Adaptable Optimization: Theory and Algorithms

by

Constantine Caramanis

A.B., Mathematics, Harvard University (1999)
M.S., Electrical Engineering & Computer Science,
Massachusetts Institute of Technology (2001)

Submitted to the
Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy
in Electrical Engineering and Computer Science

at the

Massachusetts Institute of Technology
[September 2006]
June 2006

© 2006 Massachusetts Institute of Technology. All rights reserved.

MASSACHUSETTS INSTITUTE
OF TECHNOLOGY
JAN 1 1 2007
LIBRARIES

Signature of Author ....
Department of Electrical Engineering and Computer Science
June 29, 2006

Certified by ....
Dimitris J. Bertsimas
Boeing Professor of Operations Research
Thesis Supervisor

Accepted by ....
Arthur C. Smith
Chairman, Committee on Graduate Students
Department of Electrical Engineering and Computer Science

ARCHIVES

Adaptable Optimization: Theory and Algorithms

by

Constantine Caramanis

Submitted to the Department of Electrical Engineering
and Computer Science on June 29, 2006,
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

Optimization under uncertainty is a central ingredient for analyzing and designing systems with incomplete information. This thesis addresses uncertainty in optimization, in a dynamic framework where information is revealed sequentially, and future decisions are adaptable, i.e., they depend functionally on the information revealed in the past. Such problems arise in applications where actions are repeated over a time horizon (e.g., portfolio management, or dynamic scheduling problems), or that have multiple planning stages (e.g., network design).

The first part of the thesis focuses on the robust optimization approach to systems with uncertainty. Unlike the probability-driven stochastic programming approach, robust optimization is built on deterministic set-based formulations of uncertainty. This thesis seeks to place Robust Optimization within a dynamic framework. In particular, we introduce the notion of finite adaptability. Using geometric results, we characterize the benefits of adaptability, and use these theoretical results to design efficient algorithms for finding near-optimal protocols. Among the novel contributions of the work are the capacity to accommodate discrete variables, and the development of a hierarchy of adaptability.

The second part of the thesis takes a data-driven view to uncertainty. The central questions are (a) how can we construct adaptability in multi-stage optimization problems given only data, and (b) what feasibility guarantees can we provide. Multi-stage Stochastic Optimization typically requires exponentially many data points. Robust Optimization, on the other hand, has a very limited ability to address multi-stage optimization in an adaptable manner. We present a hybrid sample-based robust optimization methodology for constructing adaptability in multi-stage optimization problems, that is both tractable and also flexible, offering a hierarchy of adaptability. We prove polynomial upper bounds on sample complexity. We further extend our results to multi-stage problems with integer variables in the future stages. We illustrate the ideas above on several problems in Network Design, and Portfolio Optimization.

The last part of the thesis focuses on an application of adaptability, in particular, the ideas of finite adaptability from the first part of the thesis, to the problem of air traffic control. The main problem is to sequentially schedule the departures, routes, ground-holding, and air-holding, for every flight over the national air space (NAS). The schedule seeks to minimize the aggregate delay incurred, while satisfying capacity constraints that specify the maximum number of flights that can take off or land at a particular airport, or fly over the same sector of the NAS at any given time. These capacities are impacted by the weather conditions. Since we receive an initial weather forecast, and then updates throughout the day, we naturally have a multistage optimization problem, with sequentially revealed uncertainty. We show that finite adaptability is natural, since the scheduling problem is inherently finite, and furthermore the uncertainty set is low-dimensional. We illustrate both the applicability of finite adaptability, and also its effectiveness, through several examples.

Thesis Supervisor: Dimitris Bertsimas
Title: Boeing Professor of Operations Research

Acknowledgments

My many years at MIT and the Laboratory for Information and Decision Systems (LIDS) leave me the most pleasant task of thanking the many people that marked my time here. First and foremost, I am deeply indebted to my advisor, Professor Dimitris Bertsimas. I have learned a great deal from our research collaborations. Equally importantly, through the years he has offered me guidance and advice that has been, and I am most certain will continue to be, greatly valuable to me both academically and personally.

Next, I owe thanks to the members of my thesis committee, Professors Sanjoy Mitter, John Tsitsiklis, and Pablo Parrilo. To Professor Sanjoy Mitter, I am very grateful, as he took a deep interest in my development from the day that I arrived at MIT. I leave MIT with great respect and admiration for his generosity, intellectual breadth, and eagerness to support young researchers, and to help them find their way. I have had the pleasure of knowing Professor John Tsitsiklis well before my time at MIT, and I am grateful for the interest he has shown, and the encouragement he has always given me, throughout my student days. I particularly appreciate his willingness to listen and offer his assistance on essentially whatever problem I was able to bring to him. I have been fortunate for the chance to work with Professor Pablo Parrilo at Caltech, at ETH, and then again at LIDS. I have greatly benefited from his expertise, and his willingness to share an energy that can only be described as boundless.

Academia is an inherently social endeavor. While at MIT and LIDS, I have had the chance to collaborate with many excellent researchers who have contributed a great deal to my development. I really enjoyed my collaborations with Michael Rosenblum on polyhedral-theoretic results in switching theory. Also thanks to Michael, I had the chance to work with Professor Michel Goemans, and Professor Vahid Tarokh, who both contributed to my academic development. I have had the great pleasure of a lasting collaboration (and friendship) with Shie Mannor on various topics in Machine Learning and Optimization. Shie is an expert in many areas, and I've learned much and benefited greatly from our interactions. In particular, I thank him for helping me learn game theory, learning theory, and the importance of maintaining an enduring relationship with one's butcher. In the Spring of 2005, a potential funding disaster

turned into an unexpected windfall, when Professor Moe Win very generously agreed to support me for a semester. This gave me the opportunity to work with Moe, in what I hope is the beginning of many fruitful future collaborations. In my last year at MIT I was funded by Lincoln Laboratory on a project on Air Traffic Control, where I benefited from a collaboration with Bill Moser, and Mark Weber. This was particularly helpful for Chapter 5 of the thesis.

My time at LIDS was also shaped by interactions with many others, a few of whom I would like to thank. I thank Professor Dimitri Bertsekas, who sparked my initial interest in Electrical Engineering, and also MIT and LIDS. I also thank my friend and officemate David B. Brown for the camaraderie, and the many exchanges of ideas, along with the many stadia and ga nuongs. Martin "LB" Wainwright deserves special thanks for teaching me what gitb is all about, in the early days when he became friend-number-one at MIT. I have also enjoyed many discussions about programming languages and compilers with him, and with Lijin Aryananda. I was especially lucky to have Ramesh Johari as my officemate. In addition to being a great friend, Ramesh was extremely encouraging in times when encouragement was most needed, and for this I thank him.

I'd also like to thank some other friends I've made along the way at MIT: Sekhar Tatikonda, Emin Martinian, and George Kotsalis. I also thank Rachel Cohen, Fifa Monserrate, and Doris Inslee, for their thoughtfulness, and remarkable ability to keep things running smoothly here at LIDS. There have been many others here at MIT to whom I am grateful, and I apologize for not mentioning them all here.

Lastly, and most importantly, I would like to thank my parents, Michael and Ileana, and my sister Christina; they have sustained me throughout this long process, giving love, encouragement, and friendship all in equal and overflowing portions. As they are in many ways the reason I was able to complete the PhD, they are also in many ways the very reason I began: my love and respect for the people they are, the things they have accomplished, and the example they have set, have helped me carve out my course. For all this and more, I will be forever grateful. But there is one more part to the story. One day towards the beginning of the PhD, my life changed when I met Mahshad Vakili. It's hard to imagine a life without her; thankfully, I don't have to. She is the warmth and beauty in my life, and I am thankful to her for everything she does, and everything she is.

This research was supported by a National Science Foundation Graduate Research Fellowship, by the Singapore-MIT Alliance, by a grant from Lincoln Laboratories, by the NSF under Grant ANI-0335256, and by the Army Research Office under award DAAD19-00-1-0466.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

The focus of this dissertation is dynamic (multi-stage) optimization, affected by uncertainty. Here, decisions are made sequentially over time, and the decision-maker has access to partial (perhaps noisy) observations of the uncertainty at each stage. The central theme of this thesis is the notion of adaptability: how do future stage decisions depend on (in a functional sense) the past uncertainty. The central questions we ask are: how adaptability is structured, how it impacts the tractability of the formulation, and most importantly, how we may exploit the knowledge of future stage adaptability, in order to implement a "better" first stage decision. In this sense, the principle of optimality, fundamental to the framework of Dynamic Programming, is central to the spirit of our approach. However, as we discuss in more detail below, the functional or structural similarities with Dynamic Programming, end there.

In order to tie the meaning of adaptability, uncertainty, and multiple stages to something concrete, let us consider the following basic formulation, to which we return again and again throughout this thesis. Consider then:

$$\begin{array}{rl} \min: & \boldsymbol{c}^\top \boldsymbol{x} + \boldsymbol{d}^\top \boldsymbol{y}_1(\omega_1) + \boldsymbol{f}^\top \boldsymbol{y}_2(\omega_1, \omega_2) \\ \text{s.t.}: & \boldsymbol{A}_0(\omega_1, \omega_2)\boldsymbol{x} + \boldsymbol{A}_1(\omega_1, \omega_2)\boldsymbol{y}_1(\omega_1) + \boldsymbol{A}_2(\omega_1, \omega_2)\boldsymbol{y}_1(\omega_1, \omega_2) \leq \boldsymbol{b}. \end{array} \tag{1.0.1}$$

This is a three stage optimization problem. There are three time periods, $\mathcal{T} = \{1, 2, 3\}$, and a decision is implemented at each stage: $\boldsymbol{x}$ at time 1, $\boldsymbol{y}_1$ at time 2, and $\boldsymbol{y}_2$ at time 3. The parameters ω_1 and ω_2 represent the uncertainty. The realization of these parameters controls the final realization of the parameters defining the optimization problem. The sequence of events is:

1a. Decision $\boldsymbol{x}$ is implemented.

1b. Uncertainty parameter ω_1 is realized.

2a. Decision $\boldsymbol{y}_1$ is implemented, after having observed $\boldsymbol{x}$ and ω_1.

2b. Uncertainty parameter ω_2 is realized.

3. The final decision $\boldsymbol{y}_3$ is implemented, after having observed $\boldsymbol{x}, \boldsymbol{y}_1$, and ω_1, ω_2. If the constraints are satisfied, the value of the problem is: $\boldsymbol{c}^\top \boldsymbol{x} + \boldsymbol{d}^\top \boldsymbol{y}_1 + \boldsymbol{f}^\top \boldsymbol{y}_2$.

Then the key definitions are:

* The **future stage decisions** are, in this case, decisions y_1, and y_2.

** What we refer to as **adaptability** throughout this dissertation, is the functional dependence of the future stage decisions on past realizations of the uncertainty: in this case, the functional dependence of y_1 on ω_1, and of y_2 on (ω_1, ω_2). This can be constant (no adaptability), affine, quadratic, other nonlinear, piecewise constant, etc.

*** The **uncertainty** for this problem is (ω_1, ω_2). Together, these two parameters completely specify the multistage optimization. Alone, parameter ω_1 provides partial information about the ultimate realization of the optimization problem. How we model uncertainty is one of the central concerns of this thesis.

Adaptability, as defined in (1.0.1) above, is particularly important when we have a system whose performance, as measured by the objective function and feasibility of the constraints, is a joint function of all the decisions and the full uncertainty realization. In (1.0.1), for instance, any statement about the feasibility of some first-stage decision x must also take into account the future stage decisions, as well as the potential realizations of the uncertainty, since the feasibility constraint links x to all these quantities. At a higher level, this is also the case, for example, in the Air Traffic Control problem we consider in Chapter 5. In this problem, the departure time, landing time, and flight path for commercial aircraft must be scheduled over time, so that the planes do not exceed the capacity constraints at airports or intermediate sectors of the National Air Space. The capacity constraints are impacted by the weather, and hence are uncertain. We get increasingly accurate information about the evolution of the day's weather, as the day progresses. As flights must be scheduled throughout the day, this is indeed a problem of sequential decision making, with sequentially revealed uncertainty. The performance metric of interest to us is the overall cost incurred, in terms of delay on the ground (ground holding) and in the air (air holding, and longer routes selected). This performance function is a joint function of all the actions implemented, in the sense that in isolation, the "goodness" of a single scheduling decision (aircraft A sent along route R at time T) cannot be evaluated.

In such problems, the notion of a "good" first-stage action depends on the future evolution of the system, i.e., the dynamics. This, in turn, depends on the nature of the uncertainty affecting the systems at the different stages, and also on the adaptability: how do future decisions depend on past realizations of the uncertainty. This thesis centers around these two concepts: uncertainty, and adaptability.

The natural applicability of adaptability and multi-stage optimization formulations extends far beyond applications to Air Traffic Control. Multi-stage optimization problems become relevant in many disparate frameworks. Sequential decision-making problems are natural in any scenario where decisions are made over time, and uncertainty is inherent in our imprecise forecasts about the future. In operations research, future demand or supply, including future service and setup times, as well as costs and profits, are examples where a single-stage approach must necessarily neglect an integral component of the problem. In addition to repeated rounds of decision-making indexed by time (as in a market environment), many processes naturally have decisions that must be made on different time scales. Often provisioning or design decisions are made prior to pricing, or scheduling decisions. This is the case in network design problems. Issues of timing here play a fundamental role in the structural properties of the problem formulations. Notions of feedback and control in engineering applications, from Internet congestion control, to two-stage manufacturing processes, are all fundamentally tied to adaptability and uncertainty. These are the central themes of this thesis.

Because of the widespread applicability, much work has been done in various communities. Sequential decision making in an uncertain (especially a stochastically uncertain) environment has traditionally been the subject addressed by Dynamic Programming (e.g., in the well-known texts of [19], [11], [114]) and for the discrete state-space setting, Markov Decision Processes (see, e.g., [109]). In the control theory context, sequential decision-making where uncertainty is revealed sequentially falls under the general heading of feedback control ([55], [67], [100]). Beyond high-level conceptual similarities, the approach in this thesis diverges from both the above views.

In single stage, deterministic optimization theory, the landscape of tractability (what classes of problems admit solution by tractable algorithms) is characterized by the geometric and topological notion of convexity. The success stories of Dynamic Programming, however, are much more dependent upon special problem structure, and the tractability of the solution is dependent on much finer structural properties. This tractability is typically a brittle property, and operations innocuous in single-stage optimization, such as adding convex constraints, can destroy special structure. Indeed, most problems addressed by Dynamic Programming are plagued by the well-known Bellman's Curse of Dimensionality, and finding an exact solution is often computationally hopeless. As a result, there has been considerable effort devoted to computing approximate solutions (see, e.g., [20]).

This thesis is more properly seen as an attempt to place traditional single stage optimization onto a dynamic framework, as opposed to seeking to make a contribution to approximate dynamic programming. Indeed, the starting point for this work is the single stage convex optimization problem.

■ 1.1 Uncertainty and Adaptability

Our point of departure is the single stage convex optimization problem, affected by uncertainty. In deterministic single stage optimization, the objective function, as well as the constraint functions, are known exactly to the decision-maker. In the context of perfect and deterministic knowledge of every parameter defining the problem, any multi-stage problem can be solved as a single stage problem without any loss of optimality. There is no information to be revealed, and thus all decisions can be fixed deterministically up front. In the face of uncertainty, however, the notions of multiple stages and adaptability become important. How we model the uncertainty, and how we model adaptability of future stages on past uncertainty, and the interaction of these two, is of central importance.

Models of Uncertainty

Throughout this thesis, we consider the setting where both the objective function and the constraint functions may be subject to some level of uncertainty. The description of the uncertainty that affects the constraint and objective functions, plays a crucial role in the formulation of even the concept of an "optimal solution" in the non-deterministic case. Even the notion of "feasibility" and what precisely one means by this, must be revisited, in the context of uncertainty in the optimization.

There are primarily two paradigms for dealing with uncertainty. The first, with the longer historical legacy, is that of stochastic optimization (see [81],[108], [35], and the references therein). Here, the uncertainty is assumed to have a stochastic nature. An explicit description of the stochastic uncertainty may or may not be available to the decision-maker, but nevertheless, the behavior is stochastically driven by some distribution. Uncertain constraints can then be recast as soft constraints where violation is penalized; or, one may ask that the constraints be satisfied with high probability, as in the so-called chance constraint framework. Further details of these models can be found in Chapter 2.

Robust optimization ([125],[14], [15], [17], [26], [25],[28], [46], [71]) has attracted much attention, particularly in the last decade, as an alternative modeling approach to stochastic optimization. In the robust optimization paradigm, uncertainty is not assumed to have an underlying distribution, but rather is chosen in a worst-case manner (one can imagine a malicious adversary) from a bounded set (see Chapter 2 for more details). The description of the set is known *a priori* to the decision-maker. We note that this is not the same as the stochastic optimization viewpoint with the uniform distribution. Robust optimization is inherently an analysis of the worst-case realization of the uncertainty, where "worst-case" is understood to be with respect to the given bounded set of possible realizations. A question of central theoretical and also practical importance, is to understand when one formulation might have advantages over another.

Despite the fact that the stochastic and robust optimization approaches often provide structurally different solutions to similar problems, there should be no deep cultural divide between the two. Recent work, and this thesis is no exception, seeks to exploit the benefits of each, including both modeling advantages, and also tractability advantages. This thesis examines multi-stage optimization problems under both models for the uncertainty.

A question that is implicit in this work, is what are the basic primitives of particular formulations of optimization under uncertainty. Stochastic optimization takes as a primitive the knowledge that a data-generating distribution exists, and some form of explicit knowledge of, or access to, the distribution itself. This can range from an analytical expression of the distribution, to a black box that provides sample realizations (and this may include historical realizations). In robust optimization, the primitives amount to a specification of the uncertainty set (for a viewpoint that seeks to connect data and uncertainty sets in an explicit manner through the machinery of risk measures, see [41]). The structure of the uncertainty set, much like the structure of the distribution in the stochastic optimization framework, largely determines both the quality of the solution, and, importantly, the tractability of the formulation. Chapter 3 works within the robust optimization framework, where the uncertainty sets defining the problem uncertainty are specified *a priori* and explicitly, to the decision-maker. In Chapter 4, we take a different perspective, and assume that the decision-maker only has access to a mechanism that can generate independent sample realizations of the uncertainty. Possible extensions to the case where each sample comes at a cost, and may in fact be noisy, are mentioned in Chapter 6. As further discussed in Chapter 4, there does not seem to be a clear understanding of any separation principle between estimation and optimization. Given a finite data sample, it is well known that multivariate integration is hard in the sense that the sample complexity may be quite large (see, e.g., [132]), and even approximating the volume from uniformly drawn samples can be intractable without more sophisticated rapidly mixing Markov chain techniques ([89], [83]) that require special problem structure. This all points to a need for methods that use data directly in the optimization problem. This is the approach of recent sample-based optimization approaches ([43],[51], [92]) and also the point of view we take in Chapter 4. In these approaches, the samples are incorporated directly in the optimization problem. In a sense, by incorporating the data directly into the optimization problem, there is some implicit estimation of the distribution, but specifically within the context of the optimization problem. In Chapter 4 and Chapter 6 we show that in multi-stage optimization problems, the structure of future stage adaptability plays an important role in this implicit estimation process. We believe that this is an exciting, but barely explored area that deserves much more consideration.

Adaptability in Optimization

In a multi-stage model where uncertainty is revealed sequentially, the decision-maker can naturally adjust future decisions to depend on the uncertainty realization revealed. We model this using adaptability, that is, explicitly building in a dependence of future actions on the past realizations of the uncertainty. We see this in a functional expression in (1.0.1): adaptability refers to the nature of the functional dependence of y_1, y_2 on ω_1, ω_2. As we discuss in detail in the sequel, we can have affine adaptability (y_i are affine in ω_i) and similarly quadratic adaptability, or other forms of adaptability. We refer to a static multi-stage problem as one without adaptability, i.e., where the decision-maker commits to all the decisions up front at the first stage, and implements them without recourse to the realization of the uncertainty in later stages. In the language of control theory, we can think of this as the difference between a closed loop system (we have adaptability to feedback) and an open loop system (no adaptability).

In this thesis we are concerned with different adaptability schemes. That is, we consider different functional structures for the dependence of future decisions on past uncertainty, including the benefit of adaptability, complexity, and algorithms for building it. At a high level, increasing the level of adaptability (notions such as the "level" of adaptability are made precise below) typically benefits the decision-maker. There are two effects here that are of interest. Certainly, future actions that are made in response to revealed uncertainty must be at least as good as actions taken without any knowledge of past uncertainty realizations; more subtle, is the effect on the first stage decision. Regardless of the adaptability scheme used, the decision-maker has access to the same information when it comes to implementing the first-stage decision. In systems whose behavior is not separable, in the sense that performance is assessed after all actions are made, and all uncertainty is realized, the first-stage decision cannot be evaluated in isolation of the future evolution of the system. In such situations, the very "knowledge" that future stage decisions are adaptable and thus are functions of the realized uncertainty, can change the structure of the first stage decision. In particular, in systems where feasibility is of primary concern, adaptability in future stages allows the decision-maker to be less conservative in the first-stage action.

There are several themes here that are emphasized throughout the thesis. First, is the benefit of adaptability comes in terms of the performance improvement obtained in the later stages where in the adaptable framework there is a chance to use information about the uncertainty revealed by the past realizations, as opposed to the non-adaptable, or static case, where that information is not used. This perspective emphasizes the value of information to the optimization. This theme is particularly important in Chapter 3 where we consider finite schemes for adaptability, and thus naturally have a hierarchy of increasing adaptability. Indeed, the starting point of this work may be taken to be the single-stage uncertain optimization problem, where the decision-maker has access to some side information (in a sense made precise in Chapter 3) about the uncertainty before it is realized. In this context, then, the value of adaptability is linked to the value of information in uncertain optimization.

The second important theme stressed above, is the value of adaptability not to the performance of future stages of the optimization, but to the first stage: the improvement that the very knowledge of future adaptability makes possible for the first-stage decision. In many applications, such as the Air Traffic Control scheduling application we discuss in Chapter 5, or Portfolio Management ([13], [73],[60]) to name but two, we are able to implement the optimization scheme in a folding horizon manner. Consider a T-stage multi-stage optimization problem with some uncertainty realization revealed at each stage. The *folding horizon* approach proceeds as follows: we solve the multi-stage optimization problem with some (possibly none) level of adaptability, we implement only the first stage solution, and then we re-solve the resulting $(T-1)$-stage optimization, updated to reflect the revealed realization of the uncertainty. This process continues through to the final stage.

We have defined several different approaches to a multi-stage optimization: (1) The static approach, where all decisions are made at the initial stage and have no adaptability to realizations of the uncertainty, and furthermore, all these initially computed actions are sequentially implemented; (2) The static folding horizon approach, where the decision-maker computes the static solution, but only implements the first-stage decision, and then re-solves the static problem with the updated information at the next stage; (3) The adaptable solution (with some adaptability scheme, i.e., some functional form specified for the $\boldsymbol{y}_i(\cdot)$), and (4) The adaptable solution implemented in a folding horizon framework, analogous to the static folding horizon scheme. While schemes (3) and (4) are not well-defined without specifying precisely the type of adaptability implemented, for the purpose of discussion let us assume that we are minimizing an objective function, and let us refer to the value of the optimization of scheme (i) as Z_i. Then certainly the relations:

$$Z_1 \leq Z_2 \leq Z_4,$$

and

$$Z_1 \leq Z_3 \leq Z_4,$$

hold generically. But there is no ordering of Z_2 and Z_3, as it depends on the application, and also on the level of adaptability (of course, if we have arbitrarily rich adaptability, then by definition $Z_3 = Z_4$). These ideas are made concrete in the following simple example.

