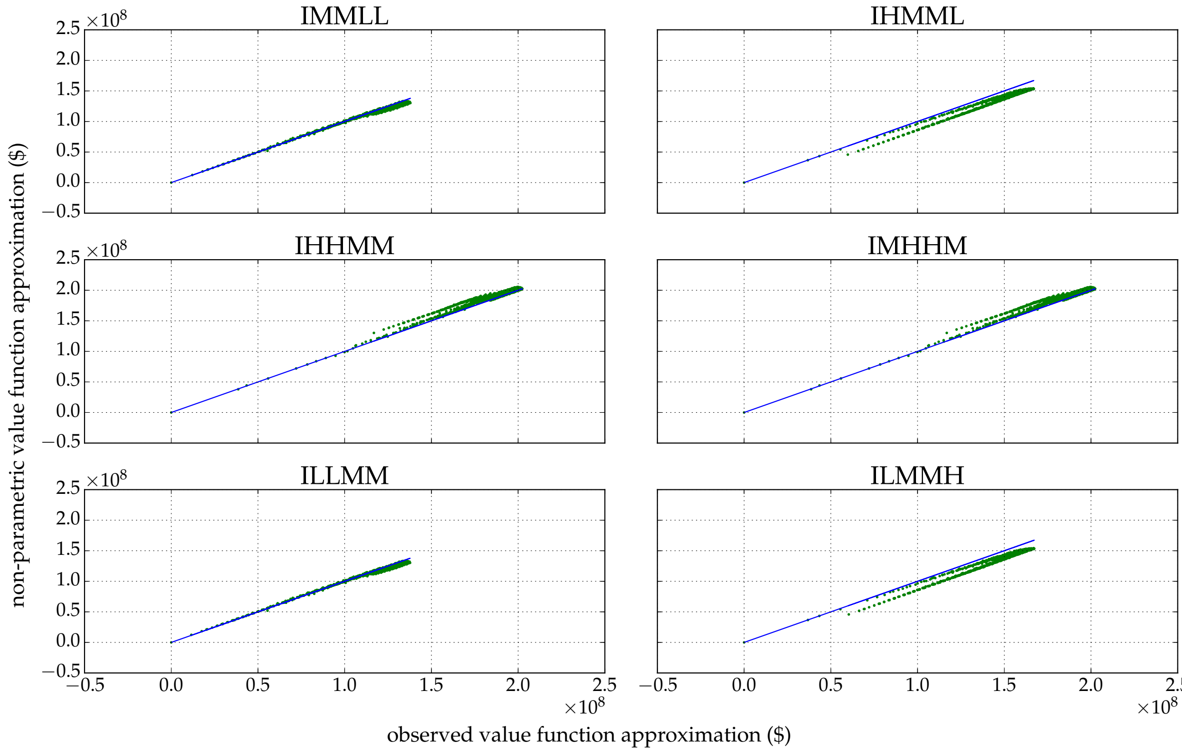


Figure A.3: Effective value function correspondence at stage 4 with $G = 3$

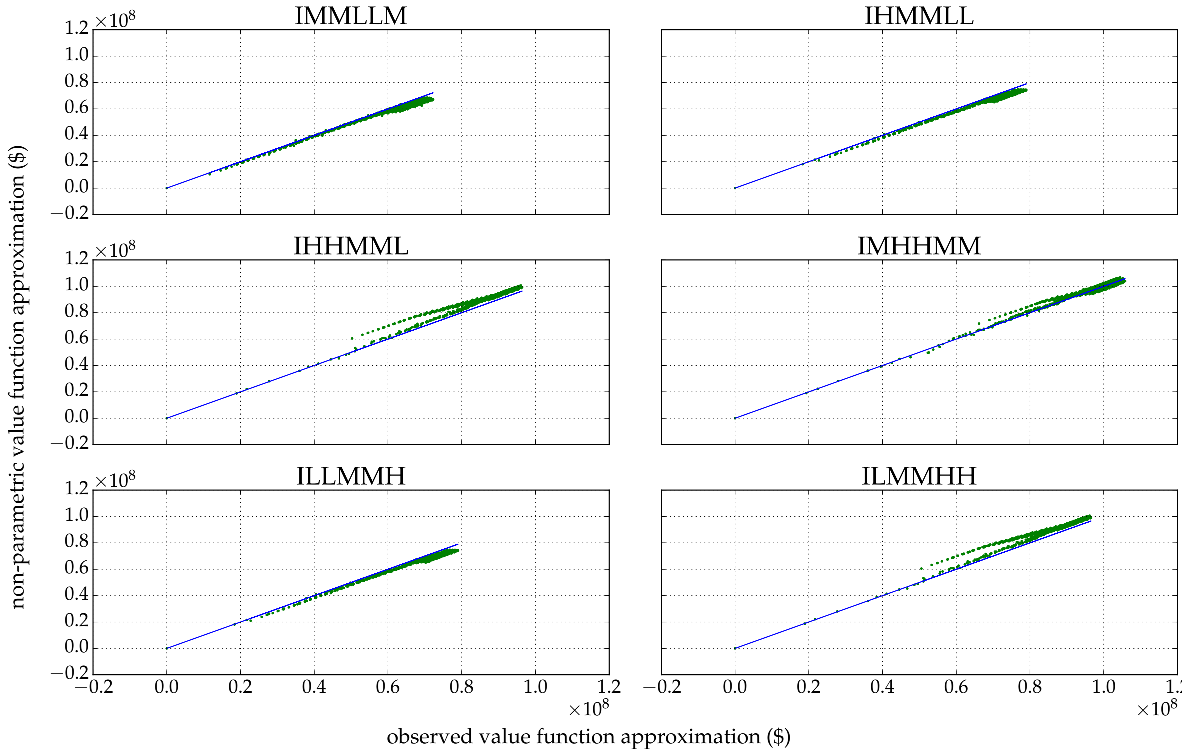


Figure A.4: Effective value function correspondence at stage 5 with $G = 3$

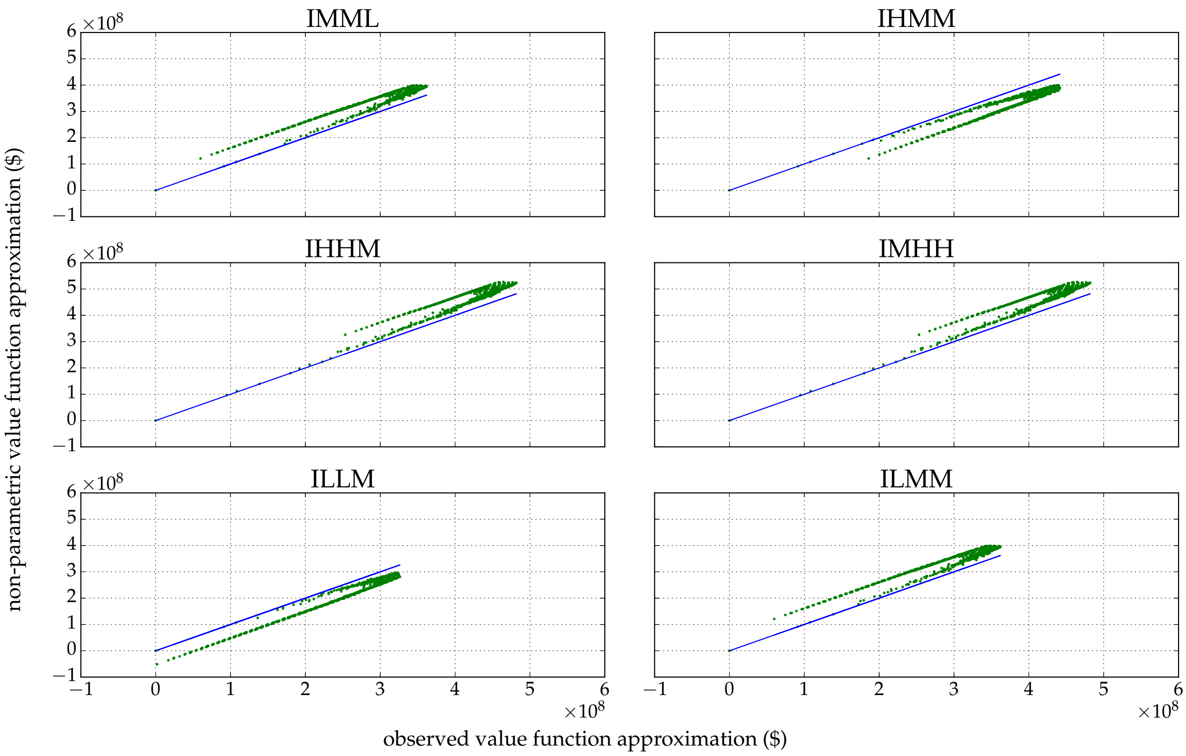


Figure A.5: Unstable value function correspondence at stage 4 with $G = 1000$

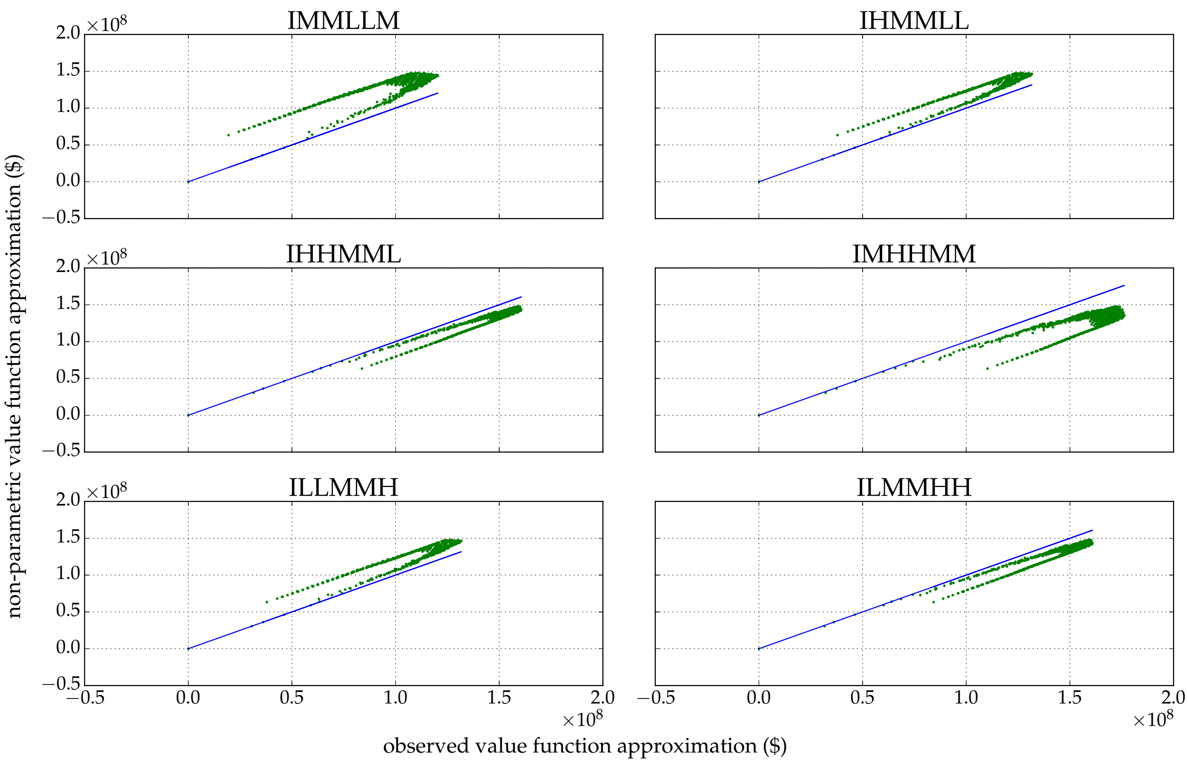


Figure A.6: Unstable value function correspondence at stage 5 with $G = 1000$

B

Appendix for deterministic experiments

This is an appendix for section 5.2 in chapter 5. It presents the optimal fleet plans for each determinisitc scenario solved independently as well as the behaviour of the ADP algorithm observed when solving non-observed scenarios.

B.1. Optimal fleet plans for each independent scenario

Table B.1: Optimal fleet plans for each deterministic scenario solved independently

					Optimal fleet for deterministic scenarios			
ID	Stage	Scenario	Total probability	Demand variation	year 0	year 1	year 2	year 3
0	3	IHHH	2.70%	52.1%	0/10/8	0/10/12	0/13/12	0/16/12
1	3	IHHM	4.50%	38.9%	0/10/8	0/10/12	0/13/12	0/14/12
2	3	IHHL	1.80%	25.6%	0/10/8	0/10/12	0/12/12	0/12/12
3	3	IHMH	4.50%	38.9%	0/10/8	0/10/12	0/11/12	0/13/13
4	3	IHMM	7.50%	26.8%	0/10/8	0/11/10	0/12/11	0/12/12
5	3	IHML	3.00%	14.7%	0/10/8	0/10/12	0/11/12	0/10/12
6	3	IHLH	1.80%	25.6%	0/10/8	0/10/11	0/10/11	0/12/12
7	3	IHLM	3.00%	14.7%	0/10/8	0/10/11	0/10/11	0/10/12
8	3	IHLL	1.20%	3.8%	0/10/8	0/10/11	0/10/11	0/9/11
9	3	IMHH	4.50%	38.9%	0/10/8	1/10/8	1/12/9	1/15/9
10	3	IMHM	7.50%	26.8%	0/10/8	1/10/8	1/12/9	1/12/11
11	3	IMHL	3.00%	14.7%	0/10/8	1/10/8	1/12/9	0/12/9
12	3	IMMH	7.50%	26.8%	0/10/8	1/10/8	0/11/9	0/14/9
13	3	IMMM	12.50%	15.8%	0/10/8	1/10/8	0/11/9	0/11/11
14	3	IMML	5.00%	4.7%	0/10/8	1/10/8	1/11/8	1/10/8
15	3	IMLH	3.00%	14.7%	0/10/8	1/10/8	0/10/8	0/10/12
16	3	IMLM	5.00%	4.7%	0/10/8	1/10/8	0/10/8	1/10/8
17	3	IMLL	2.00%	-5.2%	0/10/8	1/10/8	0/10/8	0/10/8
18	3	ILHH	1.80%	25.6%	0/10/8	0/10/8	0/11/9	0/12/12
19	3	ILHM	3.00%	14.7%	0/10/8	0/10/8	0/11/9	0/11/10
20	3	ILHL	1.20%	3.8%	0/10/8	0/10/8	0/11/9	0/10/9
21	3	ILMH	3.00%	14.7%	0/10/8	0/10/8	0/10/8	0/10/12
22	3	ILMM	5.00%	4.7%	0/10/8	0/10/8	0/10/8	1/10/8
23	3	ILML	2.00%	-5.2%	0/10/8	0/9/9	0/9/9	0/9/9
24	3	ILLH	1.20%	3.8%	0/10/8	0/9/9	0/8/9	1/9/10
25	3	ILLM	2.00%	-5.2%	0/10/8	0/9/8	1/8/8	1/8/9
26	3	ILLL	0.80%	-14.3%	0/10/8	0/9/9	0/8/9	0/8/8

B.2. ADP algorithm behaviour examples for non-observed scenarios

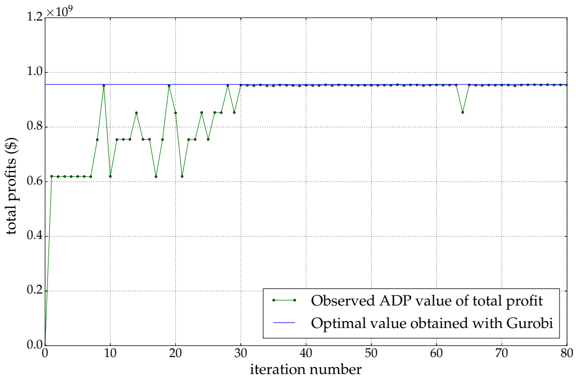


Figure B.1: IHHL scenario with $\lambda = 0.6$

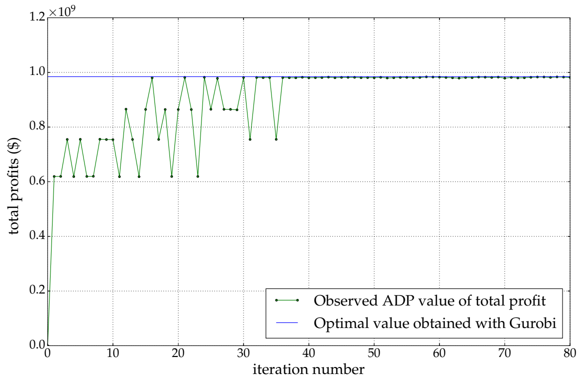


Figure B.2: IHHM scenario with $\lambda = 0.6$

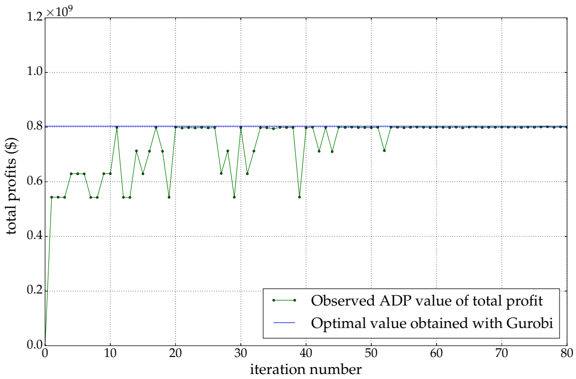


Figure B.3: ILMM scenario with $\lambda = 0.6$

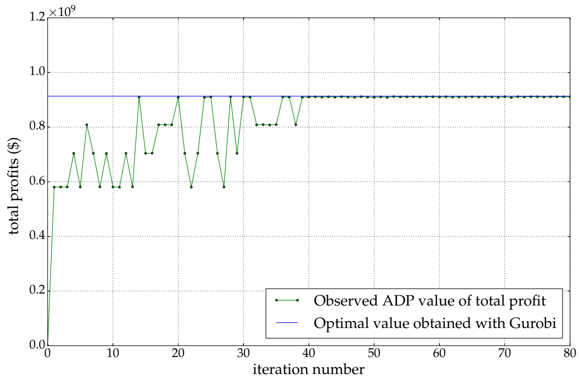


Figure B.4: IMHM scenario with $\lambda = 0.6$

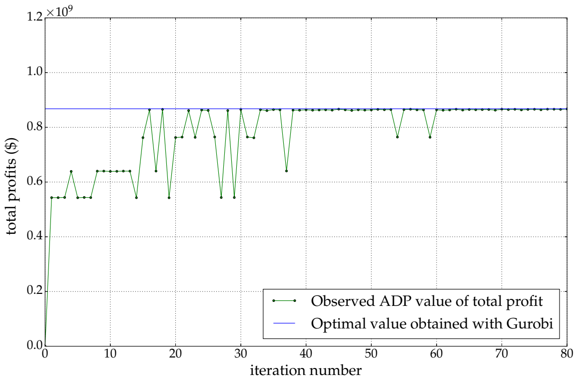


Figure B.5: ILHH scenario with $\lambda = 0.6$

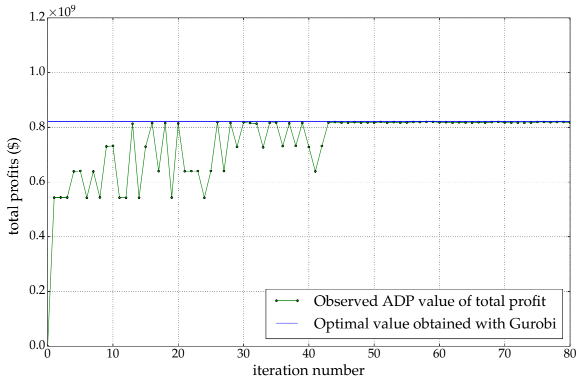


Figure B.6: ILHL scenario with $\lambda = 0.6$

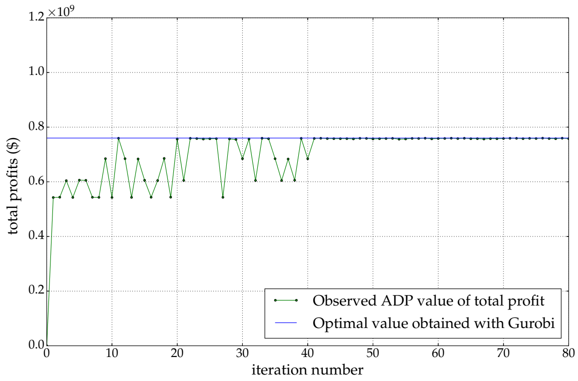


Figure B.7: ILLH scenario with $\lambda = 0.6$

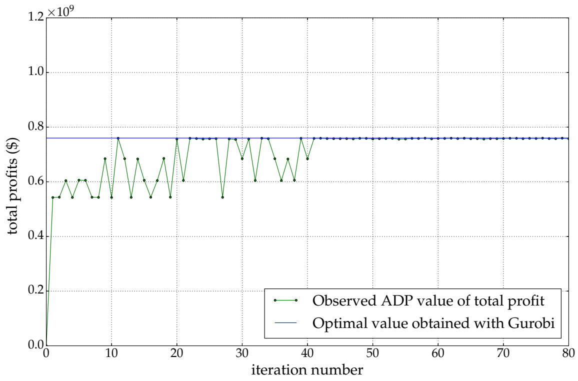


Figure B.8: ILLM scenario with $\lambda = 0.6$

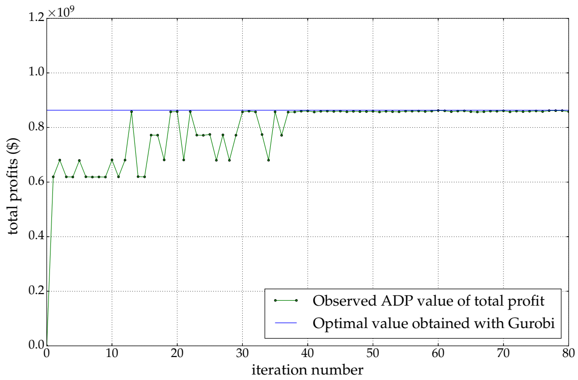


Figure B.9: IHLL scenario with $\lambda = 0.6$

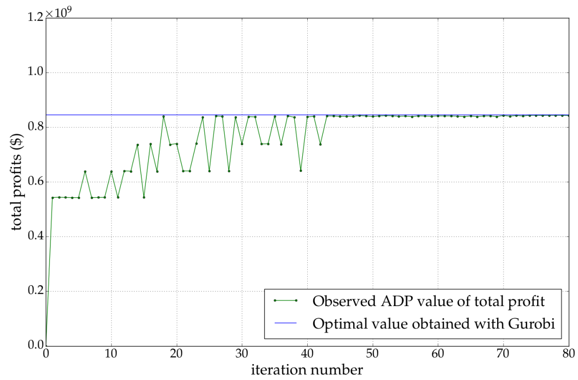


Figure B.10: ILHM scenario with $\lambda = 0.6$

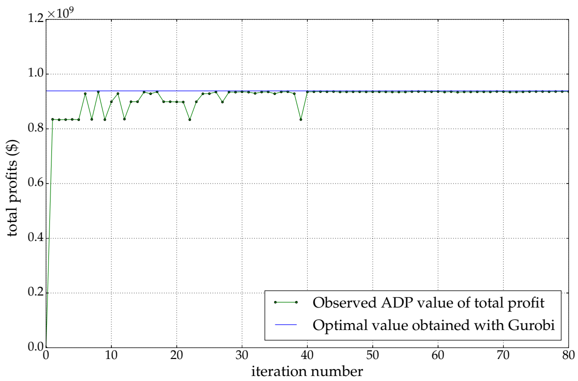


Figure B.11: IMHH scenario with $\lambda = 0.6$

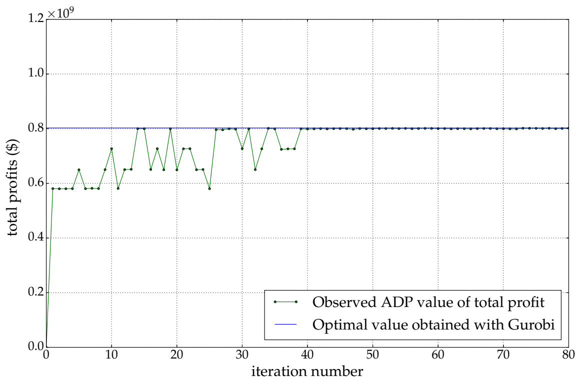


Figure B.12: IMLL scenario with $\lambda = 0.6$

C

Appendix for continuous scenario tree

This is an appendix for section 5.3.5 in chapter 5. It illustrates particular extractions of the vast continuous scenario tree generated with the Monte Carlo sampling of a normal probability distribution.

C.1. Details of the continuous scenario tree obtained with normally distributed demand growth outcomes

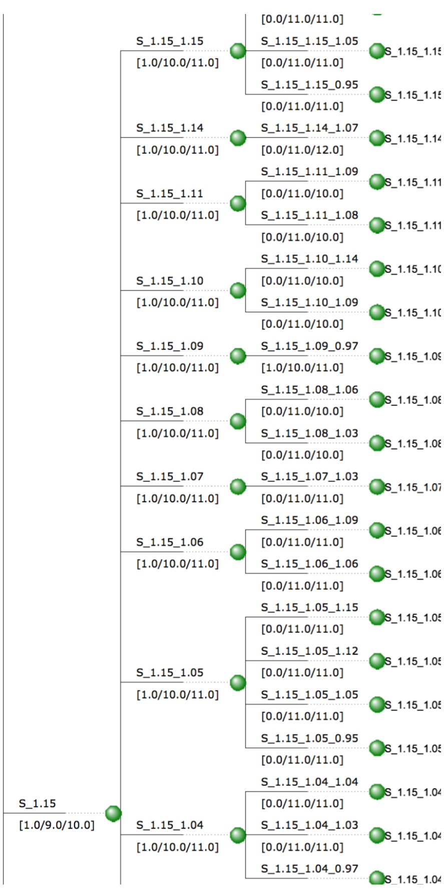


Figure C.1: Detail of optimistic branches from continuous scenario tree.

Figure C.2: Detail of pessimistic branches from continuous scenario tree.

Figure C.3: Detail of most optimistic branches from continuous scenario tree.

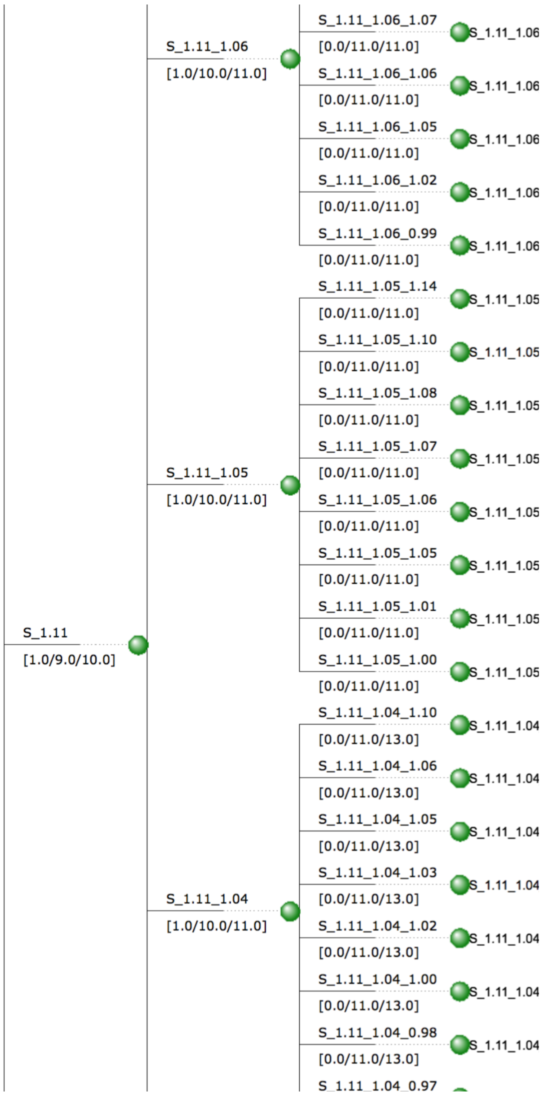


Figure C.4: Detail of most-likely/optimistic branches from continuous scenario tree.