An Example: Static Solution, Folding Horizon, and Adaptability

We consider and work through a simple stylized example, to emphasize the differences between the four schemes of adaptability described above, and to obtain some intuition about the inequality relations between the Z_i.

We consider a simple portfolio optimization example. Suppose we have $1 to build a portfolio with two stocks, and a third risk-free asset which we can think of as cash. The objective is to maximize the expected value of the portfolio after $T = 3$ stages. Let our decision at stage l be the amount of stock j to buy or sell. The uncertainty here is the return on each of the two stocks at the end of the T (three) investment periods. At the beginning of stage 2, the returns of the past period are known to the investor, and similarly at the beginning of stage 3. At the beginning of stage 1, only some uncertainty description of the returns is known to the decision-maker. For this example, we model the uncertainty as probabilistic in nature, and suppose that the objective is to maximize the expected value of the portfolio at the final stage. Letting K^l be the wealth at time l, K_0 the initial wealth, x_i^t the fraction of the portfolio invested in asset i at time t, and r_i^t the return rate on asset i at time t, we have:

$$\begin{aligned}
\max: \quad & K^T \mathbb{E}\left[\sum_i r_i^T x_i^T\right] \\
\text{s.t.}: \quad & K^t = K^{t-1}\left(\sum_i r_i^{t-1} x_i^{t-1}\right), \quad t = 1, \ldots, T \\
& \sum_i x_i^t = 1, \quad t = 1, \ldots, T.
\end{aligned}$$

The expectation is over the returns for each asset and for each period. This model is nonlinear, but we use it only for the purpose of the current discussion. We take up a similar problem in Chapter 4, which we formulate as a linear optimization, and consider various different structures for uncertainty and adaptability.

We consider next three scenarios:

1. Independent returns: Suppose that the returns at period t, t' are independent for $t \neq t'$. In this case, it is not difficult to see that adaptability is of no use. Since we know the statistics (not the realization) of the returns at each stage from the beginning, and because of the independence assumption, we have no additional useful information at time t that we did not have at time $t-1$, or time $t = 0$. Therefore we have:
$$Z_1 = Z_2 = Z_3 = Z_4.$$

2. Dependent returns: Consider next the opposite extreme where the returns over time are extremely correlated. Suppose that at the first stage, the returns on the two stocks will be $(r_1^1, r_2^1) =\in \{(1/2, 1.2), (1.2, 1/2)\}$ with equal probability, and then the future returns (r_1^2, r_2^2) and (r_1^3, r_2^3) are both equal to the first-stage returns. Suppose that the risk-free asset (cash) has return equal to 1. Then the solution for the no-adaptability (static) formulation will be to hold cash for all three periods, since the expected payoff of either of the stocks is 0.85 for a single period, and 0.9265 for three periods, and hence less than cash. The adaptable solution, on the other hand, will be to invest everything in cash for the first time period, and then in the subsequent two periods, to invest in the asset with the guaranteed return of 1.2. It is clear, then, that $Z_1 < Z_4$. What about Z_2, the fold-

ing horizon static solution? We see that the first-stage solution of the adaptable solution coincides with the first-stage of the static solution. Therefore after the first stage, the static-optimized portfolio is identical to the adaptable portfolio. Furthermore, from here we can see that the static solution for the last two stages of the optimization yields the same solution as the adaptable solution. Therefore $Z_2 = Z_4 = 1 \times 1.2 \times 1.2 = 1.44$ in this example: the static folding horizon problem is as good as the optimal adaptable solution.

3. Stage-to-stage Constraints: Consider now the above problem, again with the same dependence of returns among stages, so that after the first stage the profitable stock is identified. Consider the additional constraints that there is a transaction cost for selling or buying stocks. For the purposes of illustration, consider the extreme situation of no transaction costs for trading stocks, but extremely high transaction costs for converting to or from cash. The static solution again places the entire portfolio in the risk-free asset, cash, for all periods. In the folding horizon approach, however, in the second and later stages, the static solution no longer moves the assets in the portfolio to the profitable stock (recall that after the first stage, the profitable stock is revealed, and thereafter yields a guaranteed 1.2 return) because of the prohibitive transaction costs. The adaptable strategy, on the other hand, splits the portfolio among the two stocks, and in the second and third period moves everything to the profitable stock. After the first period, the value of the portfolio is $0.5 \times (0.5+1.2) = 0.825$, which after two more periods in the profitable stock becomes: $0.825 \times 1.44 = 1.224$.

Remark 1.1
It turns out that the notion of independence is critical. There is also set-theoretic expression of independence, which is of central importance in adaptability formulations within the context of robust optimization. This observation plays a key role in the development of the material in Chapter 3.

Remark 1.2
In the final portion of the example above, we see that the static solution diverges from the adaptable solution in the first stage. It makes a "mistake" from which it cannot recover in future stages. We see in the body of this thesis that this intuitive phenomenon is generic. Without adaptability, the static solution tends to make overly conservative decisions in the first stage, from which it can then not subsequently recover in the context of folding horizon.

A Second Example: Deterministic Uncertainty

In this example we illustrate the above concepts with a simple geometric example under a deterministic uncertainty model. The model is simple, but we revisit it in

Chapter 3 to gain intuition about finite adaptability in optimization. Consider a simple budget allocation problem, where we seek to minimize a budget, x, subject to the constraint that it is sufficient to fund two projects, where project i requires resources at least ω_i, $i = 1, 2$.

$$\begin{aligned} \min: \quad & x \\ \text{s.t.}: \quad & x \geq y_1 + y_2 \\ & y_i \geq \omega_i, \quad i = 1, 2. \end{aligned}$$

Suppose now that the individual project costs, ω_1, ω_2, are not known exactly. Nevertheless, we must commit to the budget today, while the allocation of the funds to the two projects can be postponed until the realization of the budget requirements for each budget. Thus this is a two-stage problem, the first-stage decision is x, and the second stage decisions are (y_1, y_2). The uncertainty is in the (suggestively named) vector $\boldsymbol{\omega} = (\omega_1, \omega_2)$. For this example, we consider the robust optimization paradigm where the uncertainty is deterministic and set-based. That is, we require that we satisfy the inequality:

$$y_i \geq \omega_i, \quad i = 1, 2,$$

for every realization of (ω_1, ω_2) in some uncertainty set Ω. Consider two different uncertainty sets Ω:

$$\begin{aligned} P_1 & \triangleq \{(\omega_1, \omega_2) \,:\, 0 \leq \omega_1, \omega_2 \leq 1, \ \omega_1 + \omega_2 \leq 1\} \\ P_2 & \triangleq \{(\omega_1, \omega_2) \,:\, 0 \leq \omega_1, \omega_2 \leq 1\}. \end{aligned}$$

These are pictured in Figure 1-1. For Ω_1, the optimal static solution is $x = 2$, $(y_1, y_2) =$

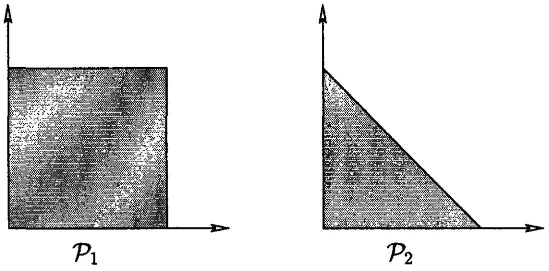

Figure 1-1. This figure shows the two uncertainty sets Ω_1 and Ω_2 in the second example above. Analogously to the first example, Ω_1 here corresponds to the "dependent" uncertainty case, while Ω_2 corresponds to the "independent" case. As in the stochastic example, adaptability is of no benefit in the face of independence.

$(1, 1)$. The optimal adaptable solution, however, cuts the budget in half, as the solution is: $x = 1, (y_1, y_2) = (\omega_1, \omega_2)$. This is feasible since if $(\omega_1, \omega_2) \in \Omega_1$, then we have $\omega_1 + \omega_2 \leq 1$. Note that in this case, $Z_1 = Z_2$, that is, the folding horizon approach cannot reduce the objective function. This is because, similarly to the third part of the portfolio example above, the static solution has already forced the decision-maker to implement a first-stage solution of $x = 2$, and since this determines the value of the optimization, even with a folding horizon approach, the decision-maker cannot recover from the pessimism (or overly conservative nature) of the first-stage static solution.

For Ω_2, however, the picture is different. The optimal static solution is again $x = 2, (y_1, y_2) = (1, 1)$, but now the optimal adaptable or dynamic solution becomes $x = 2, (y_1, y_2) = (\omega_1, \omega_2)$, and thus the costs coincide. In this case adaptability has nothing to offer, and the reason is essentially the independence (in the geometric rather than probabilistic sense) of ω_1 and ω_2. This phenomenon turns out to be rather general, and is discussed in further detail in Chapter 3. ▲

The Connection to Uncertain DP and MDP

It is worth briefly discussing the difference in perspective between the work of this dissertation, and the line of research pursued in the context of uncertain DP and MDP formulations. There has recently been an extension of the MDP and Dynamic Programming framework to the case where the problem specification itself (e.g., the rewards, or transition probabilities) has uncertainty. In [80], robust Dynamic programming is considered, and under certain conditions, the author shows that one can recover the familiar structural results of Dynamic Programming, in particular the recursion for backward Dynamic Programming. In [96], the authors consider the case of MDP with uncertain matrix transition probabilities, and are also able to recover many of the structural results of DP and MDPs. Finally, in [78], the authors consider an application of parametric programming to robust optimization, and they then show that this can be applied to the case of MDPs with uncertain rewards, and uncertain transition probabilities. However, these approaches endow the uncertainty with an independence property similar to the one illustrated in Figure 1-1. In the context of the DP and MDP problems considered in those references, this independence property essentially amounts to a condition that ensures that future actions do not benefit from explicitly incorporating any functional dependence on past realizations of uncertainty. Thus, this independence property is crucial to restore Markovianity to the problem, without which in their formulations, obtaining tractable models does not seem possible. Ultimately, however, the contribution of those works is the derivation and application of results in single-stage uncertain optimization, to DP and MDPs. The focus of this dissertation, therefore, is complementary to these results; here our primary concern is the adaptability of future stage decision on past realizations of the uncertainty.

■ 1.2 Thesis Outline and Contributions

This thesis seeks tractable extensions of uncertain optimization to the multi-stage horizon. Primarily, this is done by considering different models for the uncertainty affecting the problem, and for the adaptability. Indeed, the central theme of this thesis is structured adaptability. As is generally known, uncertainty and adaptability can quickly render many problems intractable. Linear problems that have the nicest tractability properties in their single-stage deterministic form, can be intractable (NP-hard) even for two-stage formulations. A first explanation for this difficulty is that the optimization is no longer over the space of decisions, but rather over the potentially much larger space of policies. This is indeed a primary source of the added computational difficulty. However it does not tell the full story. As the following simple example shows, even in a two-stage problem when the optimal adaptability (i.e., the second stage decision variable as a function of the first-stage uncertainty) is explicitly known to the decision-maker, and is linear in the uncertainty and hence not very high-dimensional, the resulting problem may nevertheless be quite difficult. The complexity is determined not just by the adaptability scheme, but also by its interaction with the model for the uncertainty. The robust optimization paradigm results in a tractable formulation whenever a particular subproblem is tractably solvable (we discuss this in full detail in Chapter 2). In multi-stage problems, the structure of this subproblem is affected by the structure of the adaptability, and the uncertainty set.

Example: Intractable Linear Adaptability

Consider the problem (see [12]):

$$\begin{aligned} \min: \quad & x \\ \text{s.t.}: \quad & x \geq \boldsymbol{y}^\top \boldsymbol{\omega} \\ & \boldsymbol{y} = \boldsymbol{Q}\boldsymbol{\omega}, \end{aligned} \tag{1.2.2}$$

where $\boldsymbol{\omega}$ is an uncertain parameter. In the robust optimization formulation, if the uncertainty set for $\boldsymbol{\omega}$ is $\Omega = \{\boldsymbol{\omega} \, : \, a_i \leq \omega_i \leq b_i\}$, i.e., a box set, then one can easily see that problem (1.2.2) is equivalent to the indefinite quadratic optimization over the box:

$$\begin{aligned} \max: \quad & \boldsymbol{\omega}^\top \boldsymbol{Q}\boldsymbol{\omega} \\ \text{s.t.}: \quad & \boldsymbol{\omega} \in \Omega. \end{aligned}$$

It is well-known (see, e.g., [69]) that maximizing an indefinite quadratic subject to box constraints, is NP-hard. ▲

Thus this thesis focuses on the uncertainty affecting optimization problems, the adaptability we build in to future stages, and the interaction of the two. This interaction extends to multiple levels, in the sense that it affects tractability, feasibility, and performance (i.e., value of the objective). Chapters 3 and 4 of this thesis consider different structures for adaptability, under different models for the uncertainty. The focus is on understanding when adaptability can benefit the decision-maker in the sense of both performance (value) and feasibility of the optimal solution, as well as tractability of the optimization formulation.

Contributions of this Thesis

Here we summarize the primary contributions of this thesis.

I. We propose a finite, hierarchical formulation for adaptability in two-stage linear optimization problems under a deterministic formulation for uncertainty.

   (a) This proposal allows us to accommodate integer second stage variables, the first proposal (to the best of our knowledge) that allows this extension.

   (b) Because of the inherent finiteness of this proposal, it presents the decision-maker with a trade-off of increased adaptability versus the cost of computing and implementing this finite adaptability. This is of particular interest in the case where the computed adaptability is actually implemented in the later stages, as opposed to the folding horizon approach where a new solution may be computed at a later point in time.

II. We propose a sample-based approach to structuring a hierarchy of adaptability. This moves us away from the deterministic paradigm, and more towards the question of data driven optimization. A central question is how data is best used in optimization.

   (a) By using sampling techniques, we circumvent intractability issues, and develop a framework for structuring a hierarchy of nonlinear adaptability. This yields sample complexity that is polynomial in the size of the problem, and polynomial in the number of stages of the problem. To our knowledge, this is the first such proposal.

   (b) This hierarchy of adaptability can also accommodate discrete variables.

   (c) We provide sample complexity estimates, using some convexity driven results of [43], and also results from learning theory.

III. We consider an application of the ideas of this thesis, in particular those of Finite Adaptability, to the problem of Air Traffic Control. We provide a formulation that captures the dynamic nature of the problem, as well as the weather uncertainty. As we illustrate in several examples, this is a natural application of adaptability.

■ 1.2.1 Chapter 2: Background Material

The main foundations of this dissertation lie in robust optimization, stochastic optimization, and Statistical Learning Theory. In this Chapter, we review some basic facts, results, and techniques from these three areas. The results in this chapter are from the literature, and we give numerous references to point out what has been done, where, and when. In addition to providing the background on which the remainder of the thesis is based, this chapter serves to place the contents and contributions of this dissertation in the proper context of existing research.

While we provide a review of robust and stochastic optimization, and also some results of Statistical Learning Theory, we save the more specialized material review for briefer sections in the chapters where these results are needed. The purpose of this chapter, then, is to establish a common footing and context, upon which the rest of the thesis is built.

We review some of the techniques and results of stochastic optimization. While this thesis is more motivated from the perspective of robust optimization (including Chapter 4 when we rely on probabilistic techniques), it is important to understand not only the main problems and techniques of stochastic optimization, but also where it has been successful, and where it has been less so, especially in contrast to robust optimization. We review techniques of complete recourse, where uncertain constraints are brought to the objective and penalized, and then the resulting objective becomes to minimize the expected penalty incurred. We also introduce and then review the basic results of Chance Constraints.

This motivates the review of some recent results that take a statistical learning theory view of approximating Chance Constraints. We refer to these as sampling methods, and we review some of the recent literature in this area. This is particularly important for what is to come in Chapter 4. This also gives us an opportunity to review some of the basic results that we use from Statistic Learning Theory. In this chapter, we go over only the basics that we require, deferring some of the more notationally intensive material to Appendix B.

■ 1.2.2 Chapter 3: Finite Adaptability

This Chapter launches our consideration of the value of adaptability to multi-stage optimization problems. In this chapter, we focus exclusively on the robust optimization paradigm for dealing with uncertainty; that is, we assume that the uncertainty has a deterministic set-based description. While some of the structural results do not demand this specialization, for many of the algorithmic results we require the uncertainty set to be polyhedral and defined by its extreme points (extensions are considered in Chapter 4).

We consider adaptability functions that are piecewise constant. One of the primary motivations for this is the desire to develop an adaptability framework that can accommodate discrete second-stage variables. In this case, the adaptability must be piecewise constant.

We consider the case of adaptability with a small number of pieces. The main theoretical challenge lies in partitioning the uncertainty set into these finitely many pieces. This turns out to be a difficult problem. We show that it is NP-hard to optimally partition a set into even two pieces. Nevertheless, using duality, we are able to obtain necessary conditions that any good partition must satisfy. We do this by exploiting some geometric consequences of the robust optimization formulation. Because robust optimization is inherently a "worst-case" approach, the decision-maker immunizes the optimal solution against several worst-case scenarios that could never be realized simultaneously. It turns out that for the case of linear optimization under uncertainty, we can use duality to identity what these worst-case scenarios are. We then select partitions that explicitly separate these "bad" scenarios.

Furthermore, in this chapter, we discuss the relationship of finite adaptability with affine adaptability, a continuous adaptability scheme proposed initially under the name of "linear decision rule" in the stochastic optimization literature (e.g., [110]) that has more recently appeared again in [12] and [47]. We show that the two proposals are complementary in the sense that neither dominates the other. We give an example where affine adaptability is no better than having no adaptability, while finite adaptability with just three pieces outperforms the no adaptability case considerably. Likewise, we give an example where regardless of the number of pieces of the finite adaptability scheme, no improvement is obtained over the no adaptability case, while the affine adaptability gives the optimal solution.

One of the difficulties in this chapter, is the failure of finite adaptability to readily extend to multiple stages, without causing a combinatorial explosion in the number of variables. Indeed, for the T-stage problem, even if a partition of the uncertainty of each stage is provided (thus rendering the 2-stage problem trivial) the number of variables is exponential in T. A different framework is required, to be able to address multi-stage problems, and at the same time providing a hierarchy of adaptability (as opposed to a single level, as in the affine adaptability proposals) while maintaining polynomial complexity in the number of stages. This is the focus of the next chapter.

■ 1.2.3 Chapter 4: Adaptability via Sampling

This chapter marks a departure from the techniques and setup of Chapter 3 in that we assume now that the uncertainty does have a stochastic nature. However, we assume that we do not have access to this distribution, except for our ability to generate independent and identically distributed samples from the distribution. Thus we treat the uncertainty as generated from a black box. In addition to this change, our objective is also different. Unlike many stochastic optimization formulations, we are not interested in minimizing the expected penalty. Rather, we seek a solution that guarantees feasibility with high probability. In this, our approach is closer to the sampling work of Calafiore and Campi ([43]) and de Farias and Van Roy ([51]).

There are two central observations that motivate the results in this chapter. First, we observe that within the robust optimization framework, affine and higher order adaptability formulations (not to mention more general nonlinear adaptability) are typically intractable precisely because of the solution to the subproblem (as explained in Chapter 2). Yet for any particular realization of the uncertainty, this subproblem reduces to a simple linear constraint. This continues to be true for any nonlinear mapping of the uncertainty, that leads to adaptability that is affine in the decision variables (although not necessarily in the uncertainty parameter itself). Furthermore, by controlling the structure of these nonlinear functions, for example by imposing polynomial structure of fixed degree on future stage adaptability, we are able to prove polynomial upper bounds on the sample complexity required to provide particular reliability and feasibility guarantees. Thus, by combining sampling ideas with structured adaptability, we obtain a polynomial time method for structuring higher order adaptability in multi-stage optimization.

We also show that these ideas extend to problems with integer variables in future stages. Here, of course, the resulting problem remains discrete, so we cannot hope for tractability guarantees. Nevertheless, we are able to provide feasibility and reliability guarantees, and the number of variables increases in a controlled manner. This gives a way to structure piecewise constant adaptability without explicitly constructing the regions of the partitions, as was the focus of Chapter 3.

In this chapter, we also consider a more sophisticated treatment of some results from statistical learning theory, and show that using more careful complexity measures such as the so-called fat-shattering dimension, or more generally, covering numbers, it is possible to obtain improved upper bounds on sample complexity. We also use this machinery to introduce a robustness parameter with respect to the sampling process, that itself has an explicit trade-off with the reliability and feasibility parameters in the expression for the upper bounds on sample complexity.

Finally, we introduce a feasibility maximization problem, and show that in addition to the performance improvement obtained from introducing adaptability, there can be feasibility improvements, perhaps contrary to what the upper bounds on sample complexity are able to predict. We support this with some simple examples from a two-stage and three-stage network design problem. As we discuss further in Chapter 6, this suggests a Structured Risk Minimization approach to multistage optimization under uncertainty.

■ 1.2.4 Chapter 5: Air Traffic Control

In this chapter, we consider the application of our work on multistage optimization and adaptability, to the problem of Air Traffic Control. In particular, we focus on the application of the results from Chapter 3. The problem we consider here is the global problem of scheduling all the flights that take place in the continental United States in a 24 hour period. The decisions include how to assign ground delay and air delay to each flight, and moreover what routes each flight should use from its departure airport to its destination. These decisions must be made with the goal of minimizing a cost function which relates directly to the delays experienced by the flights. This objective function takes into account the costs to the airports, airlines, and customers. The resulting schedules and routes selected for each flight must respect the landing and takeoff capacity constraints at each airport at each time segment, as well as the capacity constraints over each sector of the National Air Space (NAS). All of these capacity constraints are impacted by the local weather conditions. At each time, we have essentially precise knowledge of the weather at that time period, as well as a weather forecast of the future weather, whose accuracy degrades with the distance from the present time. Therefore we naturally have a multi-stage optimization problem with sequentially revealed uncertainty. While there have been other approaches to build in direct considerations of the evolution of the weather front (e.g., [30]) and also uncertainty in the forecast (e.g., [97], [96]) this is to the best of our knowledge, the first approach that attempts to exploit adaptability.

We build upon an integer linear optimization model proposed in [29]. The authors there dealt with fixed capacity constraints. Here we build up two layers of complexity beyond that. First, the capacity constraints change with time, as the weather front evolves and moves through the country, and second, we treat the exact evolution of the storm front as uncertain.

The resulting problem is a very large scale integer program. We implement a finite adaptability scheme, implemented as a folding horizon problem. While weather uncertainty is not available directly, we argue that the effective uncertainty is low-dimensional, and thus finite adaptability is a well-suited approach for dealing with this problem, both in terms of tractability, and also in terms of its potential effectiveness.

We present two examples that illustrate the benefit of using finite adaptability. We compare the performance of a finite adaptability folding horizon implementation, at several different levels of adaptability, with a pure (i.e., no adaptability) folding horizon implementation with robustness, and also without any robustness.

We find that finite adaptability does considerably better than the pure static robust approach. There is another point illustrated by our computations: the value of building in robustness. The solution produced by the nominal solution, i.e., where no robustness considerations are made, but rather the decision-maker solves the problem placing 100% confidence in the weather forecast, can do quite well if the actual weather trajectory follows closely what was forecast; however, even with small deviations from this, the lack of robustness can lead to very expensive decisions in the later stages, such as additional air holding.

■ 1.2.5 Chapter 6: Future Work and Conclusions

In this chapter, we conclude and present an overview of the dissertation, and the contributions of the work. In addition to this, we provide an extensive discussion of directions for future work. There are several questions that work in this thesis raises, and demand further attention. Chapter 3 leaves open the issue of developing efficient algorithms with provable performance guarantees for special classes of problems. Given the hardness results shown in Chapter 3 we cannot hope for general performance guarantees. However, obtaining improved algorithms for specific classes of problems is an important direction for future work.