D

Appendix for Kenya Airways case study

This is an appendix for chapter 5. On the one hand, it contains all relevant model parameters that have been used to solve the case study applied to Kenya Airways. On the other, this appendix presents the most relevant analysis and ADP results upon which the report discussions and conclusion have been based.

D.1. 5-stage scenario tree model

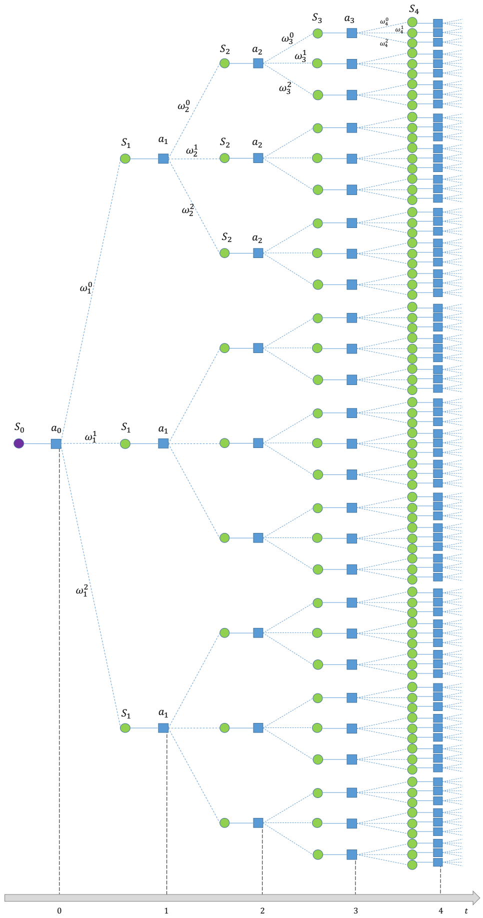


Figure D.1: 5-period scenario tree applicable to the KQ case study.

D.2. Case study parameters of KQ network

Table D.1: Route-related parameters.

	OT_r	Distance	f_r	D_{0r}	LF_r	Y_{min}
Route	[h/flight]	[km/ flight leg]	[$/flight leg]	[-/weekly]	[%]	[-/week]
1	3.51	2811	326	274	0.7	0
2	3.05	2436	326	393	0.7	3
3	3.88	3104	314	237	0.7	0
4	3,86	3086	327	517	0.7	4
5	3.01	2410	301	459	0,7	7
6	1.07	858	301	141	0.7	0
7	3,83	3064	327	125	0,7	0
8	3.55	2842	301	155	0.7	0
9	3.60	2880	329	127	0.7	0
10	0.35	280	90	1717	0,7	27
11	0.53	424	100	5506	0.7	62
12	0.52	412	103	510	0.7	14
13	1.13	904	142	1428	0.7	3
14	0.84	668	178	2232	0.7	35
15	1,97	1579	225	301	0.7	7
16	0.65	523	147	3042	0.7	35
17	2.82	2252	270	212	0,7	5
18	0.58	464	178	385	0.7	0
19	2.62	2099	264	284	0.7	4
20	2.81	2251	354	702	0.7	7
21	3,00	2400	147	657	0.7	0
22	4.35	3480	535	153	0.7	0
23	2.30	1840	159	812	0.7	7
24	8.34	6672	509	1570	0.7	7
25	8,55	6841	519	1308	0.7	7
26	6.73	5386	529	573	0.7	0
27	8.54	6830	529	1670	0.7	7
28	7.88	6301	529	546	0.7	0
29	11,13	8901	387	1150	0.7	4
30	11.13	8904	395	750	0.7	3
31	10.86	8685	428	650	0.7	3
32	11.53	9225	484	451	0.7	0
33	5,67	4534	282	1895	0.7	14
34	4.45	3556	236	1900	0.7	11
35	3.20	2556	263	178	0.7	2
36	4.62	3696	274	201	0.7	0
37	4.00	1719	195	815	0.7	9
38	2.04	1632	259	241	0.7	0
39	1.13	905	255	859	0.7	14
40	2.43	1940	253	386	0.7	7
41	4,00	3200	195	250	0.7	0
42	2.00	1600	140	646	0.7	14
43	1.93	1546	416	179	0.7	4
44	2.05	1636	298	291	0.7	3
45	2.08	1661	216	548	0.7	3
46	3.54	2831	341	248	0.7	5
47	3.64	2910	330	2626	0.7	21
48	3,44	2755	284	216	0.7	3
49	1.79	1428	232	188	0.7	7
50	2.39	1916	240	345	0.7	0
51	2.26	1809	278	143	0.7	7
52	2.61	2086	277	1036	0.7	7
53	3.45	2762	292	332	0.7	5
54	3.88	3103	520	273	0.7	3
55	4.30	3440	414	290	0.7	0
56	7.15	5720	450	389	0.7	0
57	8.03	6421	418	598	0.7	3
58	4.35	3483	310	189	0.7	3
59	7.82	6252	387	384	0.7	3
60	4.79	3831	353	572	0.7	7
61	4.92	3937	364	386	0.7	3
62	10.30	8240	600	950	0.7	7
63	11.30	9040	600	750	0.7	7
64	15.00	12002	587	541	0.7	0

Table D.2: Aircraft-route variable costs

	c_i^{var} [$/ operated route]				
Route	E70	E90	738	788	772
1	M	30383	47698	62393	96137
2	M	27314	44069	54499	84928
3	M	M	52173	68549	104880
4	M	M	51898	68171	104342
5	M	27101	35533	53952	73774
6	13878	14399	17817	25283	37759
7	M	M	51569	67718	103699
8	M	30637	48172	63045	97063
9	M	29102	41088	63845	98199
10	7018	7261	8974	9117	20482
11	8340	8834	12500	18148	33158
12	8031	10749	10993	11895	24428
13	18293	18566	24673	32252	39134
14	11738	12249	16535	21284	32080
15	21034	22606	28848	36460	59311
16	10345	10836	14740	22232	27746
17	M	25808	39145	50626	79428
18	9967	11174	11789	12990	25982
19	26195	27769	32167	47405	74854
20	M	25993	33675	50605	79398
21	M	27019	41410	53741	83852
22	M	35858	57934	76474	116134
23	23625	22436	32842	41954	67112
24	M	M	M	143664	211166
25	M	M	M	147221	216599
26	M	M	87095	116594	173107
27	M	M	M	146989	229053
28	M	M	M	135854	200457
29	M	M	M	190582	278175
30	M	M	M	190645	278264
31	M	M	M	186036	271718
32	M	M	M	197402	287859
33	M	M	54958	98660	134561
34	M	M	50116	78074	115169
35	M	27526	39241	57025	88515
36	M	M	61238	81021	122591
37	20606	24860	52113	55407	63496
38	21560	20733	29659	37576	60895
39	16680	15232	18536	22273	39164
40	22272	21690	32924	44059	70102
41	M	33567	53650	70581	107765
42	21243	20472	29170	36902	59939
43	20362	18046	28343	35765	58324
44	21600	20766	29720	37660	61015
45	21843	25531	37042	38175	61747
46	M	30083	41650	62814	96735
47	M	29242	42232	64476	94913
48	M	28644	44663	61214	94463
49	20218	20167	33595	38281	54797
50	28720	25343	39116	43543	77525
51	23317	22182	32367	41301	66186
52	25715	26641	38210	47121	76401
53	M	27878	36562	61361	94672
54	M	M	52165	68539	104865
55	M	35531	57322	75632	114938
56	M	54191	92206	123625	183091
57	M	M	96896	138380	204044
58	M	M	57979	76538	116224
59	M	M	113260	134823	198993
60	M	M	52700	83863	126626
61	M	M	65123	86094	129794
62	M	M	130762	176669	258417
63	M	M	143002	193508	282330
64	M	M	M	255856	370867

Table D.3: Annual route traffic growth forecast aggregated per region and dated 2015

		Route demand growth aggregated per region		
Region .	Route	Best case	Most-likely case	Worst case
	1	0.00%	0.00%	-8.00%
	2	0.00%	0.00%	0.00%
	3	15.00%	7.85%	2.00%
	4	15.00%	7.85%	2.00%
Region A	5	15.00%	7.85%	2.00%
	6	0.00%	0.00%	0.00%
	7	15.00%	7.85%	2.00%
	8	0.00%	0.00%	0.00%
	9	15.00%	7.85%	2.00%
	10	5.00%	2.00%	-1.00%
Region B	11	5.00%	2.00%	-1.00%
	12	5.00%	2.00%	-1.00%
	13	8.00%	4.45%	-3.00%
	14	8.00%	4.45%	-3.00%
	15	8.00%	4.45%	-3.00%
	16	8.00%	4.45%	-3.00%
	17	8.00%	4.45%	-3.00%
Region C	18	8.00%	0.00%	-3.00%
	19	8.00%	4.45%	-3.00%
	20	8.00%	4.45%	-3.00%
	21	8.00%	4.45%	-3.00%
	22	8.00%	4.45%	-3.00%
	23	8.00%	4.45%	-3.00%
	24	5.00%	0.69%	-4.00%
	25	5.00%	0.69%	-4.00%
Region D	26	5.00%	0.69%	-4.00%
	27	5.00%	0.69%	-4.00%
	28	5.00%	0.69%	-4.00%
	29	5.00%	0.98%	-1.00%
Region E	30	5.00%	0.98%	-1.00%
	31	5.00%	0.98%	-1.00%
	32	5.00%	0.98%	-1.00%
	33	9.00%	5.00%	-3.00%
Region F	34	9.00%	5.00%	-3.00%
	35	9.00%	5.00%	-3.00%
	36	9.00%	5.00%	-3.00%
	37	5.00%	2.00%	-6.00%
	38	5.00%	2.00%	-6.00%
Region G	39	5.00%	2.00%	-6.00%
	40	5.00%	2.00%	-6.00%
	41	5.00%	2.00%	-6.00%
	42	5.00%	2.00%	-6.00%
	43	10.00%	5.00%	-3.00%
	44	10.00%	5.00%	-3.00%
	45	10.00%	5.00%	-3.00%
	46	10.00%	5.00%	-3.00%
	47	10.00%	5.00%	-3.00%
	48	10.00%	5.00%	-3.00%
Region H	49	10.00%	5.00%	-3.00%
	50	10.00%	5.00%	-3.00%
	51	10.00%	5.00%	-3.00%
	52	10.00%	5.00%	-3.00%
	53	10.00%	5.00%	-3.00%
	54	10.00%	5.00%	-3.00%
	55	10.00%	5.00%	-3.00%
	56	10.00%	5.00%	-3.00%
	57	8.00%	3.03%	-5.00%
	58	8.00%	3.03%	-5.00%
	59	8.00%	3.03%	-5.00%
Region I	60	8.00%	3.03%	-5.00%
	61	8.00%	3.03%	-5.00%
	62	8.00%	3.03%	-5.00%
	63	8.00%	3.03%	-5.00%
Region J	64	5.00%	1.00%	0.00%

Table D.4: Year-over-year entry-into-market indicators for KQ rollout plan

	Entry-into-market indicator δ_{rt}				
Route	FY 0	FY 1	FY 2	FY 3	FY 4
1	0	0	1	1	1
2	1	1	0	0	0
3	1	1	1	1	1
4	1	1	1	1	1
5	1	1	1	1	1
6	0	0	1	1	1
7	0	1	1	1	1
8	0	0	0	1	1
9	1	1	1	1	1
10	1	1	1	1	1
11	1	1	1	1	1
12	1	1	1	1	1
13	1	1	1	1	1
14	1	1	1	1	1
15	1	1	1	1	1
16	1	1	1	1	1
17	1	1	1	1	1
18	0	0	0	1	1
19	1	1	1	1	1
20	1	1	1	1	1
21	0	1	1	1	1
22	0	1	1	1	1
23	1	1	1	1	1
24	1	1	1	1	1
25	1	1	1	1	1
26	0	0	0	1	1
27	1	1	1	1	1
28	0	0	0	1	1
29	1	1	1	1	1
30	1	1	1	1	1
31	1	1	1	1	1
32	0	0	0	1	1
33	1	1	1	1	1
34	1	1	1	1	1
35	1	1	1	1	1
36	0	0	0	1	1
37	1	1	1	1	1
38	0	0	0	1	1
39	1	1	1	1	1
40	1	1	1	1	1
41	0	1	1	1	1
42	1	1	1	1	1
43	1	1	1	1	1
44	1	1	1	1	1
45	1	1	1	1	1
46	1	1	1	1	1
47	1	1	1	1	1
48	1	1	1	1	1
49	1	1	1	1	1
50	1	1	1	1	1
51	1	1	1	1	1
52	1	1	1	1	1
53	1	1	1	1	1
54	1	1	1	1	1
55	0	1	1	1	1
56	0	1	1	1	1
57	1	1	1	1	1
58	1	1	1	1	1
59	1	1	1	1	1
60	1	1	1	1	1
61	1	1	1	1	1
62	1	1	1	1	1
63	1	1	1	1	1
64	0	0	1	1	1