Chapter 4 proposes a feasibility maximization problem, and a tractable approximation of this problem. This avenue, along with the simulation data, suggests that one could take a structural risk minimization approach for optimization. We discuss this further in this chapter.

CHAPTER 2

Background

The purpose of this chapter is to give several sections of prerequisite material. The aim is twofold. First, this chapter collects the main facts and results upon which we build our results in the sequel. In addition to collecting here the main background elements for the convenience of the reader, this chapter also serves the purpose of building the proper context for what is to follow.

The majority of the results in this dissertation build upon the concepts of convexity and duality. This theory we do not review, and instead refer the reader to several fine textbooks on Linear Optimization [31], [116], Nonlinear Optimization [18], and Convex Optimization [16], [39], [21].

The starting point for this background chapter is Robust Optimization. In Section 2.2 we primarily review the results of Ben-Tal and Nemirovski, as well as those of Bertsimas and Sim. Robust Optimization takes a deterministic set-based view of uncertainty in optimization. While this is inherently a worst-case view, the deterministic formulation buys us tractability in a wide class of problems. In addition to this, there is a philosophical point about the Robust modeling paradigm: it assumes no knowledge of the underlying distribution. Indeed, it doesn't even assume that the underlying distribution exists (in the sense that there is a fixed distribution generating identically distributed realizations of the uncertainty over time). In many applications, it is not reasonable to assume the existence of an underlying distribution, let alone to hope for any concrete knowledge of the distribution. This, therefore, is an additional motivation for the Robust Optimization point of view, in addition to tractability benefits, as discussed below. In Section 2.3, we review some of the basic results of Stochastic Optimization. In contrast to the Robust Optimization perspective, Stochastic Optimization assumes an underlying stochastic nature to the uncertainty. In Section 2.4, we review some recent sampling approaches to so-called chance constraints, and also some related results from statistical learning theory. Section 2.4.2 considers some sampling approximations to chance constraint problems. These sampling techniques have reliability and feasibility guarantees that one obtains from convexity considerations, as in the work of [43]. It is also possible, as in [51] to obtain such guarantees from uniform learnability results. The latter type of results motivate us to further consider some of the results of Uniform Limit Theorems and Statistical Learning Theory, in Chapter 4.

We review the necessary background here in Section 2.4.3.

■ 2.1 Uncertain Optimization

The starting point for the work in this dissertation is the single stage optimization problem. Since we consider for the most part linear optimization problems, we start with just that:

$$\begin{array}{rl} \min: & \boldsymbol{c}^{\mathsf{T}}\boldsymbol{x} \\ \text{s.t.}: & \boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}. \end{array}$$

In the usual set-up, the parameters defining the optimization, namely $(\boldsymbol{c}, \boldsymbol{A}, \boldsymbol{b})$, are known deterministically. Since the objective function is linear, and the feasible set convex, there will always be an optimal solution at a vertex of the feasible set. Thus optimization naturally pushes the solution to the very boundary of the feasible set, i.e., the boundary of feasibility. By its nature then, the solution is not designed to be *robust* in perturbations in the feasible set. Indeed, as has been observed by Ben-Tal and Nemirovski in [15], and documented by those authors in 90 problems from the Netlib Library ([94]), even small perturbations of the problem can result in "optimal" solutions that are over 100% infeasible with respect to some of the constraints. These "optimal" solutions, then, are essentially meaningless, especially if the constraints in the optimization model are in fact hard constraints that cannot be violated. In the Stochastic Optimization community as well, it has long been observed (see, e.g., [82], [32]) that replacing uncertain parameters with a single deterministic value (possibly their mean value when the notion of mean is available) can lead to very poor solutions.

A reasonable solution, then, is to move away from the boundary of the nominal feasible set towards the interior, trading off optimality for some sense of robustness. The fundamental question is how to optimally choose this trade-off, and where to move in the interior. This depends essentially upon how we model the error, and thus what precisely we mean by performance and robustness to error.

There are essentially two paradigms for modeling uncertainty. In Stochastic Optimization, the uncertainty is modeled as having a stochastic nature. The resulting optimization problem has an objective function, and constraints, that have a probabilistic interpretation. One therefore seeks to optimize in some appropriate probabilistic sense (for example, we may optimize the expected value) subject to again some probabilistic notion of constraint satisfaction.

The other approach is that of Robust Optimization. Here we model the uncertainty in a deterministic set-based way. While in this thesis we consider probabilistic uncertainty (Chapter 4), the foundation is in fact the Robust Optimization perspective. We therefore review this first, and then move on to Stochastic Optimization.

■ 2.2 Robust Optimization

Consider the linear optimization written above, and suppose that the defining parameters (c, A, b) are uncertain. To reflect this, we write them explicitly as a function of a parameter ω representing the uncertainty. First we note that without loss of generality, we can always assume that the vectors c and b are known deterministically, and it is only the matrix A that is subject to uncertainty (this is true because we can easily transform the system by, e.g., adding a variable to convert an objective function into a constraint thus moving uncertainty in c into the matrix A, and similarly for the right hand side vector b). In the Robust Optimization framework, rather than assume that the parameter ω has a probabilistic description and behavior, instead we assume that it can take values on some bounded set Ω that is known *a priori* to the decision-maker. Then feasibility requires that a solution x be feasible to every realization in the uncertainty set Ω. Thus the optimization takes the form:

$$\begin{aligned} \min: \quad & c^{\top} x \\ \text{s.t.}: \quad & A(\omega) x \leq b \quad \forall \omega \in \Omega. \end{aligned} \tag{2.2.1}$$

This deterministic view of the uncertainty essentially amounts to a worst-case viewpoint. Such worst-case perspectives have been used widely in other fields. In the computer science community, the worst-case approach has seen many applications (see, e.g., [75]). In Control Theory as well, in the theory of Robust Control[1], worst-case formulations have been widely considered (e.g., [58], [144], [7], and references therein). In Optimization, however, worst-case formulations have a more recent history. The Robust Optimization formulation was first proposed by Soyster in 1973 ([125]). However, the attention Robust Optimization has attracted in the last decade, is really due to the important work of Ben-Tal and Nemirovski ([15], [14], [17]) on Robust Linear Optimization, Robust Quadratic Programming, and Robust Conic Programming, and also by work of El Ghaoui et al. ([70], [71]) on Robust Least Squares and Robust SDP, and then more recently by work of Bertsimas and Sim ([25], [26], [27], [28]).

Remark 2.1

There has also been some recent work on Robust Optimization models for non-convex problems, such as those whose feasible set is described by a solution to a partial differential equation. See, for instance, [142],[143], for work on Robust Optimization applied to a general Nonlinear Optimization context. This dissertation is more concerned with sets with convex constraints, or convex constraints intersected with a discrete set like the integers, or the corners of the hypercube. Therefore we do not consider work in this vein, although it certainly deserves consideration.

[1] We note that the word *Robust* in Control does not necessarily imply a deterministic worst-case formulation, where as in the Optimization literature, Robust Optimization has become synonymous with the worst-case perspective

Before we discuss the specific contributions of these results in more detail, we give the following simple observation.

Lemma 2.1 *Let the rows of the matrix $A(\omega)$ be denoted by $a_i(\omega)$. If A has m rows, make m copies of Ω, so that $\Omega^{(i)} = \Omega$. Then the optimization problem (2.2.1) is equivalent to the following formulation:*

$$\begin{array}{rll} \min: & c^{\top}x & \\ \text{s.t.}: & a_1(\omega^{(1)})^{\top}x \le b_1 & \forall \omega^{(1)} \in \Omega^{(1)}, \\ & \vdots & \\ & a_m(\omega^{(m)})^{\top}x \le b_m & \forall \omega^{(m)} \in \Omega^{(m)}. \end{array} \tag{2.2.2}$$

Remark 2.2
The i^{th} constraint may only use some subset of the components of ω. Let Ω_i be the projection of Ω onto those components. Then the effective uncertainty set can be regarded as the rectangular set $\tilde{\Omega} \overset{\Delta}{=} \Omega_1 \times \cdots \times \Omega_m$. This idea is further developed in Chapter 3.

The Lemma gives a straightforward result, but it is nonetheless an observation that is important for what is to follow. It says that the Robust Optimization formulation cannot capture uncertainty that is not constraint-wise.

PROOF. Certainly any solution feasible for (2.2.2) is feasible for (2.2.1). Let x_R denote the optimal solution to problem (2.2.1). If x_R is not feasible for problem (2.2.2), there must be some index i, and $\omega^{(i)} \in \Omega^{(i)}$ such that $a_i(\omega^{(i)})^{\top}x_R > b_i$. But this is impossible since then $A(\omega^{(i)}x_R \le b$ is not satisfied. □

Now let us consider the solution of the Robust Optimization formulation, and thus the tractability of the solution as well. It is convenient to consider the equivalent constraint-wise formulation of the above lemma. Consider the i^{th} uncertain constraint (we drop the superscripts on ω and Ω since they are not necessary):

$$a_i(\omega)^{\top}x \le b_i, \quad \forall \omega \in \Omega. \tag{2.2.3}$$

Equivalently, we can replace this constraint by an equivalent maximization formulation:

$$\left[\begin{array}{rl} \max: & a_i(\omega)^{\top}x \\ \text{s.t.}: & \omega \in \Omega \end{array}\right] \le b_i. \tag{2.2.4}$$

Note that so far the discussion is completely general. If we have a general constraint, $f(x, \omega) \le 0$, again the robust version of this can be written as the maximization problem

$$\left[\begin{array}{rl} \max: & f(x, \omega) \\ \text{s.t.}: & \omega \in \Omega \end{array}\right] \le 0. \tag{2.2.5}$$

This inner maximization is what we refer to as the **inner problem** of the Robust Optimization formulation. The tractable solvability of any robust optimization formulation, depends critically on the structure of this inner problem. Indeed, if for any fixed ω, the optimization problem is convex, i.e., the sub-level sets $\{x : f(x, \omega) \leq \alpha\}$ are convex, then the tractability of the robust optimization is essentially determined by the subproblem.[2] If we replace the maximization problem by its dual, by weak duality, any feasible to the dual problem (which is a minimization) that satisfies the inequality of the constraint, is an upper bound on the value of the maximization, and hence a certificate that the point x is feasible. If the dual is a tractable convex problem, then the overall problem we need to solve becomes convex, and hence tractable. The value of this optimization problem provides a bound on the optimal value of the original problem. If in addition, the inner problem can be expressed as a convex optimization problem satisfying an appropriate constraint qualification so as to have no duality gap (see, e.g., [21], [112]), then strong duality holds, there is therefore duality gap between the maximization and its dual minimization problem, and hence we have an exact reformulation of the original robust problem as a convex optimization.

This general use of duality as described above, is the driving force behind the results of Ben-Tal and Nemirovski, Bertsimas and Sim, and El Ghaoui et al. The structure of the uncertainty set, and also the function $f(x, \omega)$, determines the structure of the dual of the subproblem, and hence the structure of the explicit reformulation of the robust problem. We briefly mention some of these structural results.

■ 2.2.1 Polyhedral Uncertainty Set

The use of polyhedral uncertainty sets was used with great success in Bertsimas and Sim, as in the case where the uncertainty affects the constraints in an affine manner, the dual to the subproblem is again a linear program. In this case, the explicit reformulation of a robust linear optimization, again becomes a linear optimization. To illustrate this, let us consider uncertainty sets directly in terms of the uncertain matrix A (rather than maintaining explicitly the uncertain parameter Ω):

$$\Omega_i = \{a_i : D_i a_i \leq d_i\}.$$

Now consider the problem:

$$\begin{aligned} \min : \quad & c^\top x \\ \text{s.t.} : \quad & \max\nolimits_{\{a_i \in \Omega_i\}} a_i^\top x \leq b_i, \quad i = 1, \ldots, m. \end{aligned}$$

[2] If the sublevel sets are not convex, then the nominal problem (without uncertainty) is not convex, and thus may itself be intractable.

The dual of the subproblem can be written as:

$$\left[\begin{array}{ll} \max: & \boldsymbol{a}_i^\top \boldsymbol{x} \\ \text{s.t.}: & \boldsymbol{D}_i \boldsymbol{a}_i \le \boldsymbol{d}_i \end{array}\right] \longleftrightarrow \left[\begin{array}{ll} \min: & \boldsymbol{p}_i^\top \boldsymbol{d}_i \\ \text{s.t.}: & \boldsymbol{p}_i^\top \boldsymbol{D}_i = \boldsymbol{x} \\ & \boldsymbol{p}_i \ge 0. \end{array}\right]$$

and therefore the robust linear optimization now becomes:

$$\begin{array}{ll} \min: & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{s.t.}: & \boldsymbol{p}_i^\top \boldsymbol{d}_i \le b_i, \quad i = 1, \ldots, m \\ & \boldsymbol{p}_i^\top \boldsymbol{D}_i = \boldsymbol{x}, \quad i = 1, \ldots, m \\ & \boldsymbol{p}_i \ge 0, \quad i = 1, \ldots, m. \end{array}$$

Bertsimas and Sim ([26]) use this duality with a family of polyhedral sets that encode a *budget of uncertainty* in terms of cardinality constraints. That is, the uncertainty sets they consider control the number of parameters of the problem that are allowed to vary from their nominal values. This budget of uncertainty helps control the trade-off between the optimality of the solution, and its robustness to parameter perturbation. In [24], the authors show that these cardinality constrained uncertainty sets can be expressed as norm-bounded uncertainty sets.

The cardinality constrained uncertainty sets are as follows. Given an uncertain matrix, $\boldsymbol{A} = (a_{ij})$, suppose that for row i, the entries a_{ij} for $j \in J_i \subseteq \{1, \ldots, n\}$ are subject to uncertainty. Furthermore, each component a_{ij} is assumed to vary in some interval about its nominal value, $[a_{ij} - \hat{a}_{ij}, a_{ij} + \hat{a}_{ij}]$. Rather than protect against the case when every parameter can deviate, as in the original model of Soyster ([125]), we allow at most Γ_i coefficients to deviate. Thus in this sense, the positive number Γ_i denotes the budget of uncertainty for the i^{th} constraint. [3] Given values $\Gamma_1, \ldots, \Gamma_m$, the robust formulation becomes:

$$\begin{array}{lll} \min: & \boldsymbol{c}^\top \boldsymbol{x} & \\ \text{s.t.}: & \sum_j a_{ij} x_j + \max_{\{S_i \subseteq J_i \,:\, |S_i| = \Gamma_i\}} \sum_{j \in S_i} \hat{a}_{ij} y_j \le b_i & 1 \le i \le m \\ & -y_j \le x_j \le y_j & 1 \le j \le n \\ & \boldsymbol{l} \le \boldsymbol{x} \le \boldsymbol{u} & \\ & \boldsymbol{y} \ge \boldsymbol{0}. & \end{array} \tag{2.2.6}$$

Then taking the dual of the inner maximization problem, one can show that the above is equivalent to the following linear formulation, and therefore is tractable (and more-

[3] For the full details see [26].

over is a linear optimization):

$$\begin{array}{lll} \max: & \boldsymbol{c}^{\mathsf{T}}\boldsymbol{x} & \\ \text{s.t.}: & \sum_j a_{ij}x_j + z_i\Gamma_i + \sum_j p_{ij} \leq b_i & \forall i \\ & z_i + p_{ij} \geq \hat{a}_{ij}y_j & \forall i,j \\ & -y_j \leq x_j \leq y_j & \forall j \\ & \boldsymbol{l} \leq \boldsymbol{x} \leq \boldsymbol{u} & \\ & \boldsymbol{p} \geq \mathbf{0} & \\ & \boldsymbol{y} \geq \mathbf{0}. & \end{array}$$

2.2.2 Ellipsoidal Uncertainty Set

Ben-Tal and Nemirovski, as well as El Ghaoui et al., consider ellipsoidal uncertainty sets. One of the motivations for this is that the truncated normal distribution yields ellipsoidal uncertainty sets, and therefore this gives a concrete physical intuition for the meaning of the uncertainty sets. Here, the budget of uncertainty takes the form of the size of the ellipsoidal sets, rather than the number of parameters allowed to vary.

In addition, now the subproblem is no longer a maximization over a polytope, but rather over a quadratically defined set. Therefore we resort to quadratic optimization duality rather than linear optimization duality. As a consequence, the resulting dual problem is not linear. If the original problem is linear, the robust equivalent is a second order cone (SOCP). Second order cone problems become Semidefinite programs (SDP), and in general, robust SDPs do not have a reformulation as a tractable convex optimization problem, when the uncertainty set is an intersection of ellipsoids.

We illustrate here only how to obtain the explicit reformulation of a robust quadratic constraint, subject to simple ellipsoidal uncertainty[4] Here we follow Ben-Tal, Nemirovski and Roos ([17]). Consider the quadratic constraint

$$\boldsymbol{x}^{\mathsf{T}}\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\boldsymbol{x} \leq 2\boldsymbol{b}^{\mathsf{T}}\boldsymbol{x} + \boldsymbol{c}, \quad \forall(\boldsymbol{A},\boldsymbol{b},\boldsymbol{c}) \in \Omega, \tag{2.2.7}$$

where the uncertainty set Ω is an ellipsoid about a nominal point $(\boldsymbol{A}^0, \boldsymbol{b}^0, \boldsymbol{c}^0)$:

$$\Omega \triangleq \left\{(\boldsymbol{A},\boldsymbol{b},\boldsymbol{c}) := (\boldsymbol{A}^0,\boldsymbol{b}^0,\boldsymbol{c}^0) + \sum_{l=1}^{L} \omega_l(\boldsymbol{A}^l,\boldsymbol{b}^l,\boldsymbol{c}^l) \ : \ ||\omega||_2 \leq 1\right\}.$$

As in the previous section, a vector $\boldsymbol{x}$ is feasible for the robust constraint (2.2.7) if and only if it is feasible for the constraint:

$$\begin{bmatrix} \max: & \boldsymbol{x}^{\mathsf{T}}\boldsymbol{A}^{\mathsf{T}}\boldsymbol{A}\boldsymbol{x} - 2\boldsymbol{b}^{\mathsf{T}}\boldsymbol{x} - \boldsymbol{c} \\ \text{s.t.}: & (\boldsymbol{A},\boldsymbol{b},\boldsymbol{c}) \in \Omega \end{bmatrix} \leq 0.$$

[4] Here, *simple ellipsoidal uncertainty* means the uncertainty set is a single ellipsoid, as opposed to an intersection of several ellipsoids.

This is the maximization of a convex quadratic objective (when the variable is the matrix $\boldsymbol{A}$, $\boldsymbol{x}^\top \boldsymbol{A}^\top \boldsymbol{A}\boldsymbol{x}$ is quadratic and convex in $\boldsymbol{A}$ since $\boldsymbol{x}\boldsymbol{x}^\top$ is always semidefinite) subject to a single quadratic constraint. It is well-known that while this problem is not convex (we are maximizing a convex quadratic) it nonetheless enjoys a hidden convexity property (for an early reference, see [40]) that allows it to be reformulated as a (convex) semidefinite optimization problem. Related to this and also well-known, is the so-called S-lemma (or S-procedure) in control [38]:

Lemma 2.2 (S-lemma) *Let F and G be quadratic in $\boldsymbol{x} \in \mathbb{R}^n$:*

$$\begin{aligned} \boldsymbol{F}(\boldsymbol{x}) &= \boldsymbol{x}^\top \boldsymbol{P}\boldsymbol{x} + 2\boldsymbol{p}_1^\top \boldsymbol{x} + p_0, \\ \boldsymbol{G}(\boldsymbol{x}) &= \boldsymbol{x}^\top \boldsymbol{Q}\boldsymbol{x} + 2\boldsymbol{q}_1^\top \boldsymbol{x} + q_0, \end{aligned}$$

where $\boldsymbol{P}, \boldsymbol{Q}$ are symmetric matrices. Suppose further that there exists some $\boldsymbol{x}_0$ such that $G(\boldsymbol{x}_0) > 0$. Then

$$F(\boldsymbol{x}) \geq 0 \quad \forall \boldsymbol{x} \in \{\boldsymbol{x} : G(\boldsymbol{x}) \geq 0\},$$

if and only if there exists a scalar $\tau \geq 0$ such that

$$G(\boldsymbol{x}) - \tau F(\boldsymbol{x}) \geq 0, \quad \forall \boldsymbol{x} \in \mathbb{R}^n.$$

Note that the condition that there exist some $\boldsymbol{x}_0$ such that $G(\boldsymbol{x}_0) > 0$, is exactly a Slater-type condition, and this guarantees that strong duality holds.

There are numerous restatements of this result in the literature. An equivalent restatement of this statement is that the first sum of squares relaxation of the quadratic optimization, is exact (see [102]). Indeed, the global nonnegativity in the last statement above, is equivalent to a positive semidefiniteness of the matrix defining the quadratic function $G(\boldsymbol{x}) - \tau F(\boldsymbol{x})$. This lemma is important in our context, because it essentially gives the boundary between what we can solve exactly, and where solving the subproblem becomes difficult. Indeed, if the uncertainty set is an intersection of ellipsoids, then exact solution of the subproblem is NP-hard.[5] In Section 2.2.6 we review an extension of the robust framework to multistage optimization. We see there that the solution of the subproblem is precisely the tractability bottleneck, and the S-lemma essentially marks the landscape of what can be solved exactly.

As an immediate corollary of the S-lemma, we then obtain a solution to our original problem, of feasibility for the robustified quadratic constraint. It amounts to the feasibility of an SDP. Therefore subject to mild regularity conditions (e.g., Slater's condition) strong duality holds, and therefore by using the dual to the SDP, we have a convex exact reformulation of the subproblem in the Robust Optimization.

[5] Nevertheless, there are some approximation results available: [17].

Corollary 2.3

Given a vector $\boldsymbol{x}$, it is feasible to the robust constraint (2.2.7) if and only if there exists a scalar $\tau \in \mathbb{R}$ such that the following matrix inequality holds:

$$\left(\begin{array}{c|ccc|c} c^0 + 2\boldsymbol{x}^\top \boldsymbol{b}^0 - \tau & \frac{1}{2}c^1 + \boldsymbol{x}^\top \boldsymbol{b}^1 & \cdots & c^L + \boldsymbol{x}^\top \boldsymbol{b}^L & (\boldsymbol{A}^0\boldsymbol{x})^\top \\ \hline \frac{1}{2}c^1 + \boldsymbol{x}^\top \boldsymbol{b}^1 & \tau & & & (\boldsymbol{A}^1\boldsymbol{x})^\top \\ \vdots & & \ddots & & \vdots \\ \frac{1}{2}c^L + \boldsymbol{x}^\top \boldsymbol{b}^L & & & \tau & (\boldsymbol{A}^L\boldsymbol{x})^\top \\ \hline \boldsymbol{A}^0\boldsymbol{x} & \boldsymbol{A}^1\boldsymbol{x} & \cdots & \boldsymbol{A}^L\boldsymbol{x} & I \end{array}\right) \succeq \mathbf{0}.$$

Thus in this section, as well as in the previous section, we have seen that the tractability of the subproblem in Robust Optimization, depends on two factors: the nature of the dependence of the parameters of the problem on the uncertainty (affine and quadratic dependence, respectively, in the last two sections) and the structure of the uncertainty set itself.

The next section briefly considers some of the extensions to more general classes of convex problems.

■ 2.2.3 Extensions to General Conic Robust Optimization

The Robust Counterpart to a general conic convex optimization problem is typically nonconvex and intractable ([14]). This is implied by the results described above, since conic problems include semidefinite optimization. Nevertheless, there are some approximate formulations of the general conic convex robust problem. We refer the interested reader to the recent paper, [28].