D.3. Optimal fleet plans for each independent scenario

Table D.5: Optimal fleet for each scenario solved independently

				Optimal fleet for independent scenarios				
ID	Scenario	Total probability	Demand variation	year 0	year 1	year 2	year 3	year 4
0	IHHHH	0.39%	32.02%	3/15/10/6/4	1/14/9/9/0	1/14/9/10/0	1/14/9/12/0	1/14/10/12/0
1	IHHHM	0.78%	26.76%	3/15/10/6/4	1/14/9/9/0	1/14/9/10/0	1/14/9/12/0	1/14/9/12/1
2	IHHHL	0.39%	18.89%	3/15/10/6/4	1/14/9/9/0	1/14/9/10/0	1/14/9/11/1	1/14/9/11/1
3	IHHMH	0.78%	27.19%	3/15/10/6/4	1/14/9/9/0	1/14/9/10/0	1/14/9/11/0	1/14/9/12/0
4	IHHMM	1.56%	22.14%	3/15/10/6/4	1/14/9/9/0	1/14/9/10/0	1/14/9/11/0	1/14/9/11/1
5	IHHML	0.78%	14.57%	3/15/10/6/4	1/14/9/9/0	1/14/9/10/0	1/14/9/11/0	1/14/9/11/0
6	IHHLH	0.39%	19.79%	3/15/10/6/4	1/14/9/9/0	1/14/9/10/0	1/14/9/11/0	1/14/9/12/0
7	IHHLM	0.78%	15.05%	3/15/10/6/4	1/14/9/9/0	1/14/9/10/0	1/14/9/10/1	1/14/9/10/1
8	IHHLL	0.39%	7.98%	3/15/10/6/4	1/14/9/9/0	1/14/9/10/0	1/15/9/10/0	1/15/9/10/0
9	IHMHH	0.78%	27.25%	3/15/10/6/4	1/14/9/9/0	1/15/9/9/0	1/15/9/10/1	1/15/9/11/1
10	IHMHM	1.56%	22.20%	3/15/10/6/4	1/14/9/9/0	2/14/9/9/0	1/14/9/10/1	1/14/9/11/1
11	IHMHL	0.78%	14.63%	3/15/10/6/4	1/14/9/9/0	2/14/9/9/0	1/14/9/10/1	1/14/9/10/1
12	IHMMH	1.56%	22.63%	3/15/10/6/4	1/14/9/9/0	1/14/9/10/0	1/14/9/10/1	1/14/9/11/1
13	IHMMM	3.13%	17.78%	3/15/10/6/4	1/14/9/9/0	2/14/9/9/0	1/14/9/10/1	1/14/9/10/1
14	IHMML	1.56%	10.51%	3/15/10/6/4	1/14/9/9/0	1/14/9/10/0	1/14/9/10/1	1/14/9/10/1
15	IHMLH	0.78%	15.54%	3/15/10/6/4	1/14/9/9/0	1/14/9/10/0	1/14/9/10/0	1/14/9/11/0
16	IHMLM	1.56%	10.98%	3/15/10/6/4	1/14/9/9/0	1/13/9/10/0	1/14/9/10/0	1/14/9/10/1
17	IHMLL	0.78%	4.19%	3/15/10/6/4	1/14/9/9/0	1/14/9/10/0	1/14/9/10/0	1/14/9/10/0
18	IHLHH	0.39%	20.00%	3/15/10/6/4	1/14/9/9/0	1/14/9/9/0	1/14/9/10/1	1/14/9/11/1
19	IHLHM	0.78%	15.25%	3/15/10/6/4	1/14/9/9/0	1/14/9/9/0	1/14/9/11/0	1/14/9/11/0
20	IHLHL	0.39%	8.18%	3/15/10/6/4	1/14/9/9/0	1/14/9/9/0	1/15/9/9/1	1/15/9/9/1
21	IHLMH	0.78%	15.68%	3/15/10/6/4	1/14/9/9/0	1/14/9/9/0	1/14/9/10/0	1/14/9/11/0
22	IHLMM	1.56%	11.12%	3/15/10/6/4	1/14/9/9/0	1/14/9/9/0	1/14/9/10/0	1/14/9/10/1
23	IHLML	0.78%	4.32%	3/15/10/6/4	1/14/9/9/0	1/14/9/9/0	1/14/9/10/0	1/14/9/10/0
24	IHLLH	0.39%	9.08%	3/15/10/6/4	2/13/9/9/0	2/13/9/9/0	2/13/9/9/1	2/14/9/9/1
25	IHLLM	0.78%	4.79%	3/15/10/6/4	2/13/9/9/0	2/13/9/9/0	2/13/9/10/0	2/13/9/10/0
26	IHLLL	0.39%	-1.57%	3/15/10/6/4	1/14/9/9/0	1/14/9/9/0	2/14/9/9/0	2/14/9/9/0
27	IMHHH	0.78%	27.66%	3/15/10/6/4	1/14/9/9/0	1/14/9/10/0	1/14/9/10/1	1/15/9/11/1
28	IMHHM	1.56%	22.58%	3/15/10/6/4	1/14/9/9/0	1/14/9/10/0	1/14/9/11/0	1/14/9/11/1
29	IMHHL	0.78%	14.98%	3/15/10/6/4	1/14/9/8/1	1/14/9/9/1	1/14/9/10/1	1/14/9/10/1
30	IMHMH	1.56%	23.02%	3/15/10/6/4	1/14/9/9/0	1/14/9/10/0	1/14/9/11/0	1/14/10/11/0
31	IMHMM	3.13%	18.15%	3/15/10/6/4	1/14/9/9/0	1/14/9/10/0	1/14/9/10/1	1/15/9/10/1
32	IMHML	1.56%	10.84%	3/15/10/6/4	1/14/9/9/0	2/14/9/9/0	1/14/9/10/1	1/14/9/10/1
33	IMHLH	0.78%	15.89%	3/15/10/6/4	1/14/9/8/1	1/14/9/9/1	1/14/9/9/1	1/14/9/10/1
34	IMHLM	1.56%	11.31%	3/15/10/6/4	1/14/9/9/0	1/14/9/10/0	1/14/9/10/0	1/14/9/10/1
35	IMHLL	0.78%	4.48%	3/15/10/6/4	2/13/9/9/0	1/13/9/10/0	2/13/9/10/0	2/13/9/10/0
36	IMMHH	1.56%	23.08%	3/15/10/6/4	1/14/9/9/0	1/14/9/9/0	1/14/9/10/1	1/14/9/11/1
37	IMMHM	3.13%	18.21%	3/15/10/6/4	1/14/9/9/0	1/14/9/9/0	1/14/9/10/1	1/14/9/10/1
38	IMMHL	1.56%	10.90%	3/15/10/6/4	1/14/9/9/0	1/14/9/9/0	1/14/9/10/1	1/14/9/10/1
39	IMMMH	3.13%	18.64%	3/15/10/6/4	1/14/9/9/0	1/14/9/9/0	1/14/9/10/0	1/14/9/12/0
40	IMMMM	6.25%	13.96%	3/15/10/6/4	1/14/9/9/0	1/14/9/9/0	1/14/9/10/0	1/14/9/11/0
41	IMMML	3.13%	6.93%	3/15/10/6/4	1/14/9/9/0	1/14/9/9/0	1/14/9/10/0	1/14/9/10/0
42	IMMLH	1.56%	11.80%	3/15/10/6/4	1/14/9/9/0	1/14/9/9/0	1/14/9/10/0	1/14/9/11/0
43	IMMLM	3.13%	7.40%	3/15/10/6/4	2/13/9/9/0	2/13/9/9/0	2/13/9/10/0	2/13/9/10/0
44	IMMLL	1.56%	0.84%	3/15/10/6/4	2/13/9/9/0	2/13/9/9/0	2/13/9/10/0	2/13/9/10/0
45	IMLHH	0.78%	16.10%	3/15/10/6/4	1/14/9/9/0	1/14/9/9/0	1/14/9/10/0	1/14/9/10/1
46	IMLHM	1.56%	11.51%	3/15/10/6/4	1/14/9/9/0	1/14/9/9/0	1/14/9/10/0	1/14/9/10/1
47	IMLHL	0.78%	4.68%	3/15/10/6/4	2/13/9/9/0	2/13/9/9/0	2/13/9/10/0	2/13/9/10/0
48	IMLMH	1.56%	11.95%	3/15/10/6/4	2/13/9/9/0	2/13/9/9/0	2/13/9/9/1	2/13/9/10/1
49	IMLMM	3.13%	7.54%	3/15/10/6/4	2/13/9/9/0	2/13/9/9/0	2/13/9/9/1	2/13/9/9/1
50	IMLML	1.56%	0.97%	3/15/10/6/4	2/13/9/9/0	2/13/9/9/0	2/13/9/10/0	2/13/9/10/0
51	IMLLH	0.78%	5.58%	3/15/10/6/4	2/13/9/9/0	2/13/9/9/0	2/13/9/9/0	2/13/9/10/0
52	IMLLM	1.56%	1.44%	3/15/10/6/4	2/13/9/9/0	2/13/9/9/0	2/13/9/9/0	2/13/9/9/1
53	IMLLL	0.78%	-4.70%	3/15/10/6/4	2/13/9/9/0	2/13/9/9/0	2/13/9/9/0	2/13/9/9/0
54	ILHHH	0.39%	21.15%	3/15/10/6/4	2/14/9/8/0	1/14/9/9/0	1/14/9/10/1	1/15/9/10/1
55	ILHHM	0.78%	16.34%	3/15/10/6/4	2/14/9/8/0	1/14/9/9/0	1/14/9/10/1	1/14/9/10/1
56	ILHHL	0.39%	9.17%	3/15/10/6/4	2/14/9/8/0	1/14/9/9/0	1/15/9/10/0	1/15/9/10/0
57	ILHMH	0.78%	16.77%	3/15/10/6/4	2/14/9/8/0	1/14/9/9/0	1/14/9/10/0	1/14/9/11/0
58	ILHMM	1.56%	12.16%	3/15/10/6/4	2/14/9/8/0	1/14/9/9/0	1/14/9/9/1	1/14/9/10/1
59	ILHML	0.78%	5.26%	3/15/10/6/4	2/14/9/8/0	1/14/9/9/0	1/14/9/10/0	1/14/9/10/0
60	ILHLH	0.39%	10.08%	3/15/10/6/4	2/13/9/8/0	2/13/9/9/0	2/13/9/10/0	2/14/9/10/0
61	ILHLM	0.78%	5.74%	3/15/10/6/4	2/13/9/8/0	2/13/9/9/0	2/13/9/10/0	2/13/9/10/0
62	ILHLL	0.39%	-0.70%	3/15/10/6/4	2/13/9/8/0	2/13/9/9/0	2/14/9/9/0	2/14/9/9/0
63	ILMHH	0.78%	16.84%	3/15/10/6/4	2/14/9/8/0	1/14/9/9/0	1/14/9/10/0	1/16/9/10/0
64	ILMHM	1.56%	12.22%	3/15/10/6/4	2/14/9/8/0	1/14/9/9/0	1/14/9/10/0	1/14/9/10/1
65	ILMHL	0.78%	5.33%	3/15/10/6/4	2/13/9/8/0	2/13/9/9/0	2/13/9/10/0	2/13/9/10/0
66	ILMMH	1.56%	12.65%	3/15/10/6/4	2/13/9/8/0	2/13/9/9/0	2/13/9/10/0	2/13/9/11/0
67	ILMMM	3.13%	8.22%	3/15/10/6/4	2/13/9/8/0	2/13/9/9/0	2/13/9/10/0	2/13/9/10/0
68	ILMML	1.56%	1.59%	3/15/10/6/4	2/13/9/8/0	2/13/9/9/0	2/13/9/9/1	2/13/9/9/1
69	ILMLH	0.78%	6.23%	3/15/10/6/4	2/13/9/8/0	2/13/9/9/0	2/13/9/9/0	2/13/9/10/0
70	ILMLM	1.56%	2.06%	3/15/10/6/4	2/13/9/8/0	2/13/9/9/0	2/13/9/9/0	2/13/9/9/1
71	ILMLL	0.78%	-4.14%	3/15/10/6/4	2/13/9/8/0	2/13/9/9/0	2/13/9/9/0	2/13/9/9/0
72	ILLHH	0.39%	10.29%	3/15/10/6/4	2/13/9/8/0	2/13/9/8/0	2/13/9/9/1	2/13/9/10/1
73	ILLHM	0.78%	5.94%	3/15/10/6/4	2/14/9/8/0	2/13/9/8/0	2/13/9/10/0	2/13/9/10/0
74	ILLHL	0.39%	-0.51%	3/15/10/6/4	2/14/9/8/0	2/14/9/8/0	2/14/9/9/0	2/14/9/9/0
75	ILLMH	0.78%	6.37%	3/15/10/6/4	2/13/9/8/0	2/13/9/8/0	2/13/9/9/0	2/13/9/10/0
76	ILLMM	1.56%	2.20%	3/15/10/6/4	2/13/9/8/0	2/13/9/8/0	2/13/9/8/1	2/13/9/9/1
77	ILLML	0.78%	-4.01%	3/15/10/6/4	2/13/9/8/0	2/13/9/8/0	2/13/9/9/0	2/13/9/9/0
78	ILLLH	0.39%	0.40%	3/15/10/6/4	2/13/9/8/0	2/13/9/8/0	2/13/9/9/0	2/14/9/9/0
79	ILLLM	0.78%	-3.53%	3/15/10/6/4	2/13/9/8/0	2/13/9/8/0	2/13/9/9/0	2/13/9/9/0
80	ILLLL	0.39%	-9.33%	3/15/10/6/4	2/13/9/8/0	2/13/9/8/0	2/13/9/8/1	2/13/9/8/1