■ 2.2.4 Extensions to Discrete Robust Problems

There has also been some work extending the Robust Optimization framework to the discrete setting. Primarily, we refer to the work of Bertsimas and Sim, in [25]. Philosophically, this is of interest to us because a central motivation to the work in this thesis, including our main application to Air Traffic Control in Chapter 5, is to build a framework that can accommodate discrete variables.

■ 2.2.5 Probability Guarantees

Thus far in our discussion of the development and result of Robust Optimization, we have said nothing about distributions. Indeed, the formulation itself is free from probability theory. In a sense, this may account for the tractability advantages of Robust Optimization over Stochastic models, at least in the case of the solvable setups discussed above.

Yet, as Ben-Tal and Nemirovski, and also Bertsimas and Sim show, given a particular robustification scheme, it is possible to obtain probability bounds on the feasibility of the solution, for different levels of noise, and under different distributional assumptions. It is important to stress that the actual solution is designed to be deterministically feasible to some level (and description) of uncertainty. One can then prove that in fact this solution also has a certain level of probabilistic protection to uncertainty, when the uncertainty has a stochastic description. Therefore in this sense, probability is used only after the fact, to provide a probabilistic analysis of the solution. The construction is entirely deterministic.

Consider the formulation of the robust linear problem, with polyhedral uncertainty set given by the cardinality constraint and the budget of uncertainty for each row, i, $\{\Gamma_i\}$, as in the exposition above in Section 2.2.1. In [26], the authors consider a model of uncertainty where each entry a_{ij} of the uncertain matrix that is can vary from its nominal value (and hence $j \in J_i$, as in (2.2.6)) is modeled as a random variable taking values in a symmetric interval about its nominal value: $[a_{ij} - \hat{a}_{ij}, a_{ij} + \hat{a}_{ij}]$. Assume that the random variable $\eta_{ij} \triangleq (\hat{a}_{ij} - a_{ij})/\hat{a}_{ij}$ follows a symmetric distribution over $[-1, 1$. This distribution is assumed to be symmetric, but otherwise unknown. Then, Bertsimas and Sim prove:

Proposition 2.4
For $\boldsymbol{x}^*$ the optimal solution of (2.2.6), and $S_i^* \subset J_i$ the worst-case set of varying parameters in the robust optimization, then for any symmetric distribution of the matrix entries around their nominal value:

1. *The probability that the i^{th} constraint is violated is bounded as follows:*

$$\mathbb{P}\left(\sum_j \hat{a}_{ij} x_j^* > b_i\right) \leq \mathbb{P}\left(\sum_{j \in J_i} \gamma_{ij}\eta_{ij} \geq \Gamma_i\right)$$

where

$$\gamma_{ij} = \begin{cases} 1, & \text{if } j \in S_i^*, \\ \frac{\hat{a}_{ij}|x_j^*|}{\hat{a}_{ir^*}|x_{r^*}^*|}, & \text{if } j \in J_i \setminus S_i^* \end{cases}$$

and

$$r^* = \arg\min_{r \in S_i^*} \hat{a}_{ir}|x_r^*|.$$

2. *We can further bound the probability bound above:*

$$\mathbb{P}\left(\sum_{j \in J_i} \gamma_{ij}\eta_{ij} \geq \Gamma_i\right) \leq \exp\left(-\frac{\Gamma_i^2}{2|J_i|}\right).$$

There are a number of results along these lines including [26],[28],[16], [15] and references therein. There are also results that have focused also on obtaining improved *a posteriori* probability bounds when more information about the underlying distribution is available, such as the work in [46] and [47].

■ 2.2.6 A Multistage Extension: Affine Adaptability

Thus far, the discussion on Robust Optimization has been limited to the single-stage case. Indeed, multi-stage extensions for the Robust Optimization framework have only recently started attracting the attention of the research community, and thus far not much work has been done.

We briefly review one of the pioneering papers in this area, again due to Ben-Tal, Nemirovski, along with two co-authors, Goryashko and Guslitzer: [12]. Here, the authors consider multi-stage linear optimization, where the parameters of the problem are affinely impacted by uncertainty, that takes values in a general ellipsoidal set.[6]

We consider an uncertain two-stage optimization problem, where the uncertainty is treated within the Robust Optimization framework, and the second stage variables are allowed to depend on the uncertainty, while the first-stage variables have no adaptability to the uncertainty. The interpretation is the usual one: the uncertainty is realized after the first-stage decisions must be implemented, but before the implementation of the second-stage decisions. Let $\boldsymbol{x}$ denote the first-stage decision, $\boldsymbol{y}$ the second-stage decisions, and $\boldsymbol{\omega}$ the uncertainty vector. We note that we can without loss of generality (e.g., by adding a variable if necessary) assume that the objective function involves only the first-stage variables. Similarly, we assume that the objective function and the right hand side vector are known deterministically. Then we have:

$$\begin{aligned} \text{min} :\ & \boldsymbol{c}^{\mathsf{T}}\boldsymbol{x} \\ \text{s.t.} :\ & \boldsymbol{A}(\boldsymbol{\omega})\boldsymbol{x} + \boldsymbol{B}(\boldsymbol{\omega})\boldsymbol{y}(\boldsymbol{\omega}) \leq \boldsymbol{b}, \quad \forall \boldsymbol{\omega} \in \Omega. \end{aligned}$$

For general adaptability, i.e, for arbitrary dependence of the second-stage decisions $\boldsymbol{y}$ on $\boldsymbol{\omega}$, this is equivalent to:

$$\begin{aligned} \text{min} :\ & \boldsymbol{c}^{\mathsf{T}}\boldsymbol{x} \\ \text{s.t.} :\ & \boldsymbol{x} \in \mathcal{K}, \end{aligned} \tag{2.2.8}$$

where the feasible set $\mathcal{K}$ is given by

$$\mathcal{K} \triangleq \{\boldsymbol{x} \,|\, \forall \boldsymbol{\omega} \in \Omega \ \exists \boldsymbol{y} \text{ s.t. } \boldsymbol{A}(\boldsymbol{\omega})\boldsymbol{x} + \boldsymbol{B}(\boldsymbol{\omega})\boldsymbol{y} \leq \boldsymbol{b}\}.$$

The set $\mathcal{K}$ is convex, however the quantifier in the definition renders the resulting optimization problem (2.2.8) NP-hard (in general). Indeed, recall the simple example

[6] By *general ellipsoidal set* we mean a set that is the intersection of potentially many ellipsoids, as opposed to a *simple* ellipsoidal set.

(1.2.2) from Chapter 1:

$$\begin{array}{rl} \min: & x \\ \text{s.t.}: & x \geq \boldsymbol{y}^\top \boldsymbol{\omega} \\ & \boldsymbol{y} = \boldsymbol{Q}\boldsymbol{\omega}. \end{array}$$

In this case, the set $\mathcal{K}$ is convex, and one-dimensional:

$$\begin{array}{rcl} \mathcal{K} & = & \{x : x \geq \boldsymbol{\omega}^\top \boldsymbol{Q}\boldsymbol{\omega}, \quad \forall \boldsymbol{\omega} \in \Omega\} \\ & = & [x_0, \infty), \end{array}$$

but computing the lower bound, x_0, of the interval is NP-hard.

Rather than allow arbitrary adaptability, the authors in [12] restrict the functional form of the adaptability, to be affine in the uncertain parameters, i.e., in $\boldsymbol{\omega}$. Of course, from the example above, again this cannot buy tractability. However, for a large class of problems, they are able to use a generalization of the S-lemma proved in [17], to obtain approximate solutions. Consider, then, an affine scheme for adaptability:

$$\boldsymbol{y}(\boldsymbol{\omega}) = \boldsymbol{Q}\boldsymbol{\omega} + \boldsymbol{q},$$

where now the decision variables for the second stage are given by the matrix $\boldsymbol{Q}$ and the vector $\boldsymbol{q}$. The two-stage optimization problem, then, becomes:

$$\begin{array}{rl} \min: & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{s.t.}: & \boldsymbol{A}(\boldsymbol{\omega})\boldsymbol{x} + \boldsymbol{B}(\boldsymbol{\omega})[\boldsymbol{Q}\boldsymbol{\omega} + \boldsymbol{q}] \leq \boldsymbol{b}, \quad \forall \boldsymbol{\omega} \in \Omega. \end{array}$$

But this is a single-stage problem, that is linear in the decision-variables, and quadratic in the coefficients of the uncertainty.[7] The subproblem, therefore, is:

$$\begin{array}{rl} \max: & \boldsymbol{a}_i(\boldsymbol{\omega})\boldsymbol{x} + \boldsymbol{b}_i(\boldsymbol{\omega})[\boldsymbol{Q}\boldsymbol{\omega} + \boldsymbol{q}] \leq b_i \\ \text{s.t.}: & \boldsymbol{\omega} \in \Omega. \end{array}$$

This is the maximization of a possibly indefinite quadratic function (in ω) subject to a convex constraint Ω. If Ω is defined by a single ellipse, then the S-lemma applies, and the subproblem is tractable, as there is no duality gap because of the property of hidden convexity (and Slater's condition). In the case where Ω is a general ellipsoidal set, Ben-Tal, Nemirovski and Roos show ([17]) that while there is in general a duality gap, they can bound its size.

There are two main observations that are particularly motivating to us. First, we see that the subproblem seems to be the tractability bottleneck, and within the robust framework there does not seem to be much of a way around this, as long as $P \neq$

[7] The quadratic term is $B(\omega)Q\omega$, when $\boldsymbol{B}(\cdot)$ is affine in ω. If the dependence of $\boldsymbol{B}(\cdot)$ on ω is not affine, then the subproblem is further complicated.

NP. Second, the complexity of solving the subproblem is further complicated for other classes of adaptability (not to mention broader classes of problems, such as those with nonlinear dependence on the uncertainty), and as a result, this approach does not permit the construction of a hierarchy of adaptability. We provide an example in Chapter 3 where, issues of complexity aside, the affine adaptable formulation is no better than the original static robust formulation. A central motivation in this thesis is to build a hierarchy of adaptability, so that we can always have recourse to a higher level of adaptability if the computational resources are available.

■ 2.3 Stochastic Optimization

In this section, we switch gears to talk about the stochastic paradigm for modeling uncertainty. Here, we assume that the uncertain parameter behaves according to a probability distribution.

Stochastic Optimization has a long history, dating at least as far back as Dantzig's original paper [48]. Since then, much work has been done, in various aspects of Stochastic Optimization. We refer the reader to several textbooks ([79] [35], [108], [81]) and the many references therein for a more comprehensive picture of Stochastic Optimization.

In Robust Optimization, once the uncertainty set is specified, the constraints are treated as hard constraints, and thus this approach does not leave room for further modeling decisions: constraints must be satisfied for every realization of the uncertainty. This is not the case for the stochastic framework. Here, important modeling decisions include specifying the sense in which a solution is optimal, or feasible. Indeed, consider a linear problem of the form:

$$\begin{aligned} \min: &\quad \boldsymbol{c}(\boldsymbol{\omega})^{\top}\boldsymbol{x} \\ \text{s.t.}: &\quad \boldsymbol{A}(\boldsymbol{\omega})\boldsymbol{x} \leq \boldsymbol{b}(\boldsymbol{\omega}). \end{aligned}$$

Since the objective function, $\boldsymbol{c}(\boldsymbol{\omega})$ is itself a random variable, two solutions $\boldsymbol{x}_1, \boldsymbol{x}_2$ may not be comparable, in the sense that one may not dominate the other with probability 1. A typical approach in Stochastic Optimization is to try to optimize in expectation, often adding guarantees for downside risk, say, by attempting to place certain bounds on the performance at tails of the distribution.

Let us consider the feasibility constraints, $\boldsymbol{A}(\boldsymbol{\omega})\boldsymbol{x} \leq \boldsymbol{b}(\boldsymbol{\omega})$. If we interpret the constraints as "hard" constraints, in the optimization sense, so that we assign value $+\infty$ to the optimization problem if the constraints are violated, then any expectation operation (or anything like it) necessarily reduces to demanding that the constraint be satisfied with probability 1. But this is essentially a Robust Optimization constraint, as the feasible set becomes the set of vectors $\boldsymbol{x}$ that satisfy the constraints deterministically for every realization $\boldsymbol{\omega} \in \Omega = \text{supp}(\boldsymbol{\omega})$, the support of the distribution.

There are several ways to relax the constraint:

$$\mathbb{P}(\boldsymbol{A}(\omega)\boldsymbol{x} \leq \boldsymbol{b}(\omega)) = 1.$$

The first involves a notion known as *recourse*, and converting the single-stage Stochastic Optimization into a two-stage problem. The second approach that is often used, involves directly relaxing the probability of feasibility constraint, so that we enforce feasibility with some probability $(1-\varepsilon)$ (for $\varepsilon > 0$), and then we place no restriction or penalty on what happens outside a set of measure $(1-\varepsilon)$. The constraint

$$\mathbb{P}(\boldsymbol{A}(\omega)\boldsymbol{x} \leq \boldsymbol{b}(\omega)) \geq 1-\varepsilon,$$

is called a *chance constraint*.

We discuss recourse first, and then we consider chance constraints.

2.3.1 Recourse in Stochastic Optimization

One way to avoid facing a hard constraint for every possible realization of the uncertainty, is to interpret the constraint as a soft constraint, penalize its violation, and subsequently seek to minimize the expected penalty. The motivation for this approach comes from the observation that in many situations which we model with hard constraints, the constraints can be satisfied "at extra cost" if original planning proves inadequate. For instance, if a supplier fails to provision adequately to meet demand, it may well be possible to buy extra provisions "at the last minute" in order to meet demand. Of course, each unit purchased at the last minute, may have a price well above the original production or supply cost.

Thus we transform a single stage problem into a two-stage problem. The second-stage variables, then, correspond to the actions that must be implemented to mitigate the unexpected realization of the uncertainty, and in order to make the first-stage decisions feasible to the original problem. For example, a generic linear problem of the form given above:

$$\begin{array}{rl} \min: & \boldsymbol{c}(\omega)^\top \boldsymbol{x} \\ \text{s.t.}: & \boldsymbol{A}(\omega)\boldsymbol{x} \geq \boldsymbol{b}(\omega) \end{array}$$

would become:

$$\begin{array}{rl} \min: & \boldsymbol{c}(\omega)^\top \boldsymbol{x} + \boldsymbol{d}^\top \boldsymbol{y}(\omega) \\ \text{s.t.}: & \boldsymbol{A}(\omega)\boldsymbol{x} + \boldsymbol{y}(\omega) \geq \boldsymbol{b}(\omega), \end{array}$$

where $\boldsymbol{y}$ represents the second-stage decisions, and $\boldsymbol{d}$ the cost of the second stage variables. There are two aspects of this problem that are worth pointing out. First, note that the matrix multiplying the second-stage function $\boldsymbol{y}$ is constant. This is known as *constant recourse*. Second, note that as written, any first-stage decision $\boldsymbol{x}$ can be made feasible under any possible realization of the uncertainty, by proper choice of the second-stage variable $\boldsymbol{y}$. This is known as *complete recourse*, and it is a consequence of the matrix multiplying $\boldsymbol{y}$ having full column rank. This is clearly the case, in this case, since the matrix in question is just the identity matrix. We note that in our previous formulations of two-stage problems, we made no mention of constant, or complete recourse. This point is important in the sequel. In Chapter 4, the probability of feasibility of the first-stage decision is particularly important, and we consider the impact of adaptability on the first-stage feasibility. With complete recourse, this quantity is always equal to one.

Once we add the second stage variable, and thus remove issues of feasibility, we are left with the minimization of the random quantity:

$$\boldsymbol{c}(\omega)^{\top}\boldsymbol{x} + \boldsymbol{d}^{\top}(\boldsymbol{b}(\omega) - \boldsymbol{A}(\omega)\boldsymbol{x}).$$

Typically, we then seek to minimize the expectation of this quantity, and the problem becomes:

$$\min : \mathbb{E}[\boldsymbol{c}(\omega)^{\top}\boldsymbol{x} + \boldsymbol{d}^{\top}(\boldsymbol{b}(\omega) - \boldsymbol{A}(\omega)\boldsymbol{x})].$$

Computing expectations of continuous functions in high dimensions can be particularly challenging. There are several approaches to computing expectations. One is to do it by discretizing the expectation, approximating it by considering samples of the distribution. But the complexity of this may be quite large (see, e.g., [132], [98]).

Because of this, much work has been done in obtaining approximations for the expectation of convex functions, since early in the history of Stochastic Optimization. For example, [59], [86] obtain upper bounds on expectation of convex functions, and these are applied to stochastic optimization in [87]. More recent work, including bounds based on higher order moments of the distribution, and bounds approximating the distribution, and bounds approximating the function to be integrated, include [84], [36], [34], [91], and [54]. See the textbook [35] for more on such bounds, and their application to Stochastic Optimization.

Another approach to approximating expectations is via Monte Carlo sampling methods. The idea is that if we have access to samples from the distribution, we can generate a finite approximation to the underlying distribution, with which we can approximate the expectations directly. Sampling methods are of central importance to this thesis, particularly in Chapter 4. We discuss them further below in the context of approximating Chance Constraints. The complexity of integration via sampling can be prohibitively bad for two reasons: first, the number of samples required crucially depends on the nature of the integrand, since an interesting result in sampling are those uniform over some set of possible integrands. A uniform result over a set of integrands whose regularity is not controlled, is of course impossible.[8]

[8] Consider, for instance, the hopeless task of obtaining a uniform bound on the number of samples required to approximate the integral of a piecewise constant function with an arbitrary (but finite) number

■ 2.3.2 Chance Constraints

Chance constraints, also called probabilistic constraints, present a rather natural relaxation for Stochastic Optimization. Moreover, it has long been known that a solution that is feasible with probability $(1-\varepsilon)$, can have performance dramatically better than a solution that is feasible with probability 1. Indeed, in particular for situations where the uncertain parameter has unbounded support, say, whenever we model the noise as Gaussian, then demanding that a solution be feasible with probability 1 is overly conservative and restrictive, and typically would result in an infeasible formulation.

Moreover, this approach is well-motivated in many applications, where we want to guarantee performance or reliability of a system with high probability. Chance constraints are of course also motivated by work in statistics, stochastic processes, large deviations, and reliability theory. The earliest work on chance constraints dates back to work of Charnes et al. in [45] who considered such constraints in the context of the management of heating oil production, and [44], and also in [90]. One of the more significant contributions to this area of Stochastic Programming has been made by Prékopa, who initiated the study of the general model of Stochastic Optimization using Chance Constraints (see [104],[105],[106] for the early work, [107] for a survey with many useful references, and [108] for a textbook on Stochastic Optimization).

The challenge in dealing with Chance Constraints comes primarily from the fact that the sets of ε-feasibility of some constraint $\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}$, defined as

$$\mathcal{X}_\varepsilon \stackrel{\Delta}{=} \{\boldsymbol{x} \,:\, \mathbb{P}(\boldsymbol{A}\boldsymbol{x} \leq \boldsymbol{b}) \geq 1-\varepsilon\},$$

are generally non-convex. In some very special (and limited cases) Chance Constraints are convex, i.e., the sets $\mathcal{X}_\varepsilon$ are convex. In general, convexity requires log concavity properties of the probability distribution. For constraints of the form given above, more is required. As outlined in [107], if the distribution of ω is such that the induced distribution on $\boldsymbol{A}$ and $\boldsymbol{b}$ is jointly normal with some additional restrictions on the covariance matrix ([107]), then $\mathcal{X}_\varepsilon$ is convex, for $\varepsilon > 1/2$. These conditions are not typically satisfied, however, and thus optimizing over the sets becomes a challenging task.

There have been several approaches that aim at obtaining convex approximations to the non-convex sets $\mathcal{X}_\varepsilon$. The papers on Robust Optimization reviewed in this chapter, give some convex approximations to the feasibility sets, as they show that deterministic feasibility to a given uncertainty set can be shown to imply feasibility with high probability, when the uncertainty is stochastically generated. A more direct approach appears in a paper of Nemirovski and Shapiro [93]. There, the authors construct an approximation to the chance constraints using ideas from Large Deviations and Bernstein approximation (see [53] for some of the background ideas involved in

of pieces.

the work in this paper). The authors require some additional structural assumptions, namely, the uncertainty must affect the parameters of the problem in an affine manner. We see in Chapter 4 that this limits the application of such results, in particular with respect to the application to multi-stage uncertain optimization problems with more than two stages. Furthermore, the particular techniques in [93] require some direct knowledge of the underlying distribution. We are interested in the case where our only knowledge of the distribution comes indirectly through a (possibly noisy) sample of data.

■ 2.3.3 Multistage Stochastic Optimization

Many of the results stated thus far have also been extended to the multi-stage case. However, the multi-stage case poses many problems that have not been adequately addressed by the methods outlined above. Chance constraints, in particular, have not been successfully extended to the case of multi-stage stochastic optimization with more than two stages. One of the challenges in such formulations is enforcing the causality constraints in the adaptability functions of different stages. This is necessary in order to avoid overly optimistic formulations. This issue of causality is taken up again in Chapter 4.

Many results for multi-stage problems exploit special structure in the problem, that allows decomposition of the problem. The main idea is that in the presence of recourse when we are minimizing expectations, the objective function involves a linear term (in the case of Stochastic Linear Optimization) and then the expectation of the future costs, which is a convex term. Many approaches seek to approximate the convex term in the objective by means of supporting hyperplanes, building an outer linearization, and then solving the optimization by Benders decomposition. See, for example, [124], [33], and the references therein. These approaches, though, require special structure, and furthermore they typically lead to very large scale optimization problems, and therefore their applicability has limited scope.

Sampling approaches to the multistage stochastic optimization problem have been shown to be exponential in the number of stages. This is further discussed in Chapter 4.

■ 2.4 Sampling Results and Learning Theory

A particularly successful approach to Chance Constraints is that of sampling. Essentially, the idea is to sample some number N of the constraints from the distribution, and then find a solution that satisfies some fraction (typically all) of the sampled constraints. The sampling may in fact mean using data that are already available, such as the past behavior of a system (e.g., the stock market), or it could actually involve observing or simulating the system being modeled.

A significant advantage of the general sampling approach is that it does not require explicit knowledge of the underlying distribution. This is particularly appealing when we have large scale systems, where the only possibility is to observe the system's response to certain inputs. These ideas grew out of the statistics and learning literature (see below) and have also found application in Control (e.g., [139]).

First, we remark that there is a fundamental difference between approximating chance constraints, and sampling to approximate integrals, such as the expectations of the cost-to-go in the stochastic optimization approach with recourse, discussed above. The main advantage when it comes to chance constraints, is that they are expectations not of some arbitrary function, but of a very controlled function, namely, an indicator function. Thus the modulus of the integrand is constrained to lie in [0,1] from the very beginning.

The sampling approach to approximating chance constraints, then, replaces the set:

$$\mathcal{X}_\varepsilon \triangleq \{\boldsymbol{x} : \mathbb{P}(\boldsymbol{A}(\omega)\boldsymbol{x} \leq \boldsymbol{b}) \geq (1-\varepsilon)\},$$

by the random (convex) set

$$X_N(\omega_1,\ldots,\omega_N) \triangleq \left\{ \boldsymbol{x} : \begin{array}{c} \boldsymbol{A}(\omega_1)\boldsymbol{x} \leq \boldsymbol{b} \\ \vdots \\ \boldsymbol{A}(\omega_N)\boldsymbol{x} \leq \boldsymbol{b} \end{array} \right\}.$$

The fundamental question is, for a given number of samples, N, what can we say about the relation between the deterministic set $\mathcal{X}_\varepsilon$, and the random set $X_N(\omega_1,\ldots,\omega_N)$.