D.4. Profitability analysis

Table D.6: Performance analysis of recommended ADP policy against most-likely solution and independent-scenario solutions - operating profits expressed on a weekly basis in USD

				Independent scenarios		Most-likely solution		ADP adaptive policy	
ID	Scenario	Total probability	Demand variation	Optimality Gap	OF	Dif.Best	OF	Dif.Best	OF
0	IHHHH	0.39%	32.02%	0.82%	1289499	-5.30%	1221168	-0.87%	1278281
1	IHHHM	0.78%	26.76%	0.78%	1233933	-4.58%	1177390	-0.96%	1222100
2	IHHHL	0.39%	18.89%	0.52%	1121545	-3.66%	1080508	-2.11%	1097912
3	IHHMH	0.78%	27.19%	0.94%	1172776	-3.59%	1130704	-0.92%	1162025
4	IHHMM	1.56%	22.14%	0.66%	1114352	-3.23%	1078317	-0.83%	1105112
5	IHHML	0.78%	14.57%	0.60%	1009371	-2.86%	980456	-1.29%	996338
6	IHHLH	0.39%	19.79%	0.67%	966618	-1.53%	951818	-0.96%	957296
7	IHHLM	0.78%	15.05%	0.72%	912058	-1.51%	898319	-1.46%	898729
8	IHHLL	0.39%	7.98%	0.86%	810288	-1.71%	796459	-1.17%	800777
9	IHMHH	0.78%	27.25%	0.64%	1117959	-3.10%	1083301	-0.54%	1111962
10	IHMHM	1.56%	22.20%	0.85%	1054732	-2.26%	1030913	0.03%	1055025
11	IHMHL	0.78%	14.63%	0.73%	950646	-1.85%	933052	-0.42%	946665
12	IHMMH	1.56%	22.63%	0.83%	1004434	-1.34%	991010	0.14%	1005870
13	IHMMM	3.13%	17.78%	0.61%	941270	-0.83%	933490	-0.11%	940259
14	IHMML	1.56%	10.51%	1.37%	839001	-0.21%	837241	-0.36%	835944
15	IHMLH	0.78%	15.54%	0.56%	805490	-0.05%	805059	-1.13%	796404
16	IHMLM	1.56%	10.98%	1.11%	745383	0.16%	746546	-0.52%	741478
17	IHMLL	0.78%	4.19%	1.28%	647763	-1.03%	641114	-2.16%	633761
18	IHLHH	0.39%	20.00%	0.74%	817268	-0.35%	814367	-1.91%	801695
19	IHLHM	0.78%	15.25%	0.89%	762071	-0.16%	760868	-1.71%	749025
20	IHLHL	0.39%	8.18%	1.04%	661245	-0.34%	659008	-2.19%	646765
21	IHLMH	0.78%	15.68%	0.51%	714757	0.07%	715241	-2.09%	699815
22	IHLMM	1.56%	11.12%	0.55%	658090	-0.21%	656728	-1.93%	645362
23	IHLML	0.78%	4.32%	0.92%	559152	-1.40%	551304	-4.10%	536248
24	IHLLH	0.39%	9.08%	1.40%	513058	0.35%	514840	-3.81%	493513
25	IHLLM	0.78%	4.79%	1.50%	459691	-1.70%	451861	-5.79%	433098
26	IHLLL	0.39%	-1.57%	1.50%	346216	-5.42%	327444	-13.55%	299310
27	IMHHH	0.78%	27.66%	1.06%	1061817	-3.20%	1027822	-0.60%	1055474
28	IMHHM	1.56%	22.58%	1.01%	1001501	-2.43%	977136	-0.11%	1000381
29	IMHHL	0.78%	14.98%	1.44%	891306	-1.43%	878596	0.02%	891523
30	IMHMH	1.56%	23.02%	1.06%	950226	-1.46%	936391	0.25%	952557
31	IMHMM	3.13%	18.15%	0.98%	885261	-0.81%	878086	0.20%	887026
32	IMHML	1.56%	10.84%	0.90%	789039	-1.05%	780783	-0.98%	781335
33	IMHLH	0.78%	15.89%	1.50%	742195	0.98%	749486	-0.31%	739865
34	IMHLM	1.56%	11.31%	1.03%	691858	-0.11%	691126	-0.98%	685046
. 35	IMHLL	0.78%	4.48%	0.98%	596172	-1.48%	587353	-2.96%	578516
36	IMMHH	1.56%	23.08%	0.90%	901812	-1.56%	887708	-0.84%	894259
37	IMMHM	3.13%	18.21%	1.03%	835186	-0.69%	829404	-0.88%	827838
38	IMMHL	1.56%	10.90%	0.78%	740436	-1.13%	732101	-2.12%	724710
39	IMMMH	3.13%	18.64%	0.95%	781678	0.16%	782945	-0.59%	777054
40	IMMMM	6.25%	13.96%	1.29%	728496	0.00%	728496	-0.22%	726902
41	IMMML	3.13%	6.93%	0.83%	629653	-0.84%	624386	-0.96%	623584
42	IMMLH	1.56%	11.80%	1.17%	592897	0.35%	594942	-1.03%	586771
43	IMMLM	3.13%	7.40%	1.50%	534780	-0.56%	531792	-2.11%	523509
44	IMMLL	1.56%	0.84%	1.50%	436702	-3.12%	423085	-4.89%	415344
45	IMLHH	0.78%	16.10%	1.18%	597866	0.47%	600702	0.08%	598315
46	IMLHM	1.56%	11.51%	1.12%	542146	0.04%	542342	0.07%	542542
47	IMLHL	0.78%	4.68%	1.44%	445621	-1.58%	438569	-2.07%	436387
48	IMLMH	1.56%	11.95%	1.50%	490875	0.75%	494557	-0.33%	489256
49	IMLMM	3.13%	7.54%	1.50%	436117	-1.06%	431473	-1.76%	428424
50	IMLML	1.56%	0.97%	1.50%	338349	-4.63%	322700	-5.30%	320401
51	IMLLH	0.78%	5.58%	1.41%	290836	-5.36%	275236	-7.70%	268440
52	IMLLM	1.56%	1.44%	1.50%	229574	-9.86%	206928	-7.15%	213163
53	IMLLL	0.78%	-4.70%	1.50%	121143	-35.57%	78049	-17.45%	100006
54	ILHHH	0.39%	21.15%	1.20%	668880	-2.06%	655134	-1.92%	656043
55	ILHHM	0.78%	16.34%	1.11%	615721	-1.83%	604434	-1.85%	604353
56	ILHHL	0.39%	9.17%	1.41%	510695	-2.12%	499879	-2.24%	499274
57	ILHMH	0.78%	16.77%	1.50%	559650	-0.70%	555739	-0.90%	554601
58	ILHMM	1.56%	12.16%	1.50%	504072	-1.19%	498086	-0.95%	499290
59	ILHML	0.78%	5.26%	1.50%	404215	-3.16%	391442	-2.50%	394104
60	ILHLH	0.39%	10.08%	1.46%	368197	-2.64%	358494	-2.42%	359272
61	ILHLM	0.78%	5.74%	1.50%	312198	-4.76%	297347	-3.16%	302323
62	ILHLL	0.39%	-0.70%	1.50%	202656	-11.97%	178397	-6.60%	189273
63	ILMHH	0.78%	16.84%	1.50%	499508	-0.55%	496772	-1.71%	490978
64	ILMHM	1.56%	12.22%	1.50%	446221	-1.59%	439119	-2.24%	436240
65	ILMHL	0.78%	5.33%	1.50%	347411	-4.30%	332475	-5.43%	328535
66	ILMMH	1.56%	12.65%	1.21%	399089	-2.02%	391021	-3.15%	386525
67	ILMMM	3.13%	8.22%	1.50%	339472	-3.32%	328205	-3.01%	329254
68	ILMML	1.56%	1.59%	1.50%	240680	-9.22%	218500	-9.00%	219024
69	ILMLH	0.78%	6.23%	1.50%	194918	-10.19%	175051	-13.38%	168846
70	ILMLM	1.56%	2.06%	1.50%	135788	-20.32%	108192	-18.41%	110794
71	ILMLL	0.78%	-4.14%	1.50%	27374	-155.29%	-15135	-138.91%	-10651
72	ILLHH	0.39%	10.29%	1.50%	196031	-7.14%	182037	-16.04%	164580
73	ILLHM	0.78%	5.94%	1.50%	144590	-16.46%	120794	-13.57%	124972
74	ILLHL	0.39%	-0.51%	1.50%	37108	-94.84%	1916	-102.93%	-1089
75	ILLMH	0.78%	6.37%	1.50%	83306	-30.86%	57602	-27.39%	60488
76	ILLMM	1.56%	2.20%	1.50%	23720	-139.03%	-9257	-117.40%	-4128
77	ILLML	0.78%	-4.01%	1.50%	-82369	-60.72%	-132385	-21.32%	-99928
78	ILLLH	0.39%	0.40%	1.50%	-124688	-37.33%	-171234	-20.52%	-150273
79	ILLLM	0.78%	-3.53%	1.50%	-185460	-30.90%	-242773	-11.11%	-206070
80	ILLLL	0.39%	-9.33%	1.50%	-298990	-28.94%	-385524	-7.66%	-321899
			Expected values		618438	-1.99%	606154	-1.59%	608596