In addition to the probability of feasibility, ε, we must also incorporate the probability that the finite-length sample, $\Omega_N = \{\omega_1,\ldots,\omega_N\}$, is somehow not representative of the underlying distribution. We can think of this as the probability that the data sample generated is not typical of the true distribution, and instead looks like some other distribution. We call this notion reliability, and it is denoted by δ. The nature of the results that are of interest are so-called *sample complexity* results. These relate $\mathcal{X}_\varepsilon$, and $X_N(\omega_1,\ldots,\omega_N)$, and they say: If we have $N(\varepsilon,\delta)$ samples, then with probability at least $(1-\delta)$, we have $\mathcal{X}_N \subseteq \mathcal{X}_\varepsilon$. Or we might ask for something weaker, namely that with probability at least $(1-\delta)$, only our optimal solution, $\boldsymbol{x}^* \in \mathcal{X}_N$ satisfies $\boldsymbol{x}^* \in \mathcal{X}_\varepsilon$.

The central question is obtaining useful bounds on the sample complexity, $N(\varepsilon,\delta)$, for values of the feasibility, ε, and values for the reliability, δ.

■ 2.4.1 A Lower Bound

Consider a problem simpler than the one we are proposing: suppose we want to test the membership of a *given and fixed* vector $\boldsymbol{x}_0$, in $\mathcal{X}_\varepsilon$. Consider the random variable defined on Ω, given by

$$\zeta(\omega) = \begin{cases} 0, & \text{if } f(\boldsymbol{x}_0, \omega) \leq 0, \\ 1, & \text{otherwise.} \end{cases}$$

Consider *iid* copies of ζ, and the convergence of the empirical mean to the true mean. Without knowledge of any further structure, this defines a Bernoulli process. Sanov's theorem on finite alphabets (see, e.g., [53]) gives us a Large Deviations Principle, and thus exponential convergence of the measure to the deterministic dirac measure on the mean. The rate function given by Sanov's theorem is the relative entropy. By a Taylor series expansion, we can easily check that the relative entropy, $H(p|p+\varepsilon)$, has a leading term that is quadratic in ε, because the linear term drops out. On the other hand, near the boundary, namely, near $p = 1$, the leading term is in fact linear.[9] Therefore this (loose) argument shows that the sample complexity, $N(\varepsilon, \delta)$, will be related to $O(1/\varepsilon)$ and $O(\ln(1/\delta))$. Note that a straight application of Hoeffding's inequality ([76]) does not give us a lower bound, since the dependence of ε in Hoeffding's inequality is quadratic.[10] Hoeffding's inequality gives a large deviations result that is weaker than Sanov's theorem, and considerably weaker near the boundary.

We will see that we can achieve such a sample complexity behavior even for the more difficult question of determining membership in $\mathcal{X}_\varepsilon$ for a value $\boldsymbol{x}^*$ that depends on the sample (recall that in the above, $\boldsymbol{x}_0$ was chosen and fixed independently of the sample), or even when we ask for uniform results over the entire feasible set $\mathcal{X}_N$.

■ 2.4.2 Calafiore and Campi: Convexity

In the previous section, we give an intuitive idea of the best we might hope to do, in a general setting. In their paper [43], Calafiore and Campi show that they can essentially attain that lower bound, but for a harder problem. Namely, they are interested in the feasibility not of an *a priori* fixed and specified point $\boldsymbol{x}_0$, but rather of the point $\boldsymbol{x}^*$, that minimizes the objective function subject to satisfying all of the sampled constraints. In particular, this means that $\boldsymbol{x}^*$ is a function of the sampled data, and this considerably complicates the problem (we can no longer treat the samples of a random variable like ζ above as independent).

Their result comes directly from convexity properties. We give the main ideas here, and refer the interested reader to the paper [43] for the full details.

The result is quite powerful in its generality. Consider a single stage uncertain

[9] Essentially this is the difference between proper learning, and non-proper learning, as we discuss further below.

[10] Calafiore and Campi use Hoeffding's inequality to give a sample complexity bound for *a posteriori* estimates on probability of feasibility; the inequality they derive themselves, however, seems to be stronger because of its dependence only on $1/\varepsilon$. Furthermore, their result can easily (although there would be no reason to do it this way) be adapted to provide *a posteriori* bounds, and thus one can obtain an $O(1/\varepsilon)$ result in that way as well.

optimization problem:

$$\begin{aligned} \min: \quad & \mathbf{c}^\top \boldsymbol{x} \\ \text{s.t.}: \quad & \mathbb{P}_\omega(f(\boldsymbol{x},\omega) \le 0) \ge 1-\varepsilon \\ & \boldsymbol{x} \in \mathcal{X}. \end{aligned}$$

The assumptions in place are:

1. For any $\omega \in \Omega$, the sets $\{\boldsymbol{x} : f(\boldsymbol{x},\omega) \le 0\}$ are convex.

2. The deterministic constraint $\mathcal{X}$ is convex.

Their main result is:

Theorem 2.5 (Calafiore-Campi)
For the setup as above, if the N sampled constraints define a nonempty feasible (convex) set, then if $\boldsymbol{x}^*$ is the optimal feasible solution, we have $\boldsymbol{x}^* \in \mathcal{X}_\varepsilon$ with probability at least $(1-\delta)$, as long as

$$N \ge N(\varepsilon,\delta) = 2[n\frac{1}{\varepsilon}\ln\frac{1}{\varepsilon} + \frac{1}{\varepsilon}\ln\frac{1}{\delta} + n].$$

That is, with reliability at least $(1-\delta)$, $\boldsymbol{x}^*$ is feasible with probability at least $(1-\varepsilon)$ to the next sampled constraint.

The proof of this result relies exclusively on convexity properties. Each of the N samples, $\{\omega_i\}$, of the uncertainty corresponds to a constraint, $f(\boldsymbol{x},\omega_i) \le 0$, and hence also a convex set of feasible points,

$$\mathcal{X}_i \triangleq \{\boldsymbol{x} : f(\boldsymbol{x},\omega_i) \le 0\}.$$

They call a constraint corresponding to ω_i a *support constraint* if removing that constraint alone, results in a new optimal solution $\boldsymbol{x}^\dagger$ that is strictly better than $\boldsymbol{x}^*$.

A classical result of Helly says that for any arbitrary but finite collection of convex sets in $\mathbb{R}^n$, if every collection of $(n+1)$ sets has a nonempty intersection, then the intersection of the entire finite family is nonempty.

Essentially as a result of this, they show that given any problem of the above form, with $\mathcal{X} \subseteq \mathbb{R}^n$, there can be at most n support constraints.[11] Then the intuition of the result, is as follows. Consider $(N+1)$ constraints, and let $\boldsymbol{x}^*$ denote the optimal feasible solution. The optimal feasible solution $\boldsymbol{x}^{*,N}$ to the first N constraints will be infeasible to the $(N+1)^{st}$ constraint only if the $(N+1)^{st}$ constraint is a support constraint for

[11] Note that the notion of a support constraint is different from that of a *tight* constraint. Certainty we can have arbitrarily many tight constraints in $\mathbb{R}^n$, and indeed we have more than n for any degenerate solution of a linear optimization problem. In the presence of degeneracy, however, there are *no support constraints,* since if any single constraint is removed, because of the degeneracy, the optimal solution does not change.

x^*. Since there are at most n of these, and then by symmetry, the probability of this is at most $n/(N+1)$. This should give the idea of where the $(1/\varepsilon)$ term comes from. The full proof is more involved, and draws on some techniques from [66].

Convexity Is Necessary

We remark that the lower bound of the previous section, made no assumption whatsoever about convexity. Thus, we can think of convexity as the added price, here, that we pay in order to strengthen the result of the lower bound from an *a priori* specified point x_0, to the optimal solution x^* that depends on the sampled constraints.

We give a brief example here that shows that in the absence of convexity, we may not have sample complexity results that are independent of the size of the uncertainty set. Indeed, note that the sample complexity result given above, is dimension-dependent as there is a factor of n. This is also evident from the ideas behind the proof, since the maximum number of support constraints is equal to the dimension of the space. However, the dimension of the uncertainty set, Ω, does not play any role in the expression for the sample complexity.

Then, let M be some very large positive integer, much larger than any number N of points we can feasibly sample. Pick some real number $\omega \in [0,1]$, and use it to define the robust feasible set C, as follows: Let $\omega_i \in \{0,1\}$ denote the i^{th} digit in the dyadic expansion of ω. Then define C to be those numbers in $[0,1]$ that match the dyadic expansion of ω in the first M digits:

$$C \stackrel{\triangle}{=} \{x : x_i = \omega_i,\ 1 \le i \le M\}.$$

Now define the feasible set to be:

$$\mathcal{X}_{\text{true}} \stackrel{\triangle}{=} \left\{ (x,y) : \begin{array}{c} 0 \le y \le 1, \text{if } x \in C, \\ y = 0 \text{ for all other } x. \end{array} \right\}$$

Let m be uniformly distributed on $\{1, 2, \ldots, M\}$, and define the sets

$$\begin{array}{rcl} C(m_1,\ldots,m_k) & \stackrel{\triangle}{=} & \{x : x_{m_i} = \omega_{m_i},\ 1 \le i \le k\} \\ \mathcal{X}(m_1,\ldots,m_k) & \stackrel{\triangle}{=} & \left\{ (x,y) : \begin{array}{c} 0 \le y \le 1, \text{if } x \in C(m_1,\ldots,m_k), \\ y = 0 \text{ for all other } x. \end{array} \right\}. \end{array}$$

The sampled optimization problem, then, given sample $(m_1, \ldots, m_N)$, is:

$$\begin{array}{rl} \max : & y \\ \text{s.t.} : & (x,y) \in \mathcal{X}(m_1,\ldots,m_N). \end{array}$$

If $N << M$, then the probability that the next sample drawn, m_{N+1}, is one of those already sampled, is $N/M \approx 0$, and thus the probability that the optimal solution to the sampled problem, (x^*, y^*), is feasible, is $1/2$.

Remark 2.3
Finally, we mention here that while in spirit the Calafiore and Campi result is meant to be a robustness result, it nevertheless does not explicitly incorporate any robustness. We show in a later chapter, that if one solves a robust sampled robust problem (as opposed to an exact sampled robust problem) then there is room to improve upon the sample complexity estimates, as well as to introduce some new meaningful parameters (in addition to δ and ε which we have here) into the problem.

■ 2.4.3 Uniform Learning Results

In the past section we considered sampling complexity results driven by convexity considerations. In this section we review some results from the theory of statistical learning and uniform limit theorems, which we use substantially in Chapter 4. Statistical learning theory is a deep field with connections to statistics ([134],[135]) probability and functional analysis ([57], [136], [103]) and computer science ([4], [5], [115]), and we can only give a very brief introduction here, but we refer the interested reader to the references given above, and the wealth of references contained therein.

We focus our discussion on the classification problem, as this will be of most use to us later. The classification problem, defined on an input space $\mathcal{X}$, is as follows:

1. There is an unknown classification rule[12] that maps points of the input space Ω deterministically to a point in $\{0, 1\}$:
$$h_{\text{true}} : \Omega \to \{0, 1\}.$$

2. There is an unknown distribution, μ, on Ω.

3. We have a fixed set (possibly infinite) of classifiers, $\mathcal{H}$, that may or may not contain the true classifier h_{true}.

4. The Goal: Select a classifier $h \in \mathcal{H}$ to minimize the so-called classification error, i.e., the measure of miss-classified points:
$$\text{Error}(h) \triangleq \mu(\{\omega \in \Omega \,:\, h(\omega) \neq h_{\text{true}}(\omega)\}.$$

5. Input: The input to the problem is the so-called training data, a collection of correctly labeled points: $\{(\omega_1, y_1), \ldots, (\omega_N, y_N)\}$, where $\omega_i \in \Omega$, and $y_i \in \{0, 1\}$. The assumption is that each point, ω_i, is generated independently at random

[12] For now we assume that a deterministic classifier that has zero error in fact does exist.

from the underlying distribution μ. This is the only information the decision-maker has about the distribution, μ.

A central question that arises on the way to trying to compute an optimal classifier, is that of the quality of the data: If a classifier $\hat{h} \in \mathcal{H}$ has empirical error (also called *training error*) p on the training data, what can we say about its true error? In the case where $h_{\text{true}} \in \mathcal{H}$, which is the *proper learning* setup, there is always at least one $\hat{h} \in \mathcal{H}$ that has zero training error. Therefore we can take $p = 0$, and the question becomes: If a classifier has zero training error, what can we say about its true error.

Since the input space, Ω, is continuous, and the collection of classifiers potentially infinite, it is too much to expect that based on a *finite* amount of data, we can always compute the exactly correct classifier. Thus we must be satisfied with trying to compute an ε-optimal classifier, i.e., a classifier $\hat{h} \in \mathcal{H}$ such that $\text{Error}(\hat{h}) \leq \varepsilon$.

In addition to this, however, our finite data sample may be a "bad" sample, i.e., it may not be representative of the distribution, thus not giving us enough information to select a good classifier. This is known as the reliability of the data. We thus further relax our criterion, so that a particular procedure for selecting a classifier $\hat{h} \in \mathcal{H}$ is called a *learning algorithm* if when given enough data, then with probability at least $(1 - \delta)$ it produces a classifier that has error at most ε. This is the so-called PAC framework, of Probably Approximately Correct learning ([133]).

A natural algorithm to choose, is one that selects a classifier that has zero error on the training data. Then, in the context of the discussion above, the central question in analyzing this algorithm becomes: Given Ω and $\mathcal{H}$, and reliability and error parameters δ and ε, how big must be the number, N, of samples, to guarantee that with probability at least $(1 - \delta)$, any $h \in \mathcal{H}$ that has zero training error (i.e., perfectly classifies the training data) will have error at most ε. The minimum such number, which we denote by $N(\varepsilon, \delta)$, is called the *sample complexity*. Note that there is no mention of the distribution, μ. The sample complexity, then, guarantees a statement that holds for any distribution, μ.

The sample complexity is determined by the richness, or complexity, of the set of classifiers, $\mathcal{H}$. If, for instance, the classifier set $\mathcal{H}$ contains all possible mappings of Ω to $\{0, 1\}$, i.e., if $\mathcal{H} = 2^{\Omega}$, it is clear that there is no hope of a finite sample complexity, since for any N, there will always be a classifier that matches the true classifier perfectly on the data sample, but nowhere else. For any nonatomic measure μ, the error the classifier will be equal to 1.

For finite collections of classifiers, the complexity is controlled, and we can obtain sample complexity estimates in a straightforward manner, using tools like the union bound, and Hoeffding's inequality. For infinite classifiers, we need some measure of the complexity of the class. This is accomplished by the notion of the *growth function*. Given a set of m points, $(\omega_1, \ldots, \omega_m) \in \Omega^m$, define $\Pi_{\mathcal{H}}(\omega_1, \ldots, \omega_m)$ to be the number of distinct labellings of these points by the classifiers in $\mathcal{H}$. Then define:

$$\Pi_{\mathcal{H}}(m) \triangleq \max_{(\omega_1,\ldots,\omega_m)\in\Omega^m} \Pi_{\mathcal{H}}(\omega_1,\ldots,\omega_m).$$

We have the following fundamental theorem about proper learning (see, e.g., [4]):

Theorem 2.6
Given any $\varepsilon > 0$, and any classifier $h \in \mathcal{H}$ that has zero training error on a data sample of size m, where $m \geq 8/\varepsilon$, then

$$\mathbb{P}(\mathrm{Error}(h) \geq \varepsilon) \leq 2\Pi_{\mathcal{H}}(2m)2^{-\varepsilon m/2}.$$

Immediately we see that this error goes to zero with the number of samples, m, only if the growth function is not exponential in m.

VC Dimension and the Growth Function

The growth function can be upper bounded by a combinatorial quantity known as the VC dimension. This is a combinatorial measure of the complexity of the class $\mathcal{H}$.

Definition 2.1
The VC dimension of a set of classifiers, $\mathcal{H}$, is the maximum number, m, of points $\omega_1,\ldots,\omega_m \in \Omega^m$ that can be shattered. A set of points is said to be shattered by $\mathcal{H}$ if for every $\alpha \in \{0,1\}^m$, there exists some $h_\alpha \in \mathcal{H}$ such that

$$h_\alpha(\omega_1,\ldots,\omega_m) = (h_\alpha(\omega_1),\ldots,h(\omega_m)) = \alpha.$$

Therefore,

$$VC(\mathcal{H}) = \max\{m \,:\, \Pi_{\mathcal{H}}(m) = 2^m\}.$$

Furthermore, by a result known as Sauer's lemma (see, e.g., [37]), the growth function can be bounded by a polynomial of the VC dimension. Thus, the growth function if polynomial if and only if the VC dimension is finite. Therefore finiteness of the VC dimension exactly characterizes the cases when sample complexity, $N(\varepsilon,\delta)$, is finite.

In Chapter 4, we use these learnability results for families of classifiers with finite VC dimension. We go on to consider some related problems in learning, in particular, learning with a margin, and this allows us to consider more refined notions of complexity. We defer further discussion of this to Appendix B.

■ 2.5 Summary of the Current State of the Art

Having reviewed many old and recent results, in this section we provide a summary of the state of the art of the results available with respect to uncertainty, adaptability, and multi-stage optimization, in particular how it applies to our own framework.

1. Robust Optimization:

   (a) There has been little work in adaptability, and in the extension of the Robust Optimization framework to multistage optimization problems.

   (b) There seems to be no work that can accommodate discrete variables in second and later stages.

   (c) No work creates a hierarchy of adaptability, so that we can choose to increase the level of adaptability at the expense of greater computational resources.

2. Stochastic optimization:

   (a) Minimizing Expectation: This seems to be difficult. Results from information complexity indicate that the sample complexity of integration may be much greater than for testing feasibility. Furthermore, the complexity of integration in higher dimensions may well be prohibitive.

   (b) Recourse: There do not seem to be many results that extend to non-constant recourse. Also, much of the work done focuses on the case of complete recourse. Effectively this means that feasibility is guaranteed at the second stage, and thus considering the probability of feasibility of the first-stage variables is not an issue.

   (c) Chance constraints: For special structure, in particular, affine impact of the uncertainty on the problem parameters, there are approximation techniques ([93]), but these do not seem to extend to more general problems where the uncertainty affects the parameters in a nonlinear fashion.

   (d) Sampling chance constraints: There are importance-sampling techniques ([92]) with logarithmic dependence on the error, ε, but again these require affine dependence on the uncertainty, and furthermore make certain concentration assumptions on the underlying distributions. They do not seem to extend to more general problems.

   (e) Sampling chance constraints: The results of Campi and Calafiore ([43]) and de Farias and Van Roy ([50]) meet the lower bounds for sample complexity. Moreover the results of ([43]) make very few assumptions on the distribution. However, there is no extension to multi-stage problems.

CHAPTER 3

Finite Adaptability for Linear Optimization

The essence of adaptability, and therefore of this thesis, is the functional dependence of future stage decisions on past realizations of the uncertainty. Under the static robust optimization paradigm for linear optimization with deterministic parameter uncertainty, a decision maker selects a single robust solution in order to immunize the solution from parameter uncertainty, and future stage decisions are all determined at the initial time. In this chapter, we maintain the robust noise model, but consider a departure from the static paradigm, allowing the decision-maker some limited, finite adaptability. Here, the decision-maker can obtain some additional information about the uncertainty before committing to a decision. The central problem we address is optimally structuring this adaptability, and understanding its marginal value. We propose a hierarchy of increasing adaptability that bridges the gap between the static robust formulation, and the fully adaptable formulation. We study the geometry, complexity, formulations, algorithms, examples and computational results for finite adaptability. In contrast to the model of affine adaptability proposed in [12], our proposed framework can accommodate discrete variables. In terms of performance for continuous linear optimization, the two frameworks are complementary, in the sense that we provide examples that the proposed framework provides stronger solutions and vice versa.

■ 3.1 Introduction

Optimization under uncertainty has long been at the frontier of both theoretical and computational research. Stochastic optimization (see [35],[108], [118], [120], and references therein) explicitly incorporates a probabilistic description of the uncertainty, often relaxing hard constraints by penalizing infeasibility ([113]), or by using so-called chance constraints ([107]). Stochastic Optimization methods, including chance constraints are discussed in more detail in Chapter 2. In the last decade, much work has been done in robust optimization. Here, the decision-maker makes no probabilistic assumptions, but rather seeks deterministic protection to some bounded level of uncertainty. Recent work has considered the case of linear, semidefinite, and general conic optimization, as well as discrete robust optimization; see, e.g., [14],[15], [25], [26], [71].

In multi-stage optimization problems, the uncertainty is revealed sequentially, and hence may be partially known at the time when some decisions are made (see [49], [108], [120] and Chapter 2 for further discussion of this in the Stochastic Optimization formulation of uncertainty). The focus of this chapter is on two-stage optimization models, where the uncertainty follows the robust paradigm, i.e., it is set-based and deterministic:

$$\begin{aligned} \min: \quad & \boldsymbol{c}^{\top}\boldsymbol{x} + \boldsymbol{d}^{\top}\boldsymbol{y}(\boldsymbol{\omega}) \\ \text{s.t.}: \quad & \boldsymbol{A}(\boldsymbol{\omega})\boldsymbol{x} + \boldsymbol{B}(\boldsymbol{\omega})\boldsymbol{y}(\boldsymbol{\omega}) \leq \boldsymbol{b}, \quad \forall \boldsymbol{\omega} \in \Omega. \end{aligned} \tag{3.1.1}$$

We investigate the class of piecewise constant adaptability functions for $\boldsymbol{y}(\boldsymbol{\omega})$. For much of this chapter, we simplify the setting by focusing only on the second stage problem, assuming that the first stage variable $\boldsymbol{x}$ has already been fixed and implemented. Thus, by adding an additional variable if necessary, the second stage problem is:

$$\begin{aligned} \min: \quad & \boldsymbol{d}^{\top}\boldsymbol{y}(\boldsymbol{\omega}) \\ \text{s.t.}: \quad & \boldsymbol{B}(\boldsymbol{\omega})\boldsymbol{y}(\boldsymbol{\omega}) \leq \boldsymbol{b}, \quad \forall \boldsymbol{\omega} \in \Omega. \end{aligned} \tag{3.1.2}$$

Remark 3.1
While our central motivation is the two-stage optimization model (and extensions to multi-stage problems), it is also interesting to consider the second stage problem in isolation, as a single stage problem. In this context, piecewise constant adaptability to the uncertainty, $\boldsymbol{\omega}$, is equivalent to a formulation where the decision-maker receives some advance partial information about the realization of the uncertainty, namely, the uncertainty realization will lie in some given region of a partition of the uncertainty set Ω.

For deterministic uncertainty models, the landscape of solution concepts has two extreme cases. On the one side, we have the static robust formulation where the decision-maker has no adaptability to, or information about, the realization of the uncertainty. As discussed in Chapter 2, this typically yields overly conservative, or pessimistic, solutions.

On the other extreme is the formulation with complete adaptability, where the decision-maker has arbitrary adaptability to the exact realization of the uncertainty and then selects an optimal solution accordingly.[1] This set-up is overly optimistic for

[1] In the context of a single-stage problem, this corresponds to have complete knowledge of the exact realization of the uncertainty, as opposed to some coarse model for the advance information. As we comment throughout the chapter, while we focus on the two-stage model, the interpretation of the adaptability we introduce, in the one-stage model, is exactly one corresponding to a finite amount of information revealed to the decision-maker.

several reasons. Exact observations of the uncertainty are rarely possible. Moreover, even if in principle feasible, computing the optimal arbitrarily adaptable second stage function is typically an intractable problem. Furthermore, even implementing such complete adaptability in practice may be too expensive, since effectively it requires complete flexibility in the second stage, and hence in itself may be undesirable. This motivates us to consider the middle ground.