D.5. Fleet maps

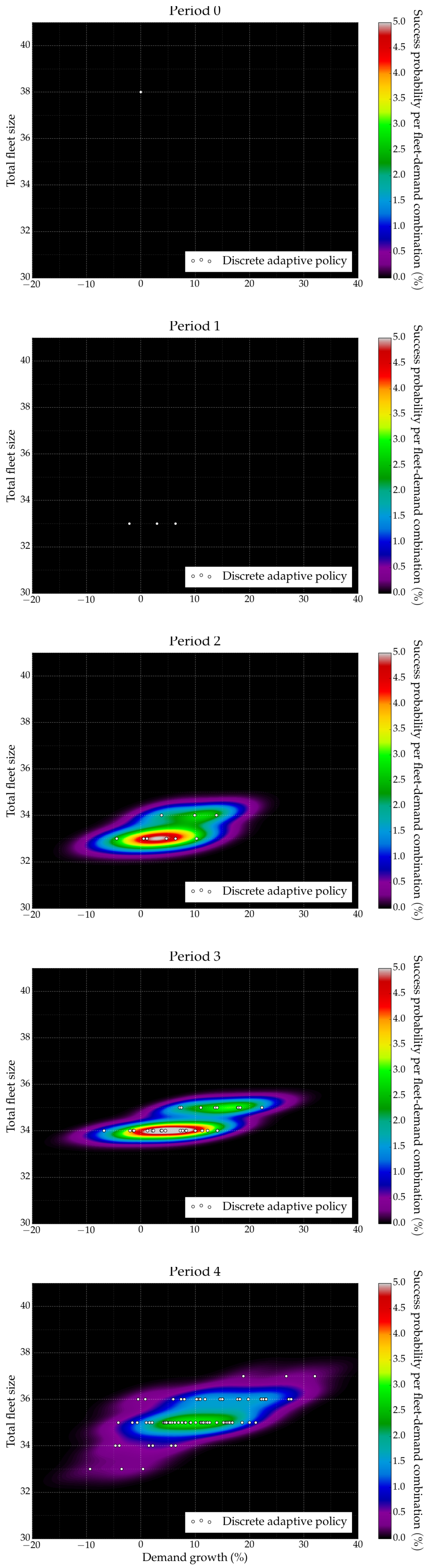


Figure D.2: Evolution of total fleet size map

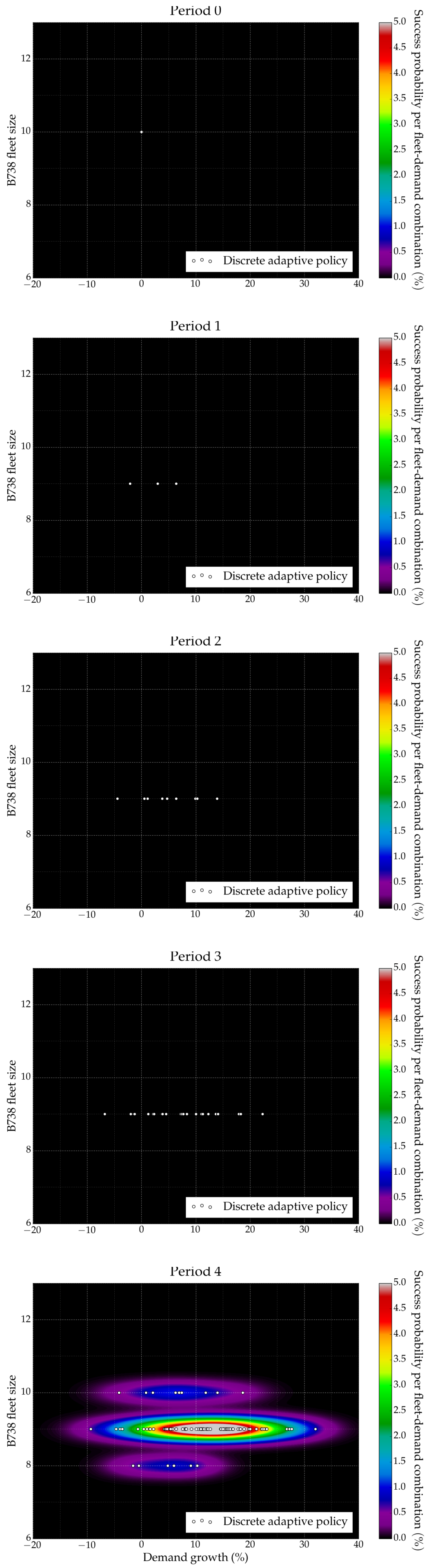


Figure D.3: Evolution of B738 fleet map

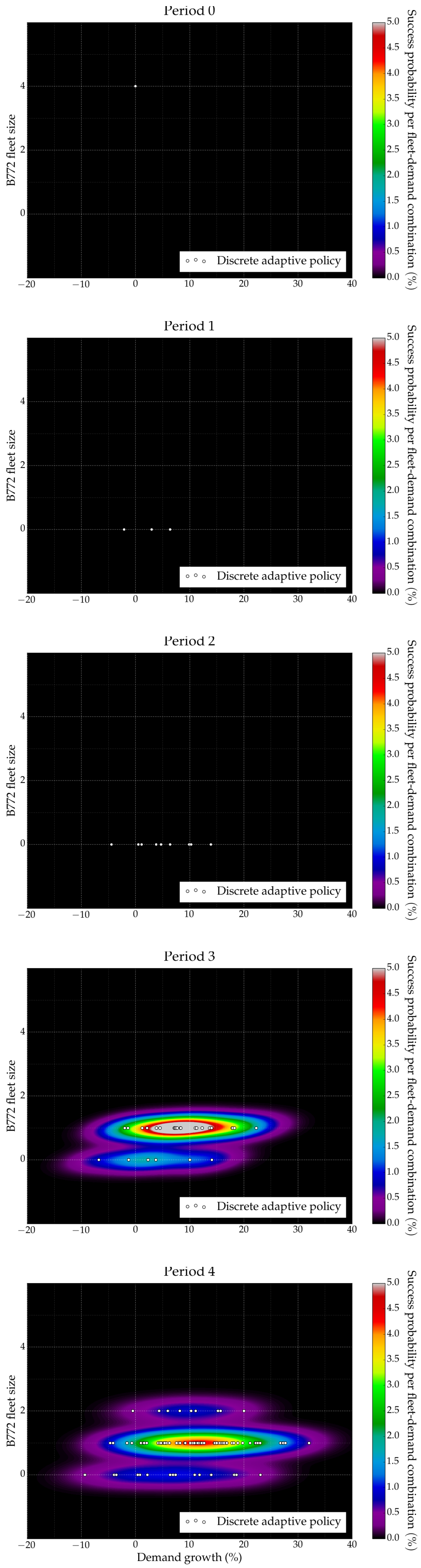


Figure D.4: Evolution of B772 fleet map

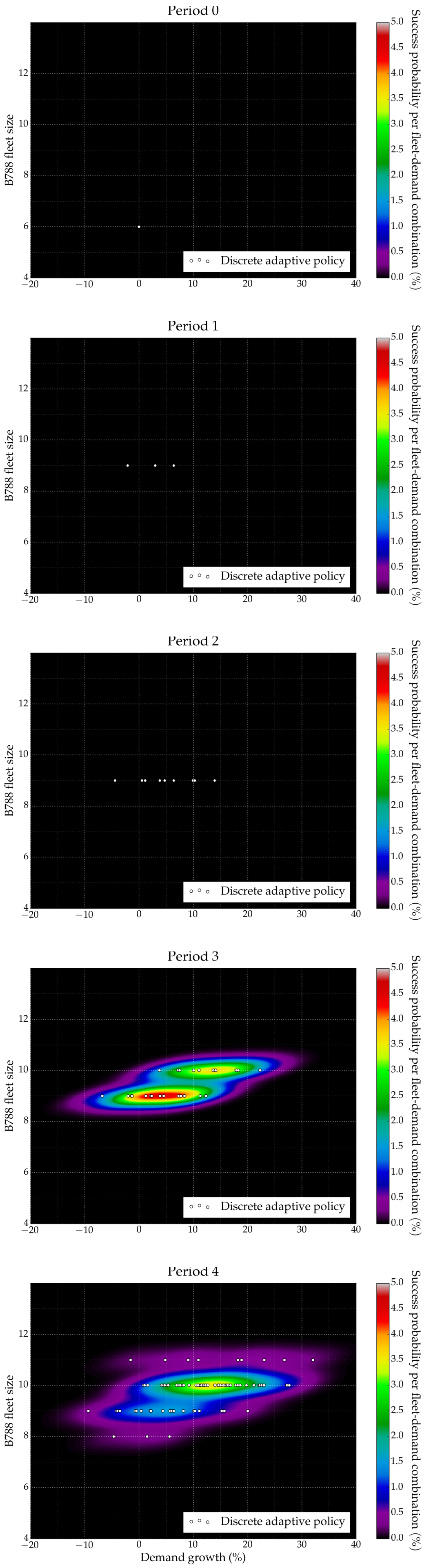


Figure D.5: Evolution of B788 fleet map

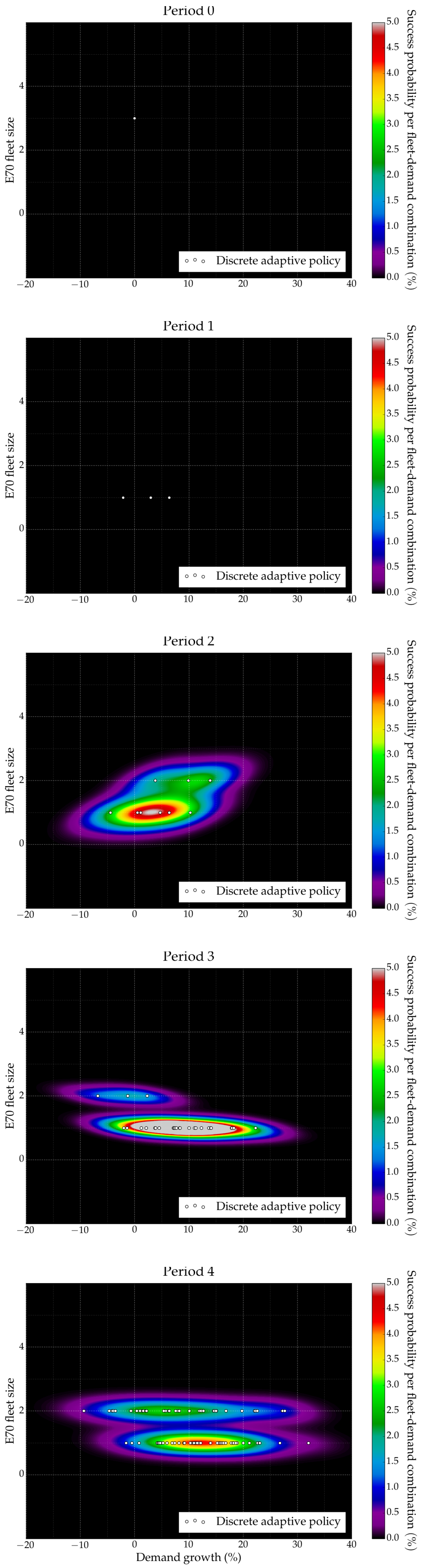


Figure D.6: Evolution of E70 fleet map

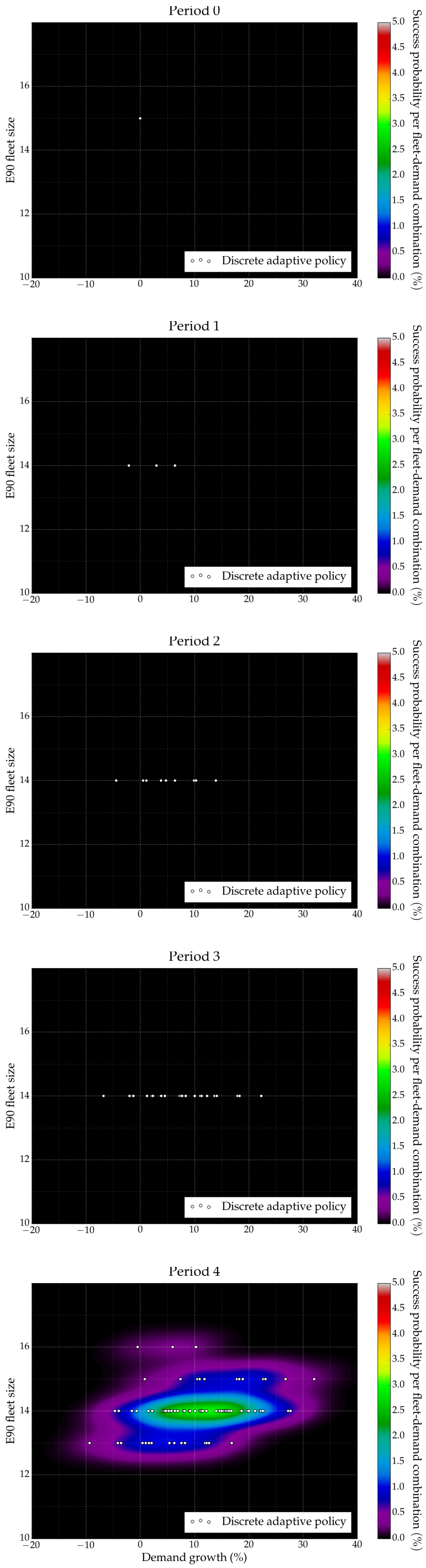
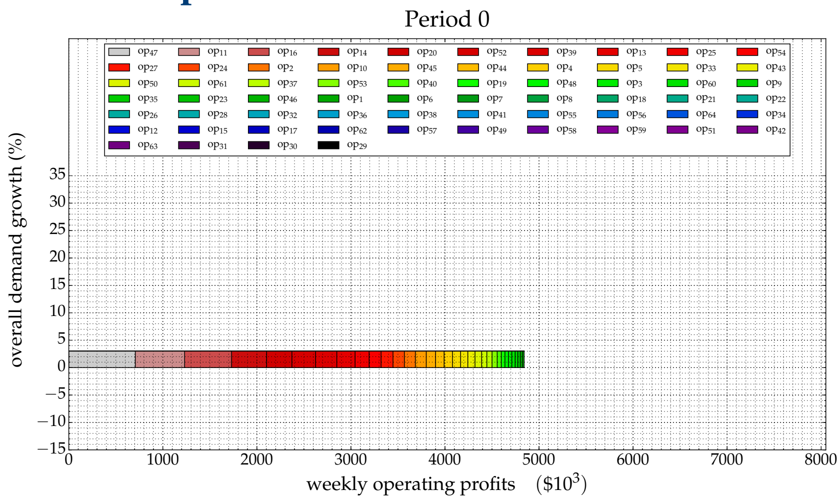
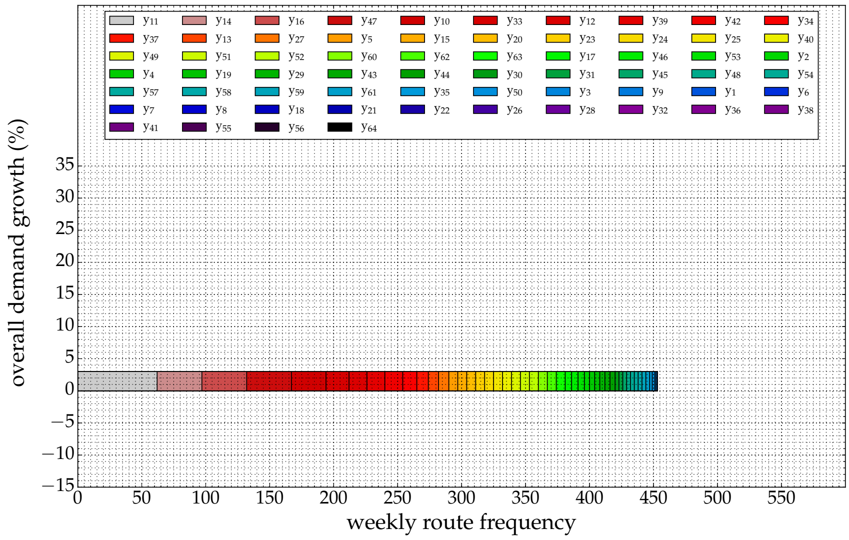


Figure D.7: Evolution of E90 fleet map

D.6. Operational maps





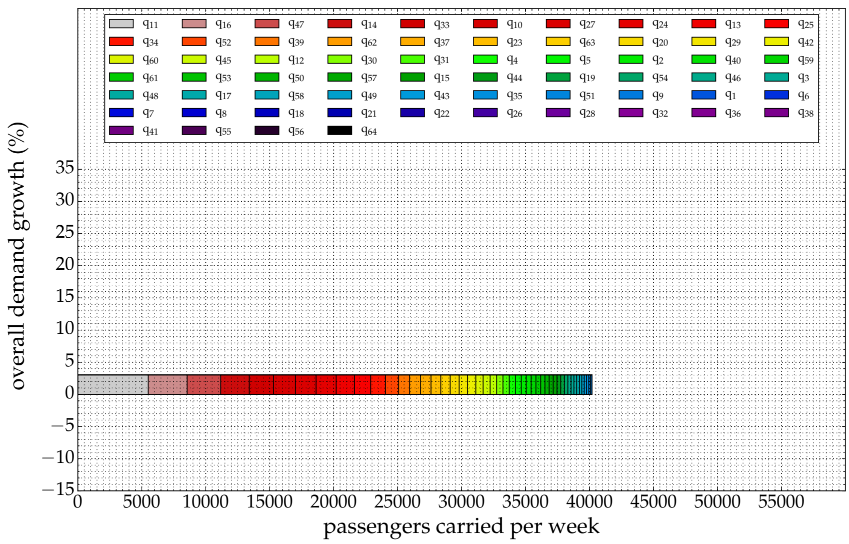


Figure D.8: Average weekly operating profits, route frequency and passengers carried per scenario in year 0