Contributions and Chapter Outline

In a departure from the static robust optimization paradigm, we consider a set-up where the decision-maker may (perhaps at some cost) be able to select some finite number, k, of contingency plans for the second stage solution, $(y_1, \dots, y_k)$, as opposed to a single robust solution, y_R. The central topic of this chapter is to understand the structure, properties and value of this finite adaptability.

Our goals in this chapter are as follows:

(1) To provide a model of adaptability that addresses the conservativeness of the static robust formulation in the case of the second stage of a two-stage optimization problem, viewed as a *single-stage optimization*. We then apply this to multi-stage optimization.

(2) To develop a hierarchy of adaptability that bridges the gap between the static robust and completely adaptable formulations, as the level, k, of adaptability increases.

(3) To investigate how to optimally structure the adaptability (i.e., how to choose the contingency plans) for small k. Furthermore, we want to understand the complexity of solving the problem optimally.

(4) In addition to structural properties and theoretical characterizations of the optimal adaptability structure, we would like practical algorithms that perform well in computational examples.

Point by point, we believe the above goals are important for the following reasons. (1) While there exist proposals for adaptability, to the best of our knowledge none are structured specifically to address the fact that the static robust formulation cannot model non-convexity in the uncertainty set, or non-constraintwise uncertainty ([15]). (2) Also, as far as we know, there exist no adaptability proposals that allow a variable degree of adaptability, specifically with the ability to cover the middle ground between the static robust and completely adaptable formulations. (3) The completely adaptable formulation is known to be NP-hard to solve in general ([12]) as are other adaptability proposals ([12], [127],[6]), as well as various approaches to Stochastic Programming and chance constraints ([108]). It is important, then, to try to understand how much is possible, and the complexity of achieving it. (4) Given the inherent difficulty of these problems, efficient practical algorithms are of high importance.

In Section 3.2, we provide the basic setup of our adaptability proposal, and we define the problem of selecting k contingency plans. Because of its inherent discrete nature, this proposal can accommodate discrete variables. To the best of our knowledge, this is the first proposal for adaptability that can reasonably deal with discrete variables. In Section 3.3, we give a geometric interpretation of the conservativeness of the static robust formulation. We provide a geometric characterization of when finite adaptability can improve the static robust solution by η, for any (possibly large) chosen $\eta \geq 0$. We obtain necessary conditions that any finite adaptability scheme must satisfy in order to improve the static robust solution by at least η. The full collection of these conditions also constitutes a sufficient condition for η improvement.

In Section 3.4, we consider an exact formulation of the k-adaptability problem. In the general case, we show it can be formulated as a bilinear optimization problem. For the special case of right hand side uncertainty, we show that the bilinear optimization becomes a discrete optimization problem and we provide an integer optimization formulation for the $k = 2$ contingency plan problem. In Section 3.5, we consider the complexity of optimally computing k-adaptability, and we show that structuring the $k = 2$ adaptability optimally, is NP-hard in the minimum of the dimension of the uncertainty, the dimension of the problem, and the number of constraints affected. In particular, we show that if the minimum of these three quantities is small, then optimally structuring 2-adaptability is theoretically tractable.

In Section 3.6, we consider an example in detail, illustrating several of the subtleties of the geometric characterizations of Section 3.3. Here, we also compare k-adaptability to the affine adaptability proposal of [12]. Following that work, there has been renewed interest in adaptability (e.g., [6],[60],[46],[127]). Our work differs from continuous adaptability proposals in several important ways. First, our model offers a natural hierarchy of increasing adaptability. Second, the intrinsic discrete aspect of the adaptability proposal makes this suitable for any situation where it may not make sense to require information about infinitesimal changes in the data. Indeed, only coarse observations may be available. In addition, especially from a control viewpoint, infinite (and thus infinitesimal) adjustability as required by the affine adaptability framework, may not be feasible, or even desirable. We provide an example where affine adaptability is no better than the static robust solution, while finite adaptability with 3 contingency plans significantly improves the solution.

In Section 3.7, we provide a heuristic algorithm based on the qualitative prescriptions of Section 3.3. This algorithm is also suitable for solving problems with discrete variables, where if the original discrete static robust problem is computationally tractable, so is our algorithm. Section 3.8 provides several computational examples, continuous and discrete, illustrating the efficient algorithm of Section 3.7. First in Section 3.8.1, we discuss an application to Air Traffic Control. This application is further considered in Chapter 5 (see also [23]), but we introduce it here to illustrate applicability of the proposed approach, and as an opportunity to discuss when we expect finite adaptability to be appropriate for large scale applications. Finally, in Section 3.8.2 and Section 3.8.3, we consider a large collection of randomly generated scheduling problems in an effort to obtain some appreciation in the generic case, for the benefit of the first few levels of the adaptability hierarchy.

■ 3.2 Definitions

We consider linear optimization problems with deterministic uncertainty in the coefficients, where the uncertainty set is polyhedral. Uncertainty in the right hand side or in the objective function can be modeled by uncertainty in the matrix (see, e.g., [25]). In Section 3.2.1, we define the static robust formulation, the completely adaptable formulation, and our finite adaptability formulation. In Section 3.2.2, we focus on the second-stage problem, and write it as a single-stage problem with adaptability. While our central focus is the two-stage model, we consider this simpler problem.

■ 3.2.1 Static Robustness, Complete and Finite Adaptability

The general two-stage problem we consider, and wish to approximate, is the one with complete adaptability, that can be formulated as:

$$\begin{aligned}\mathrm{CompAdapt}(\Omega) &\triangleq \left[\begin{array}{ll}\min: & \boldsymbol{c}^{\top}\boldsymbol{x}+\boldsymbol{d}^{\top}\boldsymbol{y}(\omega)\\ \text{s.t.}: & \boldsymbol{A}(\omega)\boldsymbol{x}+\boldsymbol{B}(\omega)\boldsymbol{y}(\omega)\leq\boldsymbol{b}, \quad \forall\omega\in\Omega\end{array}\right] \qquad (3.2.3)\\ &= \max_{\omega\in\Omega}\left[\begin{array}{ll}\min: & \boldsymbol{c}^{\top}\boldsymbol{x}+\boldsymbol{d}^{\top}\boldsymbol{y}\\ \text{s.t.}: & \boldsymbol{A}(\omega)\boldsymbol{x}+\boldsymbol{B}(\omega)\boldsymbol{y}\leq\boldsymbol{b}\end{array}\right]\end{aligned}$$

Note that only the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ have an explicit dependence on the uncertain parameter, ω, while the vectors $\boldsymbol{c}, \boldsymbol{d}$, and $\boldsymbol{b}$ are taken to be deterministically and exactly known. As discussed in Chapter 2, this assumption can be made without loss of generality, since by means of simple transformations we can bring an arbitrary (i.e., uncertain objective function or right hand side vector $\boldsymbol{b}$) problem into one of the above form. Here we have made no assumption about the nature of the adaptability of the second stage function, $\boldsymbol{y}(\omega)$, and thus we label this the completely adaptable case. We assume throughout this chapter that the parameters of the problem (that is, the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$) depend affinely on the uncertain parameter ω.

On the other end of the spectrum from the completely adaptable formulation, is the static robust formulation, where the second stage variables have no dependence on ω:

$$\mathrm{Static}(\Omega) \triangleq \begin{bmatrix} \min: & \boldsymbol{c}^\top \boldsymbol{x} + \boldsymbol{d}^\top \boldsymbol{y} & \\ \text{s.t.}: & \boldsymbol{A}(\omega)\boldsymbol{x} + \boldsymbol{B}(\omega)\boldsymbol{y}(\omega) \leq \boldsymbol{b}, & \forall \omega \in \Omega \end{bmatrix} \tag{3.2.4}$$

We assume throughout that the static robust problem (3.2.4) is feasible. In particular, this implies that the nominal problem is feasible for $(\boldsymbol{A}(\omega), \boldsymbol{B}(\omega))$ for every $\omega \in \Omega$. Therefore both Static($\mathcal{P}$) and CompAdapt($\mathcal{P}$) are finite.

In the k-adaptability problem, the decision-maker chooses k second-stage solutions, $\{\boldsymbol{y}_1, \dots, \boldsymbol{y}_k\}$, and then commits to one of them only after seeing the realization of the uncertainty. At least one of the k solutions must be feasible regardless of the realization of the uncertainty:

$$\mathrm{Adapt}_k(\Omega) \triangleq \begin{bmatrix} \min: & \boldsymbol{c}^\top \boldsymbol{x} + \max\{\boldsymbol{d}^\top \boldsymbol{y}_1, \dots, \boldsymbol{d}^\top \boldsymbol{y}_k\} & \\ \text{s.t.}: & \left\{ \begin{array}{c} \boldsymbol{A}(\omega)\boldsymbol{x} + \boldsymbol{B}(\omega)\boldsymbol{y}_1 \geq \boldsymbol{b} \\ \text{or} \\ \boldsymbol{A}(\omega)\boldsymbol{x} + \boldsymbol{B}(\omega)\boldsymbol{y}_2 \geq \boldsymbol{b} \\ \text{or} \\ \vdots \\ \text{or} \\ \boldsymbol{A}(\omega)\boldsymbol{x} + \boldsymbol{B}(\omega)\boldsymbol{y}_k \geq \boldsymbol{b} \end{array} \right\} & \forall \omega \in \Omega \end{bmatrix}. \tag{3.2.5}$$

This is a disjunctive optimization problem with infinitely many constraints. In Section 3.4, we formulate this as a (finite) bilinear optimization problem.

If we think of the collection of k second stage vectors, $(\boldsymbol{y}_1, \dots, \boldsymbol{y}_k)$ as contingency plans, where each is implemented depending on the realization of the uncertainty, then the k-adaptability problem becomes a k-partition problem. The decision-maker selects a partition of the uncertainty set Ω into k (possibly non-disjoint) regions: $\Omega = \Omega_1 \cup \dots \cup \Omega_k$. We can reformulate the optimal k-adaptability problem in (3.2.5) as

$$\mathrm{Adapt}_k(\Omega) = \min_{\Omega = \Omega_1 \cup \cdots \cup \Omega_k} \begin{bmatrix} \min: & \boldsymbol{c}^\top \boldsymbol{x} + \max\{\boldsymbol{d}^\top \boldsymbol{y}_1, \dots, \boldsymbol{d}^\top \boldsymbol{y}_k\} & \\ \text{s.t.}: & \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{y}_1 \geq \boldsymbol{b}, & \forall (\boldsymbol{A}, \boldsymbol{B}) \in \Omega_1 \\ & \vdots & \\ & \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{y}_k \geq \boldsymbol{b}, & \forall (\boldsymbol{A}, \boldsymbol{B}) \in \Omega_k \end{bmatrix}. \tag{3.2.6}$$

We state and prove this equivalence formally.

Proposition 3.1

The formulations (3.2.5) and (3.2.6) are equivalent in the following sense: If $(\boldsymbol{y}_1, \dots, \boldsymbol{y}_k)$ is a feasible solution to (3.2.5), there is a partition $(\Omega_1 \cup \dots \cup \Omega_k)$ so that $((\Omega_1 \cup \dots \cup \Omega_k), (\boldsymbol{x}, \boldsymbol{y}_1, \dots, \boldsymbol{y}_k))$ is a feasible solution for (3.2.6). Conversely, if $((\Omega_1 \cup \cdots \cup \Omega_k), (\boldsymbol{x}, \boldsymbol{y}_1, \dots, \boldsymbol{y}_k))$ is a feasible solution for (3.2.6), then $(\boldsymbol{x}, \boldsymbol{y}_1, \dots, \boldsymbol{y}_k)$ is a feasible solution for (3.2.5).

PROOF. If $(\boldsymbol{x}, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_k)$ is a feasible solution to (3.2.5), then define $\Omega_i = \{\omega \in \Omega : \boldsymbol{A}(\omega)\boldsymbol{x} + \boldsymbol{B}(\omega)\boldsymbol{y}_i \geq \boldsymbol{b}\}$, for $1 \leq i \leq k$. By assumption, $(\boldsymbol{x}, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_k)$ is feasible for (3.2.5), and hence $\Omega = \Omega_1 \cup \cdots \cup \Omega_k$. Therefore, the $\{\Omega_i\}$ indeed constitute a partition, and $((\Omega_1 \cup \cdots \cup \Omega_k), (\boldsymbol{x}, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_k))$ is feasible for (3.2.6). Conversely, let $((\Omega_1 \cup \cdots \cup \Omega_k), (\boldsymbol{x}, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_k))$ be a feasible solution for (3.2.6). Then, for any $\omega \in \Omega$, we must have $\omega \in \Omega_i$ for some $1 \leq i \leq k$, by definition. But then $\boldsymbol{A}(\omega)\boldsymbol{x} + \boldsymbol{B}(\omega)\boldsymbol{y}_i \geq \boldsymbol{b}$, and $(\boldsymbol{x}, \boldsymbol{y}_1, \ldots, \boldsymbol{y}_k)$ is feasible for (3.2.5). □

Throughout this chapter we refer equivalently to either k contingency plans, or k-partitions, for the k-adaptability problem. The inequalities $\text{Static}(\mathcal{P}) \geq \text{Adapt}_k(\mathcal{P}) \geq \text{CompAdapt}(\mathcal{P})$ hold in general.

In the area of multi-stage optimization, there has been significant effort to model the sequential nature of the uncertainty, specifically modeling the fact that some variables may be chosen with (partial) knowledge of the uncertainty. This is often known as recourse ([49],[108]). In [12], the authors consider a multi-stage problem with deterministic uncertainty, where the variables in stage t are affine functions of the uncertainty revealed up to time t. We henceforth refer to this model as *affine adaptability*. The affine adaptability approximation to (3.2.3) is

$$\text{Affine}(\mathcal{P}) \triangleq \left[\begin{array}{ll} \min: & \boldsymbol{c}^\top \boldsymbol{x} + \boldsymbol{d}^\top \boldsymbol{y}(\omega) \\ \text{s.t.}: & \boldsymbol{A}(\omega)\boldsymbol{x} + \boldsymbol{B}(\omega)\boldsymbol{y}(\omega) \geq \boldsymbol{b}, \quad \forall \omega \in \Omega \end{array}\right], \tag{3.2.7}$$

where $\boldsymbol{y}(\omega)$ is an affine function of the uncertain parameter, ω:

$$\boldsymbol{y}(\omega) = \boldsymbol{Q}\omega + \boldsymbol{q},$$

as discussed in Chapter 2. The authors show that computing affine adaptability is in general NP-hard, although in some cases it can be well approximated tractably.

We show by example that our finite adaptability proposal is not comparable to affine adaptability, in the sense that in some cases affine adaptability fails where finite adaptability succeeds, and vice versa. Unlike affine adaptability, we provide a full hierarchy of levels of adaptability. Furthermore, all levels of the hierarchy require only finite adaptability, and coarse observations. On the other hand, affine adaptability, while restricted compared to complete adaptability, nevertheless requires exact observation of the uncertainty. Furthermore, the recourse decision-variables are infinitely, and hence infinitesimally, adjustable.

■ 3.2.2 The Second-Stage Optimization

While the central motivation of this thesis, and this chapter, is the multi-stage optimization problem, and in this chapter in particular, the two-stage optimization problem, we focus explicitly on the second stage problem. We formulate the second stage optimization problem as a single stage optimization problem with adaptability. This corresponds to single-stage problem we have once the first-stage variable, $\boldsymbol{x}$, has been fixed. For notational simplicity, we use the same notation, and thus write $\boldsymbol{b}$ for the right hand side, rather than writing

$$\hat{\boldsymbol{b}}(\omega) = (\boldsymbol{b} - \boldsymbol{A}(\omega)\boldsymbol{x}),$$

and then adding additional variables so as to rewrite this in terms of yet another deterministic right hand side vector $\hat{\hat{\boldsymbol{b}}}$.

Thus, in this single-stage context, our definitions from Section 3.2.1 have the following representations. The static robust problem is:

$$\text{Static}(\Omega) \triangleq \left[\begin{array}{ll} \min: & \boldsymbol{d}^\top \boldsymbol{y} \\ \text{s.t.}: & \boldsymbol{B}(\omega)\boldsymbol{y} \geq \boldsymbol{b}, \quad \forall \omega \in \Omega \end{array} \right].$$

The completely adaptable formulation is now:

$$\begin{aligned} \text{CompAdapt}(\Omega) &\triangleq \left[\begin{array}{ll} \min: & \boldsymbol{d}^\top \boldsymbol{y}(\omega) \\ \text{s.t.}: & \boldsymbol{B}(\omega)\boldsymbol{y}(\omega) \geq \boldsymbol{b}, \quad \forall \omega \in \Omega \end{array} \right] \\ &= \max_{\omega\in\Omega} \left[\begin{array}{ll} \min: & \boldsymbol{d}^\top \boldsymbol{y} \\ \text{s.t.}: & \boldsymbol{B}(\omega)\boldsymbol{y} \geq \boldsymbol{b} \end{array} \right]. \end{aligned}$$

And finally, the k-adaptability formulation is:

$$\begin{aligned} \text{Adapt}_k(\Omega) &\triangleq \left[\begin{array}{ll} \min: & \max\{\boldsymbol{d}^\top \boldsymbol{y}_1, \ldots, \boldsymbol{d}^\top \boldsymbol{y}_k\} \\ \text{s.t.}: & [\boldsymbol{B}(\omega)\boldsymbol{y}_1 \geq \boldsymbol{b} \ \text{ or } \ \boldsymbol{B}(\omega)\boldsymbol{y}_2 \geq \boldsymbol{b} \ \text{ or } \ \cdots \ \text{ or } \ \boldsymbol{B}(\omega)\boldsymbol{y}_k \geq \boldsymbol{b}] \quad \forall \omega \in \Omega \end{array} \right] \\ &= \min_{\Omega=\Omega_1\cup\cdots\cup\Omega_k} \left[\begin{array}{ll} \min: & \max\{\boldsymbol{d}^\top \boldsymbol{y}_1, \ldots, \boldsymbol{d}^\top \boldsymbol{y}_k\} \\ \text{s.t.}: & \boldsymbol{B}(\omega)\boldsymbol{y}_1 \geq \boldsymbol{b}, \quad \forall \omega \in \Omega_1 \\ & \quad \vdots \\ & \boldsymbol{B}(\omega)\boldsymbol{y}_k \geq \boldsymbol{b}, \quad \forall \omega \in \Omega_k \end{array} \right]. \end{aligned}$$

We remark again here, that in this single stage formulation, the k-adaptability formulation can also have the interpretation of the decision-maker obtaining side-information about the uncertainty, before it is fully revealed. That is, we can equivalently consider the problem where the decision-maker selects the partition of Ω into k regions, and then receives advance information that the uncertainty realization will fall in region i, $1 \leq i \leq k$.

3.3 A Geometric Perspective

The problem we consider in this section is the second-stage formulation given above:

$$\begin{array}{rl} \min: & \boldsymbol{d}^{\top}\boldsymbol{y} \\ \text{s.t.}: & \boldsymbol{B}(\omega)\boldsymbol{y} \leq \boldsymbol{b}, \quad \forall \omega \in \Omega. \end{array}$$

We assume throughout, that there are m constraints, and $\boldsymbol{y} \in \mathbb{R}^n$. It is convenient for some of our geometric results to re-parameterize the uncertainty set Ω in terms of the actual matrices, $\boldsymbol{B}(\omega)$, rather than the space of the uncertain parameter, Ω. Then we define:

$$\mathcal{P} \triangleq \{\boldsymbol{B} = \boldsymbol{B}(\omega) \,:\, \omega \in \Omega\}.$$

Thus, for example, the static problem now becomes:

$$\text{Static}(\mathcal{P}) \triangleq \left[\begin{array}{rl} \min: & \boldsymbol{d}^{\top}\boldsymbol{y} \\ \text{s.t.}: & \boldsymbol{B}\boldsymbol{y} \geq \boldsymbol{b}, \quad \forall \boldsymbol{B} \in \mathcal{P} \end{array} \right], \tag{3.3.8}$$

We assume throughout, that the uncertainty set $\mathcal{P}$ is a polytope. We consider both the case where $\mathcal{P}$ is given as a convex hull of its extreme points, and where it is given as the intersection of half-spaces. Some results are more convenient to present in the case of the convex hull representation.

In this section, we provide a geometric view of the gap between the completely adaptable and static robust formulations, and also of the way in which finite adaptability bridges this gap. The key intuition is that the static robust formulation is inherently unable to model non-constraintwise uncertainty and, as is explained below, effectively replaces any given uncertainty set $\mathcal{P}$, with a potentially much larger uncertainty set.

We then use this geometric interpretation to obtain necessary conditions that any k-partition must satisfy in order to improve the static robust solution value by at least η, for any chosen value η.

3.3.1 The Geometric Gap

Since we consider matrix uncertainty, the elements of $\mathcal{P}$ are $m \times n$ matrices, $\boldsymbol{B} = (b_{ij})$, $1 \leq i \leq m$ and $1 \leq j \leq n$. Given any uncertainty region $\mathcal{P}$, let $\pi_l(\mathcal{P})$ denote the projection of $\mathcal{P}$ onto the components corresponding to the l^{th} constraint of (3.3.8), i.e., this is the projection onto the l^{th} row of the matrix:

$$\pi_l(\mathcal{P}) = \{(b_{l1}, \ldots, b_{ln}) \,:\, (\hat{b}_{ij}) \in \mathcal{P}, \hat{b}_{lj} = b_{lj}, 1 \leq j \leq n\}.$$

Then, we define:

$$(\mathcal{P})_R \triangleq \text{conv}(\pi_1\mathcal{P}) \times \text{conv}(\pi_2\mathcal{P}) \times \cdots \times \text{conv}(\pi_m\mathcal{P}). \tag{3.3.9}$$

The set $(\mathcal{P})_R$ is the smallest hypercube (in the above sense) that contains the set $\mathcal{P}$. In this chapter we focus on polyhedral uncertainty sets, which are thus already convex. We consider the application of the ideas presented here to non-convex uncertainty sets elsewhere. See Figure 3-1 for a very simple illustration of this definition.

Lemma 3.2 *For $\mathcal{P}$ and $(\mathcal{P})_R$ defined as above, we have*

(a) $\text{Static}(\mathcal{P}) = \text{Static}((\mathcal{P})_R)$ *and* $\text{Static}(\mathcal{P}) = \text{CompAdapt}((\mathcal{P})_R)$.

(b) *For $\mathcal{P} = \mathcal{P}_1 \cup \cdots \cup \mathcal{P}_k$ the optimal k-partition of the uncertainty set,*

$$\text{Adapt}_k(\mathcal{P}) = \text{CompAdapt}((\mathcal{P}_1)_R \cup \cdots \cup (\mathcal{P}_k)_R).$$

(c) *There is a sequence of partitions $\{\mathcal{P}_{k1} \cup \mathcal{P}_{k2} \cup \cdots \cup \mathcal{P}_{kk}\}$ so that*

$$(\mathcal{P}_{k1})_R \cup \cdots \cup (\mathcal{P}_{kk})_R \longrightarrow \mathcal{P}, \quad k \to \infty.$$

The first part of the lemma says that the static robust formulation cannot capture non-convexity in the uncertainty set, nor can it model correlation across different constraints. Furthermore, it says that this is exactly the reason for the gap between the static robust formulation, and the completely adaptable formulation. The second part of the lemma explains from a geometric perspective why, and how, the adaptive solution improves the static robust cost. The third part gives a geometric interpretation of how finite adaptability bridges the gap between the static robust and completely adaptable formulations.