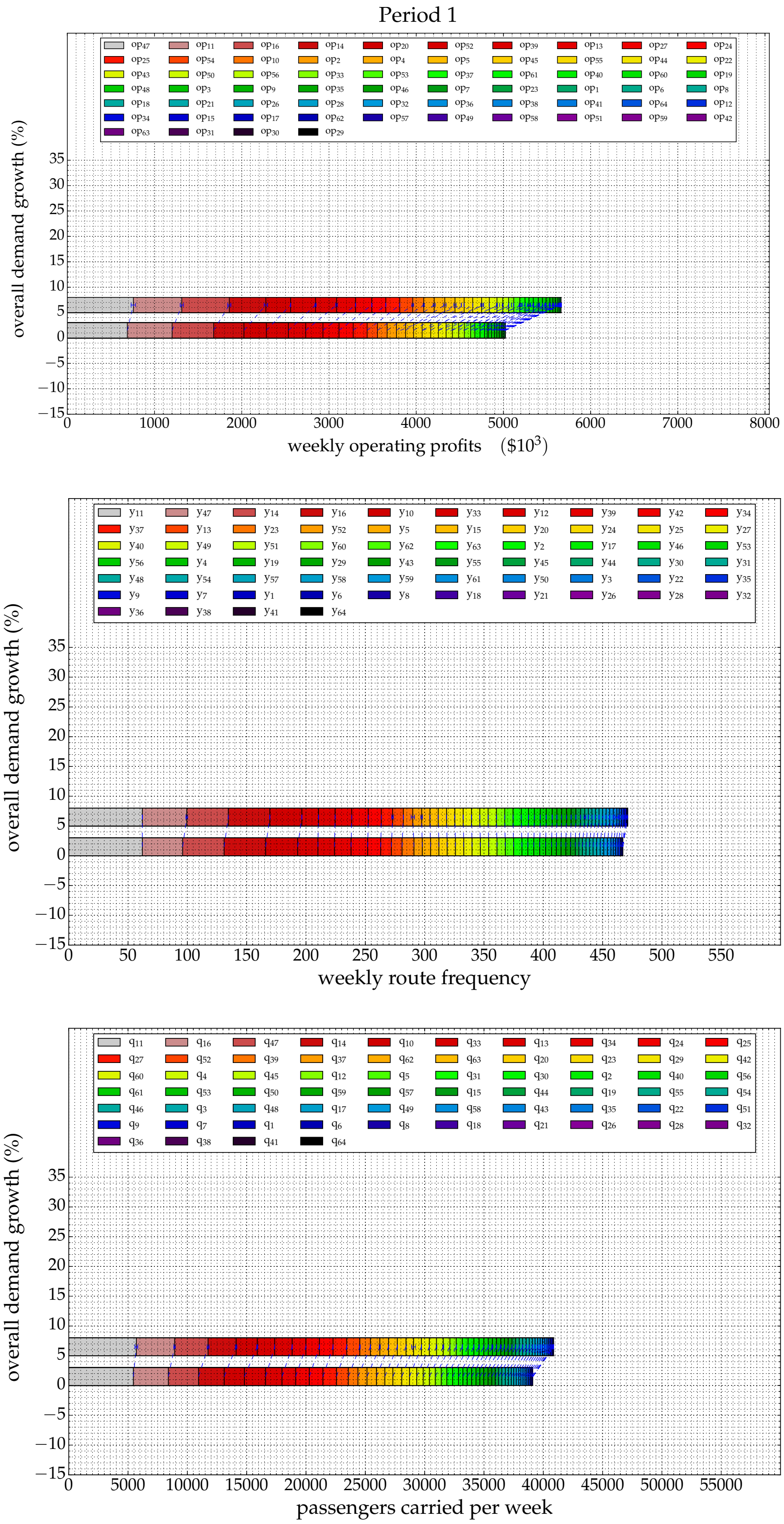


Figure D.9: Average weekly operating profits, route frequency and passengers carried per scenario in year 1

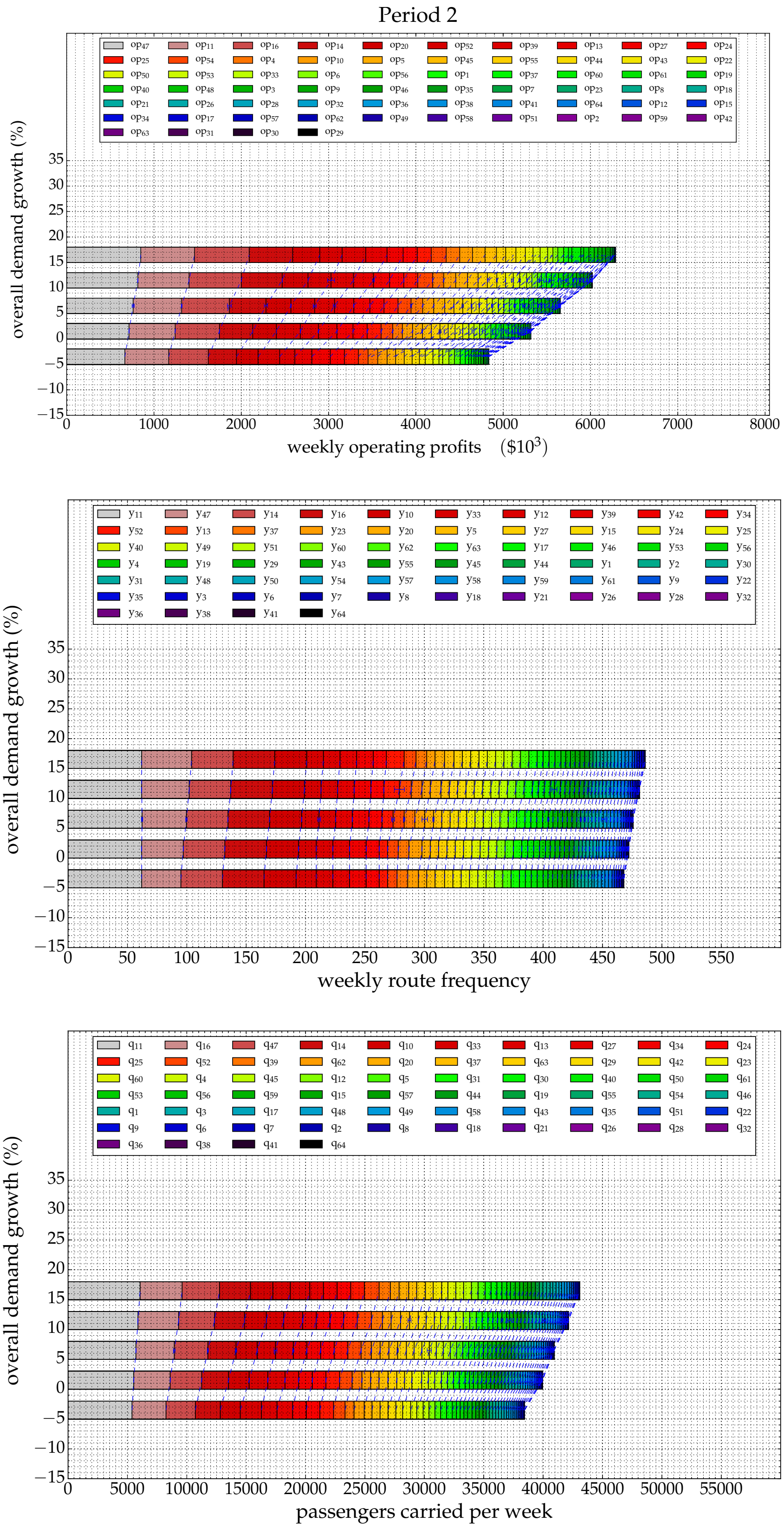


Figure D.10: Average weekly operating profits, route frequency and passengers carried per scenario in year 2

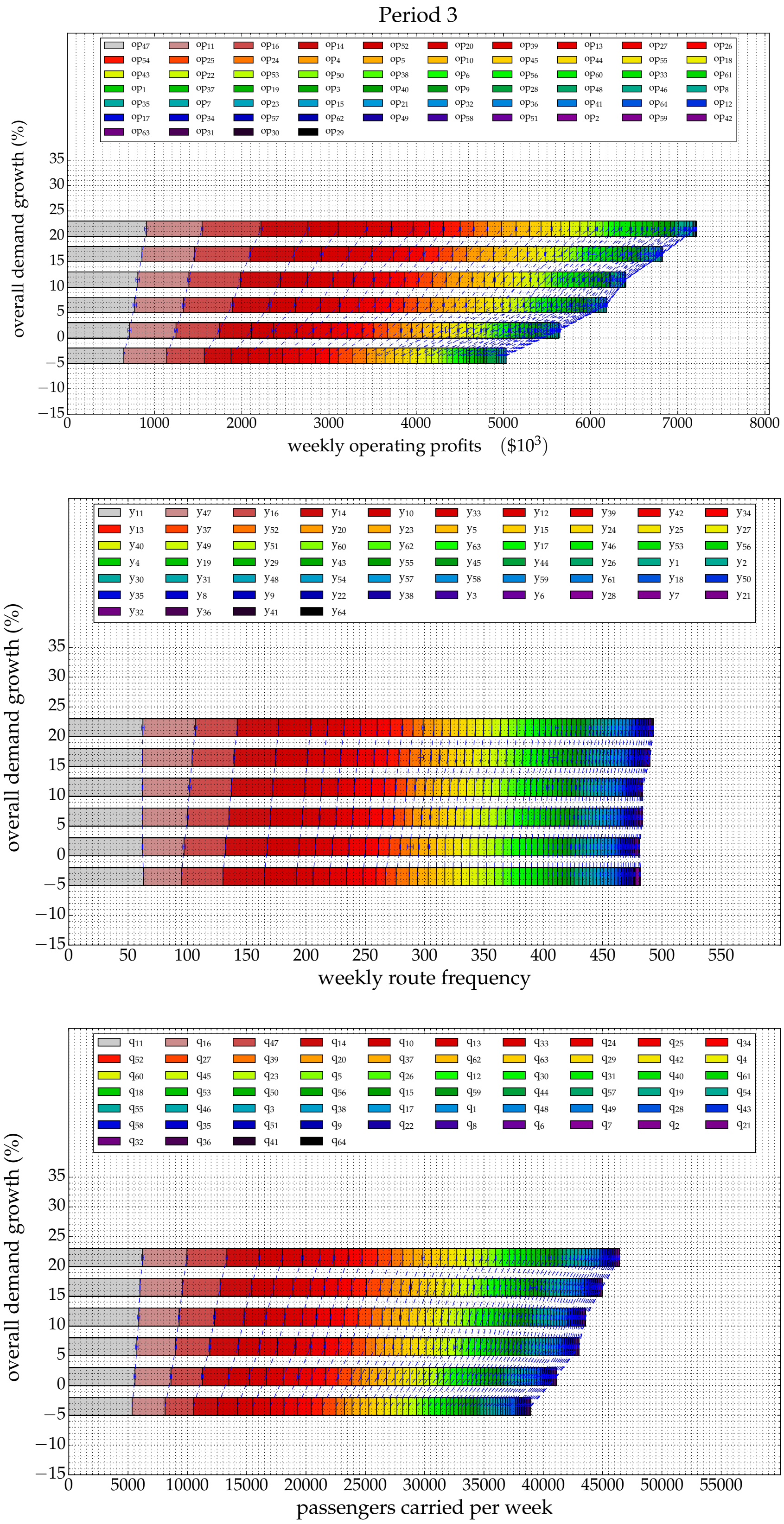


Figure D.11: Average weekly operating profits, route frequency and passengers carried per scenario in year 3

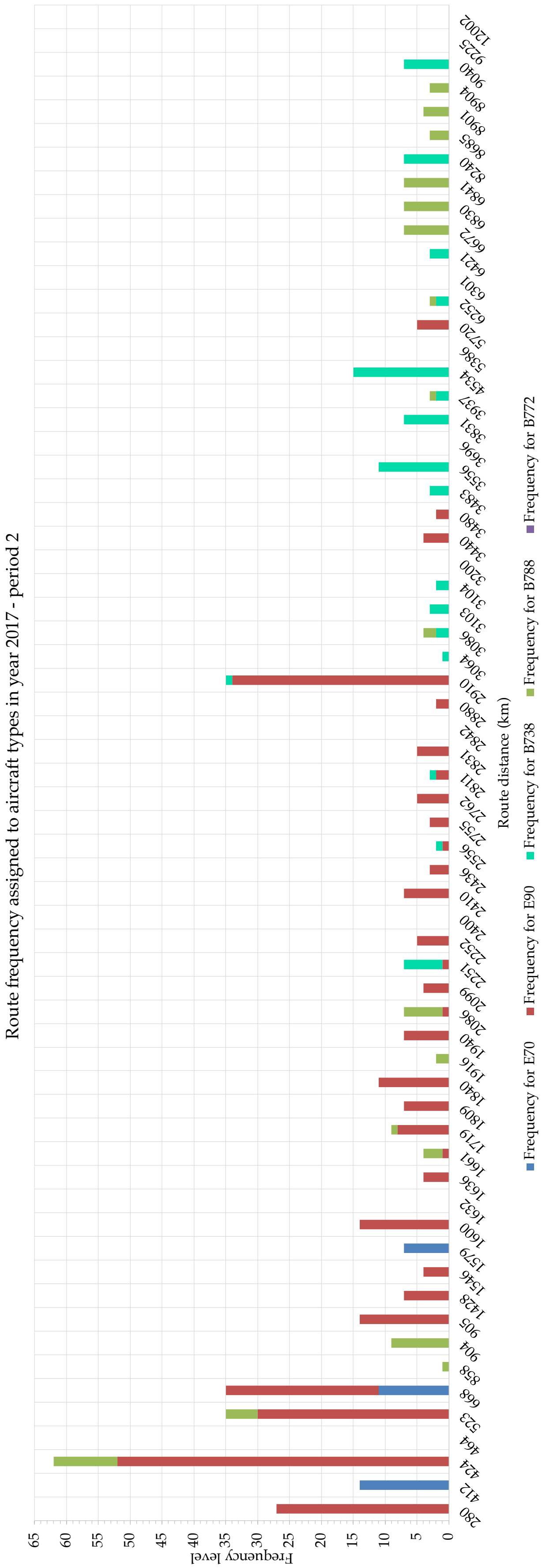


Figure D.12: Aircraft frequency assignments in function of route distance.