PROOF. **(a)** This assertion follows from duality theory (also, see [15]).

(b) Given a partition $\mathcal{P}_1 \cup \cdots \cup \mathcal{P}_k$, the optimal k contingency plans $(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_k)$ are to take $\boldsymbol{y}_i$ as the static robust solution to the problem with restricted uncertainty set $\mathcal{P}_i$. Therefore by part **(a)**, $\boldsymbol{d}^\top \boldsymbol{y}_i = \text{Static}(\mathcal{P}_i) = \text{CompAdapt}((\mathcal{P}_i)_R)$. The worst-case value is then:

$$\max_{1 \le i \le k} \boldsymbol{d}^\top \boldsymbol{y}_i = \max_{1 \le i \le k} \text{CompAdapt}((\mathcal{P}_i)_R) = \text{CompAdapt}((\mathcal{P}_1)_R \cup \cdots \cup (\mathcal{P}_k)_R).$$

(c) It suffices to consider any sequence of partitions where the maximum diameter of any region goes to zero as $k \to \infty$. As the diameter of any region goes to zero, the smallest hypercube (in the sense of (3.3.9)) also shrinks to a point. □

Example: To illustrate this geometric concept, consider the constraints $\{b_{11}y_1 \le 1, b_{22}y_2 \le 1\}$, where the uncertainty set is $\mathcal{P} = \{(b_{11}, b_{22}) : 0 \le b_1, b_2 \le 1, b_1 + b_2 \le 1\}$ (and $b_{12} = b_{21} = 0$). The set $\mathcal{P}$ can be identified with the simplex in $\mathbb{R}^2$. The set $(\mathcal{P})_R$, then, is the unit square. The sets $\mathcal{P}$, $(\mathcal{P})_R$, and various partitions, are illustrated in Figure 3-1. ▲

We would like to conclude from Lemma 3.2 that k-adaptability bridges the gap be-

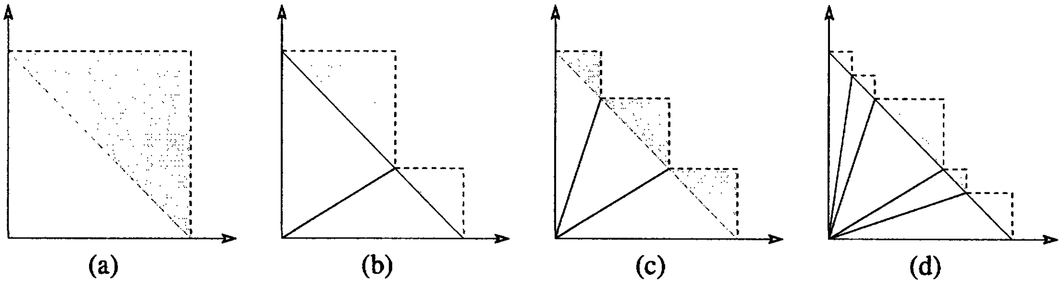

Figure 3-1. This figure illustrates the definition in (3.3.9), and Lemma 3.2. Let $\mathcal{P} = \{(b_{11}, b_{12}, b_{21}, b_{22}) : 0 \leq b_{11}, b_{22} \leq 1, b_{11} + b_{22} \leq 1, b_{12} = b_{21} = 0\}$. We identify $\mathcal{P}$ with a subset of the plane. The unshaded triangle in Figure (a) illustrates the set $\mathcal{P} = \{(b_{11}, b_{22}) : 0 \leq b_{11}, b_{22} \leq 1, b_{11} + b_{22} \leq 1\}$. The set $(\mathcal{P})_R$ is the entire square, and the shaded part is the difference, $(\mathcal{P})_R \setminus \mathcal{P}$. Figures (b),(c), and (d) show three successively finer partitions, illustrating how $(\mathcal{P}_{k1})_R \cup \cdots \cup (\mathcal{P}_{kk})_R \longrightarrow \mathcal{P}$.

tween the static robust and completely adaptable values, i.e., $\text{Adapt}_k(\mathcal{P}) \to \text{CompAdapt}(\mathcal{P})$ as k increases. With an additional continuity assumption, the proposition below asserts that this is in fact the case.

Continuity Assumption: *For any $\varepsilon > 0$, for any $\boldsymbol{B} \in \mathcal{P}$, there exists $\delta > 0$ and a point $\boldsymbol{y}$, feasible for $\boldsymbol{B}$ and within ε of optimality, such that $\forall\ \boldsymbol{B}' \in \mathcal{P}$ with $d(\boldsymbol{B}, \boldsymbol{B}') \leq \delta$, $\boldsymbol{y}$ is also feasible for $\boldsymbol{B}'$.*

The Continuity Assumption is relatively mild. It asks that if two matrices are infinitesimally close (here $d(\cdot,\cdot)$ is the usual notion of distance) then there should be a point that is almost optimal for both. Therefore, any problem that has an almost-optimal solution in the strict interior of the feasibility set, satisfies the Continuity Assumption. If the Continuity Assumption does not hold, then note that any optimization model requires exact (completely noiseless) observation of $\boldsymbol{B}$ in order to approach optimality.

Proposition 3.3

If the Continuity Assumption holds, then for any sequence of partitions of the uncertainty set, $\{\mathcal{P} = ((\mathcal{P}_{k1})_R \cup \cdots \cup (\mathcal{P}_{kk})_R)\}_{k=1}^{\infty}$, with the diameter of the largest set going to zero, the value of the adaptable solution approaches the completely adaptable value. In particular,

$$\lim_{k \to \infty} \text{Adapt}_k(\mathcal{P}) = \text{CompAdapt}(\mathcal{P}).$$

PROOF. Using Lemma 3.2 parts **(b)** and **(c)**, the proposition says that as long as the Continuity Assumption holds, then

$$[(\mathcal{P}_{k1})_R \cup \cdots \cup (\mathcal{P}_{kk})_R \longrightarrow \mathcal{P}] \Longrightarrow [\text{CompAdapt}((\mathcal{P}_{k1})_R \cup \cdots \cup (\mathcal{P}_{kk})_R) \longrightarrow \text{CompAdapt}(\mathcal{P})].$$

Indeed, given any $\varepsilon > 0$, for every $\boldsymbol{B} \in \mathcal{P}$, consider the $\delta(\boldsymbol{B})$-neighborhood around $\boldsymbol{B}$ as given by the Continuity Assumption. These neighborhoods form an open cover of

$\mathcal{P}$. Since $\mathcal{P}$ is compact, we can select a finite subcover. Let the partition $\mathcal{P} = \mathcal{P}_1 \cup \cdots \cup \mathcal{P}_k$ be (the closure of) such a subcover. Then, by the Continuity Assumption, $\mathrm{Static}(\mathcal{P}_i) \leq \mathrm{CompAdapt}(\mathcal{P}_i) + \varepsilon$. By definition, $\mathrm{CompAdapt}(\mathcal{P}_i) \leq \max_j \mathrm{CompAdapt}(\mathcal{P}_j) = \mathrm{CompAdapt}(\mathcal{P})$. We have shown that there exists a single sequence of partitions for which the corresponding adaptable solution value approaches the value of complete adaptability. This implies that $\mathrm{Adapt}_k(\mathcal{P}) \to \mathrm{CompAdapt}(\mathcal{P})$. Then recalling that the value of a linear optimization problem is continuous in the parameters, the proof is complete, as any sequence of partitions with diameter going to zero, eventually is a refinement of (a perturbation of) any given finite partition. We give an example in Section 3.6 that shows that the Continuity Assumption cannot be removed. □

■ 3.3.2 Necessary Conditions for η-Improvement

In Section 3.3.1, we use duality to show that the static robust problem and the k-adaptability problem are each equivalent to a completely adaptable problem with a larger uncertainty set. This uncertainty set is smaller in the case of the k-adaptability problem, than in the static robust problem. In this section, we characterize how much smaller this effective uncertainty set must be, in order to guarantee a given level of improvement from the static robust value. We show that the points of the larger uncertainty set that must be eliminated to obtain a given improvement level, each correspond to necessary conditions that a partition must satisfy in order to guarantee improvement. Furthermore, collectively these necessary conditions turn out to be sufficient.

Thus in this section we use the geometric characterization of the previous section to essentially characterize the set of partitions that achieve a particular level of improvement over the static robust solution.

Lemma 3.2 says that $\mathrm{Static}(\mathcal{P}) = \mathrm{CompAdapt}((\mathcal{P})_R)$. Therefore, there must exist some $\hat{\boldsymbol{B}} \in (\mathcal{P})_R$ for which the nominal problem $\min : \{\boldsymbol{d}^\top \boldsymbol{y} \, : \, \hat{\boldsymbol{B}}\boldsymbol{y} \geq \boldsymbol{b}\}$ has value equal to the static robust optimal value of (3.3.8). Let $\mathcal{B}$ denote all such matrices. In fact, we show that for any $\eta > 0$, there exists a set $\mathcal{A}_\eta \subseteq (\mathcal{P})_R$ such that if $\boldsymbol{d}^\top \boldsymbol{y} < \mathrm{Static}(\mathcal{P}) - \eta$, then $\boldsymbol{y}$ does not satisfy $\boldsymbol{B}\boldsymbol{y} \geq \boldsymbol{b}$, for any $\boldsymbol{B} \in \mathcal{A}_\eta$. We show below that the sets $\mathcal{B}$ and $\mathcal{A}_\eta$ are the images under a computable map, of a polytope associated with the dual of the static robust problem. In Proposition 3.4 we show that these sets are related to whether a given partition can achieve η-improvement over the static robust value. In Proposition 3.6 we then show that each point of these sets maps to a necessary condition which any η-improving partition must satisfy.

Proposition 3.4

(a) *The sets $\mathcal{B}$ and $\mathcal{A}_\eta$ are the images under a computable map, of a polytope associated with the dual of the static robust problem.*

(b) *Adaptability with k contingency plans corresponding to the partition $\mathcal{P} = \mathcal{P}_1 \cup \cdots \cup \mathcal{P}_k$*

improves the cost by more than η if and only if

$$((\mathcal{P}_1)_R \cup \cdots \cup (\mathcal{P}_k)_R) \cap \bar{\mathcal{A}}_\eta = \emptyset.$$

Here, $\bar{\mathcal{A}}_\eta$ denotes the closure of the set $\mathcal{A}_\eta$.
(c) *There is some $k < \infty$ for which k optimally chosen contingency plans can improve the cost by at least η if and only if $\mathcal{P} \cap \bar{\mathcal{A}}_\eta = \emptyset$.*

For the proof, we first describe a polytope associated to the dual of the robust problem, and we give the map that yields the sets $\mathcal{B}$ and $\mathcal{A}_\eta$, proving **(a)**. Then we prove parts **(b)** and **(c)** of the proposition using the results of Lemma 3.5 below.

We consider the case where the uncertainty is given as the convex hull of a given set of extreme points: $\mathcal{P} = \text{conv}\{\boldsymbol{B}^1, \ldots, \boldsymbol{B}^K\}$. The robust optimization problem has the particularly convenient form,

$$\begin{array}{rl} \min: & \boldsymbol{d}^\top \boldsymbol{y} \\ \text{s.t.}: & \boldsymbol{B}^i \boldsymbol{y} \geq \boldsymbol{b}, \quad 1 \leq i \leq K. \end{array} \tag{3.3.10}$$

For any $\eta > 0$, we consider the infeasible problem

$$\begin{array}{rl} \min: & 0 \\ \text{s.t.}: & \boldsymbol{B}^i \boldsymbol{y} \geq \boldsymbol{b}, \quad 1 \leq i \leq K \\ & \boldsymbol{d}^\top \boldsymbol{y} \leq \text{Static}(\mathcal{P}) - \eta. \end{array} \tag{3.3.11}$$

The dual of (3.3.11) is feasible, and hence unbounded. Let $\mathcal{C}_\eta(\mathcal{P})$ be the closure of the set of directions of dual unboundedness of (3.3.11):

$$\mathcal{C}_\eta = \mathcal{C}_\eta(\mathcal{P}) \stackrel{\Delta}{=} \left\{ (\boldsymbol{p}_1, \ldots, \boldsymbol{p}_K) : \begin{array}{l} (\boldsymbol{p}_1 + \cdots + \boldsymbol{p}_K)^\top \boldsymbol{b} \geq \text{Static}(\mathcal{P}) - \eta \\ \boldsymbol{p}_1^\top \boldsymbol{B}^1 + \cdots + \boldsymbol{p}_K^\top \boldsymbol{B}^K = \boldsymbol{d} \\ \boldsymbol{p}_1, \ldots, \boldsymbol{p}_K \geq 0 \end{array} \right\}.$$

Note the dependence on the uncertainty set $\mathcal{P}$. We suppress this when the uncertainty set is clear from the context. $\mathcal{C}_0 = \mathcal{C}_0(\mathcal{P})$ is the set of dual optimal solutions to (3.3.10). For $(\boldsymbol{p}_1, \ldots, \boldsymbol{p}_K) \in \mathcal{C}_\eta$, let p_{ij} denote the j^{th} component of $\boldsymbol{p}_i$. Let $(\boldsymbol{B}^i)_j$ denote the j^{th} row of the matrix $\boldsymbol{B}^i$. Construct a matrix $\tilde{\boldsymbol{B}}$ whose j^{th} row is given by

$$(\tilde{\boldsymbol{B}})_j = \begin{cases} \boldsymbol{0}, & \text{if } \sum_i p_{ij} = 0. \\ \dfrac{p_{1j}(\boldsymbol{B}^1)_j + \cdots + p_{Kj}(\boldsymbol{B}^K)_j}{\sum_i p_{ij}}, & \text{otherwise.} \end{cases} \tag{3.3.12}$$

Therefore, each nonzero row of $\tilde{\boldsymbol{B}}$ is a convex combination of the corresponding rows of the $\boldsymbol{B}^i$ matrices. Let $\hat{\boldsymbol{B}}$ be any matrix in $(\mathcal{P})_R$ that coincides with $\tilde{\boldsymbol{B}}$ on all its nonzero rows.

Lemma 3.5 *For $\hat{\boldsymbol{B}}$ defined as above,*

$$\begin{bmatrix} \min: & \boldsymbol{d}^\top \boldsymbol{y} \\ \text{s.t.}: & \hat{\boldsymbol{B}}\boldsymbol{y} \geq \boldsymbol{b} \end{bmatrix} \geq \text{Static}(\mathcal{P}) - \eta. \tag{3.3.13}$$

If $\eta = 0$, and if $\boldsymbol{y}_R$ is an optimal solution for the robust problem (3.3.10), then $\boldsymbol{y}_R$ is also an optimal solution for the nominal problem with the matrix $\hat{\boldsymbol{B}}$.

PROOF. The proof follows by duality. We first consider the case $\eta = 0$. The dual to the nominal problem $\min : \{\boldsymbol{d}^\top \boldsymbol{y} \, : \, \hat{\boldsymbol{B}}\boldsymbol{y} \geq \boldsymbol{b}\}$ is given by $\max : \{\boldsymbol{q}^\top \boldsymbol{b} \, : \, \boldsymbol{q}^\top \hat{\boldsymbol{B}} = \boldsymbol{d}, \boldsymbol{q} \geq 0\}$. We construct a solution $\boldsymbol{q}$ to this dual, and show that its objective value is equal to $\boldsymbol{d}^\top \boldsymbol{y}_R$, thus implying $\boldsymbol{q}$ is optimal. For $(\boldsymbol{p}_1, \dots, \boldsymbol{p}_K) \in \mathcal{C}_0$, define the vector $\boldsymbol{q}$ by $q_j \triangleq p_{1j} + p_{2j} + \cdots + p_{Kj}$. The vector $\boldsymbol{q}$ is nonnegative, and in addition, for any $1 \leq r \leq n$, we also have:

$$\begin{aligned}
(\boldsymbol{q}^\top \hat{\boldsymbol{B}})_r &= \sum_{j=1}^{m} q_j \hat{B}_{jr} \\
&= \sum_{j=1}^{m} \left(\sum_i p_{ij} \right) \left(\frac{1}{\sum_i p_{ij}} \sum_i p_{ij} (\boldsymbol{B}^i)_{jr} \right) \\
&= \sum_{j=1}^{m} \sum_{i=1}^{K} p_{ij} (\boldsymbol{B}^i)_{jr} \\
&= \left(\boldsymbol{p}_1 \boldsymbol{B}^1 + \cdots + \boldsymbol{p}_K \boldsymbol{B}^K \right)_r \\
&= c_r.
\end{aligned}$$

Similarly,

$$\boldsymbol{q}^\top \boldsymbol{b} = (\boldsymbol{p}_1 + \cdots + \boldsymbol{p}_K)^\top \boldsymbol{b} = \boldsymbol{d}^\top \boldsymbol{y}_R.$$

Therefore, $\boldsymbol{q}$ as constructed is an optimal (and feasible) solution to the dual of (3.3.13), with objective value the same as the dual to the original robust problem (3.3.10). Since $\boldsymbol{y}_R$ is certainly feasible for problem (3.3.13), it must then also be optimal. A similar argument holds for $\eta > 0$. □

We can now prove Proposition 3.4.

PROOF. **(a)** The collection of such $\hat{\boldsymbol{B}}$ obtained as images of points in $\mathcal{C}_0$ and $\mathcal{C}_\eta$ respectively, under the map given in (3.3.12) make up the sets $\mathcal{B}$ and $\mathcal{A}_\eta$. Lemma 3.5 shows that these sets indeed have the required properties.

(b) The value of the k-adaptable solution corresponding to the partition $\mathcal{P}_1 \cup \cdots \cup \mathcal{P}_k$ is

$$\max_{1 \leq d \leq k} \{\text{Static}(\mathcal{P}_d)\}.$$

By Lemma 3.2, $\text{Static}(\mathcal{P}_d) = \text{Static}((\mathcal{P}_d)_R)$. If $((\mathcal{P}_1)_R \cup \cdots \cup (\mathcal{P}_k)_R) \cap \bar{\mathcal{A}}_\eta \neq \emptyset$, then we can find some $\hat{\boldsymbol{B}} \in (\mathcal{P}_d)_R \cap \bar{\mathcal{A}}_\eta$, for some $1 \leq d \leq k$, and also we can find matrices $\hat{\boldsymbol{B}}_l \in \mathcal{A}_\eta$ with $\hat{\boldsymbol{B}}_l \to \hat{\boldsymbol{B}}$. By Lemma 3.5, the nominal problem with matrix $\hat{\boldsymbol{B}}_l$ must have value at least Static($\mathcal{P}$) $- \eta$, for every l. The optimal value of a linear optimization problem is continuous in its parameters. Therefore, the value of the nominal problem with matrix $\hat{\boldsymbol{B}}$ must also be at least Static($\mathcal{P}$) $- \eta$. The value of Static($\mathcal{P}_d$) can be no more than the value of the nominal problem with matrix $\hat{\boldsymbol{B}}$, and hence Static($\mathcal{P}_d$) $\geq$ Static($\mathcal{P}$) $- \eta$, which means that the improvement cannot be greater than η.

Conversely, if the partition does not improve the value by more than η, then there must exist some $1 \leq d \leq k$ such that Static($\mathcal{P}_d$) $\geq$ Static($\mathcal{P}$) $- \eta$. This implies that $\mathcal{C}_\eta(\mathcal{P}_d)$ is non-empty. Any point of $\mathcal{C}_\eta(\mathcal{P}_d)$ then maps via (3.3.12) to some $\hat{\boldsymbol{B}} \in \mathcal{A}_\eta \cap (\mathcal{P}_d)_R$, and the intersection is non-empty, as required.

(c) If $\bar{\mathcal{A}}_\eta \cap \mathcal{P} \neq \emptyset$, then the point of intersection will always belong to some element of any partition, and hence no partition can satisfy the condition of part **(b)**. Conversely, if the intersection is empty, then since both $\mathcal{P}$ and $\bar{\mathcal{A}}_\eta$ are closed, and $\mathcal{P}$ is compact, the minimum distance

$$\inf_{\boldsymbol{B} \in \mathcal{P}, \hat{\boldsymbol{B}} \in \bar{\mathcal{A}}_\eta} d(\boldsymbol{B}, \hat{\boldsymbol{B}}),$$

is attained, and therefore is strictly positive. Then by Lemma 3.2 part **(c)**, there must exist some partition of $\mathcal{P}$ that satisfies the empty intersection property of condition **(b)** above. □

We now use the characterization of Proposition 3.4 to obtain necessary conditions that any η-improving partition must satisfy. To this end, let $\boldsymbol{\alpha}_j$ denote the convex-combination coefficients used to construct the j^{th} row of $\tilde{\boldsymbol{B}}$ above for all non-zero rows, so that

$$\boldsymbol{\alpha}_j = \frac{1}{\sum_i p_{ij}} \left(p_{1j}, p_{2j}, \dots, p_{Kj}\right).$$

Using these coefficients, we define matrices $\boldsymbol{Q}_1, \dots, \boldsymbol{Q}_m \in \mathcal{P}$ by

$$\boldsymbol{Q}_j = \sum_{i=1}^{K} (\boldsymbol{\alpha}_j)_i \boldsymbol{B}^i.$$

Consider now any partition of the uncertainty set, $\mathcal{P} = \mathcal{P}_1 \cup \cdots \cup \mathcal{P}_k$. If for some $1 \leq d \leq k$, we have $\{\boldsymbol{Q}_1, \dots, \boldsymbol{Q}_m\} \subset \mathcal{P}_d$, then $\hat{\boldsymbol{B}} \in (\mathcal{P}_d)_R$. Therefore, $(\mathcal{P}_d)_R \cap \mathcal{A}_\eta \neq \emptyset$, and thus by Proposition 3.4, the proposed partition cannot improve the static robust cost by more than η. Therefore, the set of matrices $\{\boldsymbol{Q}_1, \dots, \boldsymbol{Q}_m\}$ of $\mathcal{P}$ constitutes a *necessary condition* that any η-improving partition of $\mathcal{P}$ must satisfy: a partition of $\mathcal{P}$ can improve the solution more than η only if it splits the set $\{\boldsymbol{Q}_1, \dots, \boldsymbol{Q}_m\}$. Indeed, something more general is true.