D.7. CPU performance

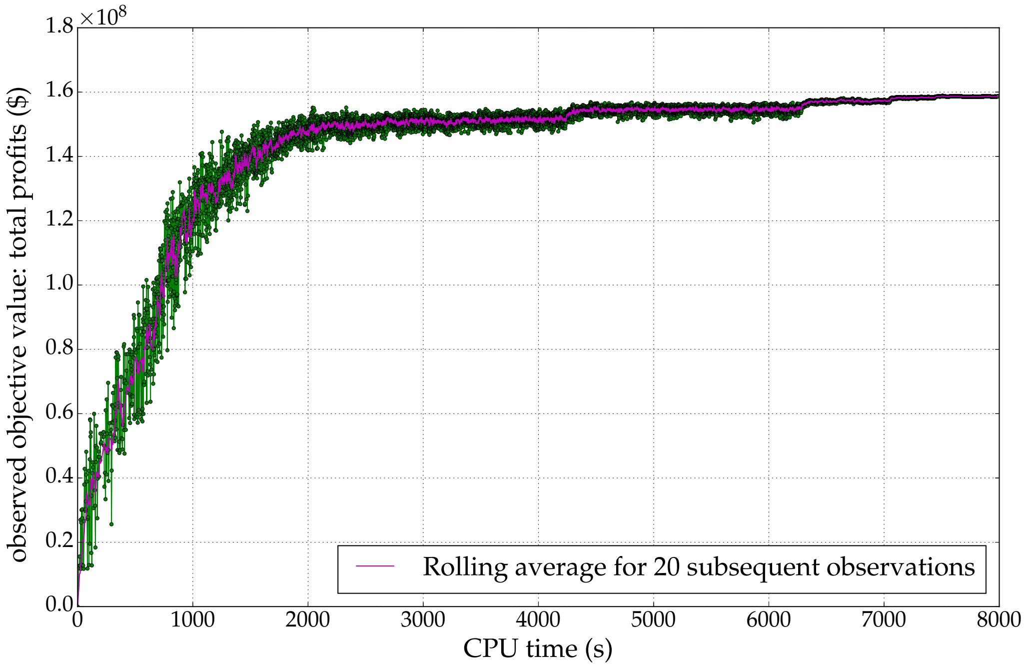


Figure D.13: 5-period scenario tree applicable to the KQ case study.

Bibliography

N. Adler. Competition in a deregulated air transportation market. *European Journal of Operational Research*, 129(2):337–345, 2001.

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

C. Barnhart, C.A. Hane, and P.H. Vance. Using branch-and-price-and-cut to solve origin-destination integer multicommodity flow problems. *Operations Research*, 48(2):318–326, 2000.

C. Barnhart, P. Belobaba, and A. Odoni. Applications of operations research in the air transport industry. *Transportation science*, 37(4):368–391, 2003.

T. Bartlett. An algorithm for the minimum number of transport units to maintain a fixed schedule. *Naval Research Logistics Quarterly*, 4(2):139–149, 1957.

M. Bazargan and J. Hartman. Aircraft replacement strategy: Model and analysis. *Journal of Air Transport Management*, 25:26–29, 2012.

P. Belobaba, A. Odoni, and C. Barnhart. *The global airline industry*. John Wiley & Sons, 2015.

D.P. Bertsekas. *Dynamic programming and optimal control*, volume 1. Athena scientific Belmont, MA, 1995.

D.P. Bertsekas and J.N. Tsitsiklis. Neuro-dynamic programming: an overview. In *Decision and Control, 1995., Proceedings of the 34th IEEE Conference on*, volume 1, pages 560–564. IEEE, 1995.

IAG British Airways. Annual report and accounts. *URL http://www. britishairways.com/cms/global/microsites/ba_reports0809/our_business/kpi3. html*, 2008-09.

A.S. Brooke, R.E. Caves, and D.E. Pitfield. Methodology for predicting european short-haul air transport demand from regional airports: An application to east midlands international airport. *Journal of Air Transport Management*, 1(1):37–46, 1994.

H. Courtney, J. Kirkland, and P. Viguerie. Strategy under uncertainty. *Harvard Business Review*, 12 1997.

DR Cox and PAWL Lewis. The statistical analysis of series of events. 1966.

M.P. Cristobal, L.F. Escudero, and J.F. Monge. On stochastic dynamic programming for solving large-scale planning problems under uncertainty. *Computers & Operations Research*, 36(8):2418–2428, 2009.

G.B. Dantzig. Linear programming under uncertainty. *Management science*, 1(3-4):197–206, 1955.

R.L. De Neufville and C.R. King. Access, fares, frequency: Effects on airport traffic. *Journal of Transportation Engineering*, 105(2), 1979.

W. Gelerman and R.L. De Neufville. Planning for satellite airports. *Journal of Transportation Engineering*, 99 (TE3), 1973.

I. Gertsbach and Y. Gurevich. Constructing an optimal fleet for a transportation schedule. *Transportation Science*, 11(1):20–36, 1977.

G.A. Godfrey and W.B. Powell. An adaptive, distribution-free algorithm for the newsvendor problem with censored demands, with applications to inventory and distribution. *Management Science*, 47(8):1101–1112, 2001.

G.A. Godfrey and W.B Powell. An adaptive dynamic programming algorithm for dynamic fleet management, i: Single period travel times. *Transportation Science*, 36(1):21–39, 2002.

G.A. Godfrey and W.B. Powell. An adaptive dynamic programming algorithm for dynamic fleet management, ii: Multiperiod travel times. *Transportation Science*, 36(1):40–54, 2002.

M. Hansen and T. Weidner. Multiple airport systems in the united states: current status and future prospects. *Transportation research record*, pages 8–17, 1995.

H. Heitsch and W. Römisch. Scenario tree modeling for multistage stochastic programs. *Mathematical Programming*, 118(2):371–406, 2009.

S. Hess, T. Ryley, L. Davison, and T. Adler. Improving the quality of demand forecasts through cross nested logit: a stated choice case study of airport, airline and access mode choice. *Transportmetrica A: Transport Science*, 9(4):358–384, 2013.

J.L. Higle. Stochastic programming: optimization when uncertainty matters. *Cole Smith J (ed) Tutorials in operations research*, pages 30–53, 2005.

C. Hsu, H. Li, S. Liu, and C. Chao. Aircraft replacement scheduling: a dynamic programming approach. *Transportation research part E: logistics and transportation review*, 47(1):41–60, 2011.

J. Huerta-Cepas, F. Serra, and P. Bork. Ete 3: Reconstruction, analysis, and visualization of phylogenomic data. *Molecular Biology and Evolution*, 33(6):1635, 2016. doi: 10.1093/molbev/msw046. URL +http://dx.doi.org/10.1093/molbev/msw046.

P.J.H. Hulshof, M.R.K. Mes, R.J. Boucherie, and E.W. Hans. Patient admission planning using approximate dynamic programming. *Flexible Services and Manufacturing Journal*, 28(1-2):30–61, 2016.

D. Julong. Introduction to grey system theory. *The Journal of grey system*, 1(1):1–24, 1989.

P. Kali and S.W. Wallace. *Stochastic programming*. Springer, 1994.

KQ Kenya Airways. Annual report and financial statements. *URL https://www.kenya-airways.com/about-us/investor-information/annual-reports/en/*, 2015.

KQ Kenya Airways. Annual report and financial statements. *URL https://www.kenya-airways.com/about-us/investor-information/annual-reports/en/*, 2016.

KQ Kenya Airways. Press release. *URL https://www.kenya-airways.com/about-us/investor-information/annual-reports/en/*, 2017.

H. Khoo and L.E. Teoh. An optimal aircraft fleet management decision model under uncertainty. *Journal of Advanced Transportation*, 48(7):798–820, 2014.

A.J. Kleywegt, A. Shapiro, and T. Homem-de Mello. The sample average approximation method for stochastic discrete optimization. *SIAM Journal on Optimization*, 12(2):479–502, 2002.

S. Lam, L. Lee, and L. Tang. An approximate dynamic programming approach for the empty container allocation problem. *Transportation Research Part C: Emerging Technologies*, 15(4):265–277, 2007.

G.F. List, B. Wood, L.K. Nozick, M.A. Turnquist, D.A. Jones, E.A. Kjeldgaard, and C.R. Lawton. Robust optimization for fleet planning under uncertainty. *Transportation Research Part E: Logistics and Transportation Review*, 39(3):209–227, 2003.

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

M. Lohatepanont. *Airline fleet assignment and schedule design: integrated models and algorithms*. PhD thesis, Massachusetts Institute of Technology, 2002.

M. Lohatepanont and C. Barnhart. Airline schedule planning: Integrated models and algorithms for schedule design and fleet assignment. *Transportation Science*, 38(1):19–32, 2004.

M. Naumann and L. Suhl. How does fuel price uncertainty affect strategic airline planning? *Operational Research*, 13(3):343–362, 2013.

G.L. Nemhauser and L.A. Wolsey. Integer programming and combinatorial optimization. *Wiley, Chichester. GL Nemhauser, MWP Savelsbergh, GS Sigismondi (1992). Constraint Classification for Mixed Integer Programming Formulations. COAL Bulletin*, 20:8–12, 1988.

T. H. Oum, A. Zhang, and Y. Zhang. Optimal demand for operating lease of aircraft. *Transportation Research Part B: Methodological*, 34(1):17–29, 2000.

G. Pantuso, K. Fagerholt, and S.W. Wallace. Solving hierarchical stochastic programs: application to the maritime fleet renewal problem. *INFORMS Journal on Computing*, 27(1):89–102, 2014.

G. Pantuso, K. Fagerholt, and S.W. Wallace. Uncertainty in fleet renewal: A case from maritime transportation. *Transportation Science*, 50(2):390–407, 2015.

D.J. Papageorgiou, Myun-Seok Cheon, G. Nemhauser, and J. Sokol. Approximate dynamic programming for a class of long-horizon maritime inventory routing problems. *Transportation Science*, 49(4):870–885, 2014.

W.B. Powell. Analysis of airline operating strategies under stochastic demand. *Transportation Research Part B: Methodological*, 16(1):31–43, 1982.

W.B. Powell. *Approximate Dynamic Programming: Solving the curses of dimensionality*, volume 703. John Wiley & Sons, 2007.

W.B. Powell, B. Bouzaiene-Ayari, C. Lawrence, C. Cheng, S. Das, and R. Fiorillo. Locomotive planning at norfolk southern: An optimizing simulator using approximate dynamic programming. *Interfaces*, 44(6): 567–578, 2014.

M.G.J. Repko and B.F. Santos. Scenario tree airline fleet planning for demand uncertainty. *Journal of Air Transport Management*, 2017.

H.R. Sayarshad and K. Ghoseiri. A simulated annealing approach for the multi-periodic rail-car fleet sizing problem. *Computers & Operations Research*, 36(6):1789–1799, 2009.

GJ. Schick and JW. Stroup. Experience with a multi-year fleet planning model. *Omega*, 9(4):389–396, 1981.

P. Schütz, A. Tomasgard, and S. Ahmed. Supply chain design under uncertainty using sample average approximation and dual decomposition. *European Journal of Operational Research*, 199(2):409–419, 2009.

N. Secomandi. Comparing neuro-dynamic programming algorithms for the vehicle routing problem with stochastic demands. *Computers & Operations Research*, 27(11):1201–1225, 2000.

A. Shapiro and A. Philpott. A tutorial on stochastic programming. *Manuscript. Available at www2.isye.gatech.edu/ashapiro/publications.html*, 17, 2007.

A. Shapiro, D. Dentcheva, et al. *Lectures on stochastic programming: modeling and theory*, volume 16. SIAM, 2014.

DP. Shube and JW. Stroup. Fleet planning model. In *Winter Computer Simulation Conference Proceedings*, pages 45–50. Society for Computer Simulation, 1975.

H.P. Simão, J. Day, A.P. George, T. Gifford, J. Nienow, and W.B. Powell. An approximate dynamic programming algorithm for large-scale fleet management: A case application. *Transportation Science*, 43(2):178–197, 2009.

H.P Simão, A. George, W.B. Powell, T. Gifford, J. Nienow, and J. Day. Approximate dynamic programming captures fleet operations for schneider national. *Interfaces*, 40(5):342–352, 2010.

R.W. Simpson. *A market share model for US domestic airline competitive markets*. Massachusetts Institute of Technology, Flight Transportation Laboratory, 1970.

R.S. Sutton and A.G. Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

D. Teodorović and E. Krcmar-Nozić. Multicriteria model to determine flight frequencies on an airline network under competitive conditions. *Transportation Science*, 23(1):14–25, 1989.

H. Topaloglu and W.B. Powell. Dynamic-programming approximations for stochastic time-staged integer multicommodity-flow problems. *INFORMS Journal on Computing*, 18(1):31–42, 2006.

Y. Wang, H. Sun, J. Zhu, and B. Zhu. Optimization model and algorithm design for airline fleet planning in a multiairline competitive environment. *Mathematical Problems in Engineering*, 2015, 2015.

W. Wei and M. Hansen. Impact of aircraft size and seat availability on airlines' demand and market share in duopoly markets. *Transportation Research Part E: Logistics and Transportation Review*, 41(4):315–327, 2005.

W. Wei and M. Hansen. Airlines' competition in aircraft size and service frequency in duopoly markets. *Transportation Research Part E: Logistics and Transportation Review*, 43(4):409–424, 2007.