Proposition 3.6
(a) *Consider any element $\tilde{\boldsymbol{B}}$ obtained from a point of $\mathcal{C}_\eta$, according to (3.3.12). Let us assume that the first r rows of the matrix $\tilde{\boldsymbol{B}}$ are nonzero. Let $\mathcal{Q}_i = \pi_i^{-1}(\tilde{\boldsymbol{B}}_i)$ denote the set of matrices in $\mathcal{P}$ whose i^{th} row equals the i^{th} row of $\tilde{\boldsymbol{B}}$, $1 \le i \le r$. Then a partition $\mathcal{P} = \mathcal{P}_1 \cup \cdots \cup \mathcal{P}_k$ can achieve an improvement of more than η only if for any region $\mathcal{P}_d$, $1 \le d \le k$, there exists some $1 \le i \le r$, such that*

$$\mathcal{P}_d \cap \mathcal{Q}_i = \emptyset.$$

(b) *Collectively, these necessary conditions are also sufficient.*

PROOF. **(a)** Suppose that there exists a region $\mathcal{P}_d$ of the partition, for which no such index i exists, and we have $\mathcal{P}_d \cap \mathcal{Q}_i \neq \emptyset$ for $1 \le i \le r$. Then we can find matrices $\boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_r$, such that $\boldsymbol{Q}_i \in \mathcal{P}_d \cap \mathcal{Q}_i$. By definition, the i^{th} row of matrix $\boldsymbol{Q}_i$ coincides with the i^{th} row of $\tilde{\boldsymbol{B}}$. Therefore, $\hat{\boldsymbol{B}} \in (\mathcal{P}_d)_R$. Now the proof of necessity follows from Proposition 3.4.
(b) Suppose that a partition $\mathcal{P} = \mathcal{P}_1 \cup \cdots \cup \mathcal{P}_k$ satisfies the full list of necessary conditions corresponding to all elements of $\tilde{\mathcal{B}}_\eta$, yet the corresponding value of $\text{Adapt}_k(\mathcal{P})$ does not achieve the guaranteed improvement, i.e., $\text{Adapt}_k(\mathcal{P}) = \text{Static}(\mathcal{P}) - \eta' \ge \text{Static}(\mathcal{P}) - \eta$, for some $\eta' \le \eta$. Then, by the structure of the finite adaptability problem there must be one region of the partition, say $\mathcal{P}_d$, such that $\text{Adapt}_k(\mathcal{P}) = \text{Static}(\mathcal{P}_d)$. Then $\mathcal{C}_\eta(\mathcal{P}_d)$ is non-empty. Given any point of $\mathcal{C}_\eta(\mathcal{P}_d)$, we can then construct $\hat{\boldsymbol{B}}$ and the corresponding unsatisfied necessary condition $\{\mathcal{Q}_1, \ldots, \mathcal{Q}_r\}$. Expressing the extreme points of $\mathcal{P}_d$ as a convex combination of extreme points of $\mathcal{P}$, this unsatisfied necessary condition corresponds to a point in $\mathcal{C}_\eta(\mathcal{P})$, a contradiction. □

Therefore, we can map any point of $\mathcal{C}_\eta$ to a necessary condition that any partition improving the solution of the static robust problem by at least η, must satisfy. In Section 3.5, we show that computing the optimal partition into two (equivalently, computing the best two contingency plans) is NP-hard. In Section 3.7, we provide an efficient, but possibly sub-optimal algorithm for the 2^k-partition problem. However, this algorithm does not offer any theoretical guarantee that more progress cannot be made with another choice of partition. While in general a particular partition must satisfy the full (infinite) set of necessary conditions to guarantee that it improves the static robust solution by at least η, a small list of necessary conditions may provide a short certificate that there does not exist a partition with $k \le k'$, that achieves η-improvement. In Section 3.6, we provide a simple example of this phenomenon. Indeed, in this example, a finite (and small) set of necessary conditions reveals the limits, and structure of 2,3,4,5-adaptability.

3.4 Exact Formulations

In this section we give exact and finite formulations of the optimal 2-adaptability problem. First, we consider the general matrix-uncertainty case, and we show that the infinite-constraint disjunctive optimization problem (3.2.5) can be formulated as a bilinear problem. Next, we specialize to the case of right hand side uncertainty. Here we show that we can formulate the problem as a $\{0,1\}$ linear integer optimization problem.

3.4.1 A Bilinear Formulation

Thus far we have considered a geometric point of view. Here we follow an algebraic development. In (3.2.5) we formulated the k-adaptability problem as an infinite-constraint disjunctive program:

$$\begin{array}{ll} \min: & \max\{\boldsymbol{d}^{\top}\boldsymbol{y}_1,\dots,\boldsymbol{d}^{\top}\boldsymbol{y}_k\} \\ \text{s.t.}: & [\boldsymbol{B}\boldsymbol{y}_1 \geq \boldsymbol{b} \text{ or } \boldsymbol{B}\boldsymbol{y}_2 \geq \boldsymbol{b} \text{ or } \cdots \text{ or } \boldsymbol{B}\boldsymbol{y}_k \geq \boldsymbol{b}] \quad \forall \boldsymbol{B} \in \mathcal{P}. \end{array} \tag{3.4.14}$$

We reformulate this problem as a (finite) bilinear optimization problem. In general, bilinear problems are hard to solve but much work has been done algorithmically (see [65],[122],[123] and references therein) toward their solution. For notational convenience, we consider the case $k=2$, but the extension to the general case is straightforward. Also, for this section as well, we focus on the case where the uncertainty set $\mathcal{P}$ is given as a convex hull of its extreme points: $\mathcal{P} = \text{conv}\{\boldsymbol{B}^1,\dots,\boldsymbol{B}^K\}$.

Proposition 3.7
The optimal 2-adaptability value, and the optimal two contingency plans, are given by the solution to the following bilinear optimization:

$$\begin{array}{ll} \min: & \max\{\boldsymbol{d}^{\top}\boldsymbol{y}_1,\boldsymbol{d}^{\top}\boldsymbol{y}_2\} \\ \text{s.t.}: & \mu_{ij}\left[(\boldsymbol{B}^l\boldsymbol{y}_1)_i - b_i\right] + (1-\mu_{ij})\left[(\boldsymbol{B}^l\boldsymbol{y}_2)_j - b_j\right] \geq 0, \quad \forall 1 \leq i,j \leq m, \quad \forall 1 \leq l \leq K \\ & 0 \leq \mu_{ij} \leq 1, \quad \forall 1 \leq i,j \leq m. \end{array} \tag{3.4.15}$$

Recall that m is the number of rows of $\boldsymbol{B}$. We can interpret the variables μ_{ij} essentially as a mixing of the constraints. For any $\{\mu_{ij}\}$, the pair $(\boldsymbol{y}_1,\boldsymbol{y}_2) = (\boldsymbol{y}_R,\boldsymbol{y}_R)$ is feasible. Indeed, fixing $\mu_{ij}=1$ for all (i,j) leaves $\boldsymbol{y}_2$ unrestricted, and the resulting constraints on $\boldsymbol{y}_1$ recover the original static robust problem. Thus, the problem is to find the optimal mixing weights.

PROOF. We show that a pair $(\boldsymbol{y}_1,\boldsymbol{y}_2)$ is a feasible solution to problem (3.4.14) if and only if there exist weights $\mu_{ij} \in [0,1]$, $1 \leq i,j \leq m$, such that

$$\mu_{ij}\left[(\boldsymbol{B}^l\boldsymbol{y}_1)_i - b_i\right] + (1-\mu_{ij})\left[(\boldsymbol{B}^l\boldsymbol{y}_2)_j - b_j\right] \geq 0, \quad \forall 1 \leq i,j \leq m, \quad \forall 1 \leq l \leq K.$$

First consider the "if" direction. Suppose that the pair $(\boldsymbol{y}_1, \boldsymbol{y}_2)$ is not a feasible solution to problem (3.4.14). Then there exists $1 \leq i,j \leq m$ and $\boldsymbol{B} \in \mathcal{P}$ such that

$$(\boldsymbol{B}\boldsymbol{y}_1)_i - b_i < 0 \quad \text{and} \quad (\boldsymbol{B}\boldsymbol{y}_2)_j - b_j < 0.$$

Since $\boldsymbol{B} \in \mathcal{P}$, we must have $\boldsymbol{B} = \sum_{l=1}^{K} \lambda_l \boldsymbol{B}^l$, for a convex combination given by $\boldsymbol{\lambda}$. For any $\mu_{ij} \in [0,1]$ we have:

$$\begin{aligned} &\mu_{ij}\left[\sum_l \lambda_l(\boldsymbol{B}^l\boldsymbol{y}_1)_i - b_i\right] + (1-\mu_{ij})\left[\sum_l \lambda_l(\boldsymbol{B}^l\boldsymbol{y}_2)_j - b_j\right] < 0 \\ \Rightarrow \quad &\sum_l \lambda_l \left[\mu_{ij}\left\{(\boldsymbol{B}^l\boldsymbol{y}_1)_i - b_i\right\} + (1-\mu_{ij})\left\{(\boldsymbol{B}^l\boldsymbol{y}_2)_j - b_j\right\}\right] < 0. \end{aligned}$$

This follows since $\sum_l \lambda_l = 1$. But then there must be some index l^* for which the corresponding term in the sum is negative, i.e.,

$$\mu_{ij}\left\{(\boldsymbol{B}^{l^*}\boldsymbol{y}_1)_i - b_i\right\} + (1-\mu_{ij})\left\{(\boldsymbol{B}^{l^*}\boldsymbol{y}_2)_j - b_j\right\} < 0.$$

For the converse, let $(\boldsymbol{y}_1, \boldsymbol{y}_2)$ be a feasible solution to problem (3.4.14). We show there exist weights $\mu_{ij} \in [0,1]$ satisfying the required inequalities. By assumption, for any $\boldsymbol{B} \in \mathcal{P}$, either $[\boldsymbol{B}\boldsymbol{y}_1 \geq \boldsymbol{b}]$, or $[\boldsymbol{B}\boldsymbol{y}_2 \geq \boldsymbol{b}]$. In particular, for any $1 \leq i,j \leq m$, the value of the following optimization over $\mathcal{P}$ is finite and non-positive (recall that $\boldsymbol{y}_1, \boldsymbol{y}_2$ are fixed).

$$\begin{array}{rl} \max: & \varepsilon \\ \text{s.t.}: & (\boldsymbol{B}\boldsymbol{y}_1)_i + \varepsilon \leq b_i \\ & (\boldsymbol{B}\boldsymbol{y}_2)_j + \varepsilon \leq b_j \\ & \boldsymbol{B} \in \mathcal{P}. \end{array}$$

Writing $\mathcal{P} = \{\sum_l \lambda_l \boldsymbol{B}^l \ : \ \sum_l \lambda_l = 1, \lambda_l \geq 0\}$, and taking the dual using dual variables μ, ν for the two inequality constraints, and w for the normalization constraint in $\boldsymbol{\lambda}$, we have:

$$\begin{array}{rl} \min: & \mu b_i + \nu b_j - w \\ \text{s.t.}: & \mu(\boldsymbol{B}^l\boldsymbol{y}_1)_i + \nu(\boldsymbol{B}^l\boldsymbol{y}_2)_j - w \geq 0, \quad \forall 1 \leq l \leq K \\ & \mu + \nu = 1 \\ & \mu, \nu \geq 0. \end{array}$$

By strong duality, this problem is feasible, and its optimal value is non-positive. In particular, the following system is feasible:

$$\left\{\begin{array}{l} \mu b_i + \nu b_j \leq w \\ \mu(\boldsymbol{B}^l\boldsymbol{y}_1)_i + \nu(\boldsymbol{B}^l\boldsymbol{y}_2)_j \geq w \\ \mu + \nu = 1 \\ \mu, \nu \geq 0 \end{array}\right\}$$

and therefore there exists $\mu \in [0,1]$, and w such that

$$\mu b_i + (1-\mu)b_j \leq w \leq \mu(\boldsymbol{B}^l\boldsymbol{y}_1)_i + \nu(\boldsymbol{B}^l\boldsymbol{y}_2)_j, \quad \forall 1 \leq l \leq K.$$

Grouping the terms on one side of the inequality, we have that there exists a weight $\mu \in [0,1]$ such that

$$\mu\left[(\boldsymbol{B}^l\boldsymbol{y}_1)_i - b_i\right] + (1-\mu)\left[(\boldsymbol{B}^l\boldsymbol{y}_2)_j - b_j\right] \geq 0, \quad \forall 1 \leq l \leq K.$$

□

■ 3.4.2 Right Hand Side Uncertainty

While matrix uncertainty subsumes right hand side (RHS) uncertainty, we now focus exclusively on RHS-uncertainty. In many cases of practical interest, the uncertainty is indeed in the right hand side; for instance, demand uncertainty, capacity uncertainty, or supply uncertainty, are all modeled through RHS-robustness. We show that through a reformulation of the problem, finding the optimal two contingency plans, or equivalently, finding the best partition of the uncertainty set into two, can be cast as a discrete optimization problem. For the case where $\mathcal{P}$ has the structure of a simplex, we obtain a mixed integer LP with $\{0,1\}$-variables. The static robust problem and linear optimization formulation are given by:

$$\left[\begin{array}{lll} \min: & \boldsymbol{d}^\top\boldsymbol{y} & \\ \text{s.t.}: & \boldsymbol{B}\boldsymbol{y} \geq \boldsymbol{b}, & \forall \boldsymbol{b} \in \mathcal{P} \\ & \boldsymbol{F}\boldsymbol{y} \geq \boldsymbol{f} & \end{array}\right] = \left[\begin{array}{ll} \min: & \boldsymbol{d}^\top\boldsymbol{y} \\ \text{s.t.}: & \boldsymbol{B}\boldsymbol{y} \geq \boldsymbol{b}_R \\ & \boldsymbol{F}\boldsymbol{y} \geq \boldsymbol{f} \end{array}\right]$$

where $\boldsymbol{b}_R$ is the point whose i^{th} coordinate is the maximization over $\mathcal{P}$ in the i^{th} coordinate direction. We can associate the smallest hypercube containing $\mathcal{P}$ (as we defined in (3.3.9)) with the single point $\boldsymbol{b}_R$: $(\mathcal{P})_R = \{\boldsymbol{b} : \boldsymbol{b} \leq \boldsymbol{b}_R\}$ as lower bounds do not affect the optimization. As with the general static robust problem (3.3.8) 2-adaptability improves the static robust cost by considering pairs of hypercubes containing $\mathcal{P}$, rather than just a single hypercube. We formalize this in the following.

Definition 3.1
We say that a pair of points $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ **covers** *the set $\mathcal{P}$ if for any $\boldsymbol{b} \in \mathcal{P}$ we have $\boldsymbol{b} \leq \boldsymbol{b}_1$ or $\boldsymbol{b} \leq \boldsymbol{b}_2$. We denote by $\mathcal{C}(\mathcal{P})$ the set of all pairs $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ that cover the set $\mathcal{P}$:*

$$\mathcal{C}(\mathcal{P}) \stackrel{\triangle}{=} \{(\boldsymbol{b}_1, \boldsymbol{b}_2) \ : \ \forall \boldsymbol{b} \in \mathcal{P} \text{ we have } \boldsymbol{b} \leq \boldsymbol{b}_1, \text{ or } \boldsymbol{b} \leq \boldsymbol{b}_2\}.$$

Next, we define the problem:

$$\begin{aligned} z_2(\boldsymbol{b}_1, \boldsymbol{b}_2) \ = \ \min : \ & \max\{\boldsymbol{d}^\top \boldsymbol{y}_1, \boldsymbol{d}^\top \boldsymbol{y}_2\} \\ \text{s.t.} : \ & \boldsymbol{B}\boldsymbol{y}_1 \geq \boldsymbol{b}_1 \\ & \boldsymbol{B}\boldsymbol{y}_2 \geq \boldsymbol{b}_2 \\ & \boldsymbol{D}\boldsymbol{y}_l \geq \boldsymbol{d}, \quad l = 1, 2. \end{aligned}$$

Thus, we can rewrite the problem of obtaining the best split of the uncertainty set, as the optimization of the convex function $z_2(\boldsymbol{b}_1, \boldsymbol{b}_2)$ over the set $\mathcal{C}(\mathcal{P})$:

$$\begin{aligned} \min : \ & z_2(\boldsymbol{b}_1, \boldsymbol{b}_2) \\ \text{s.t.} : \ & (\boldsymbol{b}_1, \boldsymbol{b}_2) \in \mathcal{C}(\mathcal{P}). \end{aligned} \tag{3.4.16}$$

Let $(\boldsymbol{b}_1, \boldsymbol{b}_2) \in \mathcal{C}(\mathcal{P})$ be any pair that covers $\mathcal{P}$, and let $\boldsymbol{b}_R$ be as above. For any $\boldsymbol{b} \in \mathcal{P}$, by definition, we must have $\boldsymbol{b} \leq \boldsymbol{b}_1$ or $\boldsymbol{b} \leq \boldsymbol{b}_2$. Therefore, for any coordinate index i, we must have $(\boldsymbol{b}_1)_i \geq (\boldsymbol{b}_R)_i$, or $(\boldsymbol{b}_2)_i \geq (\boldsymbol{b}_R)_i$. Therefore, the pair $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ induces a separation of the coordinates, $S = \{i \ : \ (\boldsymbol{b}_1)_i \geq (\boldsymbol{b}_R)_i\}$. Given a partition (S, S^c) of the coordinates, i.e., $S \subseteq \{1, \ldots, m\}$, there are elements of $\mathcal{C}(\mathcal{P})$ that induce this same partition. We define this set, and denote it by $\mathcal{C}(\mathcal{P}, S)$:

$$\mathcal{C}(\mathcal{P}, S) \stackrel{\triangle}{=} \{(\boldsymbol{b}_1, \boldsymbol{b}_2) \in \mathcal{C}(\mathcal{P}) \ : \ (\boldsymbol{b}_1)_i = (\boldsymbol{b}_R)_i, \forall i \in S, \quad (\boldsymbol{b}_2)_j = (\boldsymbol{b}_R)_j, \forall j \in S^c\}.$$

The union of the sets $\mathcal{C}(\mathcal{P}, S)$ for all S, gives us back the subset of $\mathcal{C}(\mathcal{P})$ that always contains the optimal solution. We can characterize the sets $\mathcal{C}(\mathcal{P}, S)$ using a linear optimization approach.

Lemma 3.8 *For any fixed set $S \subseteq \{1, \ldots, n\}$, the set $\mathcal{C}(\mathcal{P}, S)$ is given as the set of all pairs $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ with*

$$(\boldsymbol{b}_1)_i \ = \ \begin{cases} (\boldsymbol{b}_R)_i, & \text{if } i \in S, \\ \mu_i, & \text{otherwise,} \end{cases} \qquad \text{and} \qquad (\boldsymbol{b}_2)_j \ = \ \begin{cases} (\boldsymbol{b}_R)_j, & \text{if } j \in S^c, \\ \lambda_j, & \text{otherwise.} \end{cases}$$

The values $\{\mu_i\}$ must satisfy $0 \leq \mu_i \leq (\boldsymbol{b}_R)_i$. The values $\{\lambda_j\}$ must be such that:

$$\lambda_i \geq \max_{j \in S^c} \begin{bmatrix} \max : & \boldsymbol{e}_i^\top \boldsymbol{b} \\ \text{s.t.} : & (\boldsymbol{b})_j \geq \mu_j \\ & \boldsymbol{b} \in \mathcal{P} \end{bmatrix}.$$

PROOF. Fix values for $\boldsymbol{b}_1$. We must take $(\boldsymbol{b}_2)_j = (\boldsymbol{b}_R)_j$ for every $j \in S^c$. If the lemma does not hold, there must exist some $i \in S$ and $j \in S^c$, as well as a $\boldsymbol{b} \in \mathcal{P}$, $\boldsymbol{b}(j) \geq \mu_j$ (with $\mu_j < (\boldsymbol{b}_R)_j$), and for which $\lambda_i < \boldsymbol{b}_i$. Now take $\tilde{\boldsymbol{b}} \in \mathcal{P}$ with $(\tilde{\boldsymbol{b}})_j = (\boldsymbol{b}_R)_j > \mu_j$. Taking $\delta > 0$ small enough, we see that for $\hat{\boldsymbol{b}} = (1-\delta)\boldsymbol{b} + \delta\tilde{\boldsymbol{b}}$, we have $\hat{\boldsymbol{b}} \in \mathcal{P}$ by convexity of $\mathcal{P}$, but $\boldsymbol{b} \not\leq \boldsymbol{b}_1$, and $\boldsymbol{b} \not\leq \boldsymbol{b}_2$. Therefore, the pair $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ cannot cover $\mathcal{P}$, and the lemma is proved. □

In an important special case, the sets $C(\mathcal{P}, S)$ are polyhedral.

Proposition 3.9

Let $\mathcal{P}$ be a generalized simplex, that is, let $\mathcal{P}$ be the convex hull of scaled unit vectors:

$$\mathcal{P} = \operatorname{conv}\{\zeta_1 \boldsymbol{e}_1, \zeta_2 \boldsymbol{e}_2, \dots, \zeta_m \boldsymbol{e}_m\},$$

where $\boldsymbol{e}_i$ is the standard i^{th} unit basis vector, and the ζ_i are positive scalars. Then, for any $S \subseteq \{1, \dots, m\}$ the set $C(\mathcal{P}, S)$ is convex and polyhedral, given by:

$$C(\mathcal{P}, S) = \left\{ (\boldsymbol{b}_1, \boldsymbol{b}_2) : \begin{array}{ll} (\boldsymbol{b}_1)_i = (\boldsymbol{b}_R)_i = \zeta_i & \forall i \in S \\ (\boldsymbol{b}_2)_j = (\boldsymbol{b}_R)_j = \zeta_j & \forall j \in S^c \\ (\boldsymbol{b}_2)_i \geq \zeta_i - \left(\frac{\zeta_i}{\zeta_j}\right)(\boldsymbol{b}_1)_j & \forall i \in S,\ j \in S^c \end{array} \right\}.$$

PROOF. The proof follows Lemma 3.8. Take $S = \{1, \dots, k\}$. For any $i \in S$,

$$(\boldsymbol{b}_2)_i \geq \max_{j \in S^c} \left[\begin{array}{rl} \max: & \boldsymbol{e}_i^\top \boldsymbol{b} \\ \text{s.t.}: & (\boldsymbol{b})_j \geq (\boldsymbol{b}_1)_j \\ & \boldsymbol{b} \in \mathcal{P} \end{array} \right].$$

If $(\boldsymbol{b})_j \geq (\boldsymbol{b}_1)_j$, then the inner maximum is $\zeta_i - \left(\frac{\zeta_i}{\zeta_j}\right)(\boldsymbol{b}_1)_j$, attained at the point

$$\left(\underbrace{\overbrace{0, \dots, 0, \zeta_i - \left(\frac{\zeta_i}{\zeta_j}\right)(\boldsymbol{b}_1)_j}^{i}, 0, \dots, 0}_{k}, \underbrace{\overbrace{0, \dots, 0, (\boldsymbol{b}_1)_j}^{j-k}, 0, \dots, 0}_{m-k} \right) \in \mathcal{P}.$$

□

In particular, this tells us that if we can compute the optimal vertex partition S^*, then computing the optimal pair $(\boldsymbol{b}_1, \boldsymbol{b}_2) \in C(\mathcal{P}, S^*)$ amounts to minimizing a linear program. The set S^* is related to the necessary conditions obtained in Section 3.3.2, as it expresses a level of knowledge about which extreme points of the uncertainty set must be in separate regions of any good partition. From Proposition 3.9, we have the following.

Proposition 3.10

In the case where $\mathcal{P}$ is a (generalized) simplex, the optimal split and corresponding contingency plans, may be computed as the solution to the following $\{0,1\}$ mixed integer linear program.

$$\begin{array}{rll} \min: & \max\{\boldsymbol{d}^\top \boldsymbol{y}_1, \boldsymbol{d}^\top \boldsymbol{y}_2\} & \\ \text{s.t.}: & \boldsymbol{B}\boldsymbol{y}_1 \geq \boldsymbol{b}_1 & \\ & \boldsymbol{B}\boldsymbol{y}_2 \geq \boldsymbol{b}_2 & \\ & \boldsymbol{F}\boldsymbol{y}_l \geq \boldsymbol{f} & l = 1,2 \\ & (\boldsymbol{b}_1)_i \geq ((\boldsymbol{b}_R)_i) z_i & \forall i \\ & (\boldsymbol{b}_2)_j \geq ((\boldsymbol{b}_R)_j)(1 - z_j) & \forall j \\ & (\boldsymbol{b}_2)_i + \left(\frac{(\boldsymbol{b}_R)_i}{(\boldsymbol{b}_R)_j}\right)(\boldsymbol{b}_1)_j \geq (\boldsymbol{b}_R)_i & \forall i,j \\ & 0 \leq (\boldsymbol{b}_1)_i, (\boldsymbol{b}_2)_i \leq (\boldsymbol{b}_R)_i & \forall i \\ & z_i \in \{0,1\} & \forall i. \end{array} \tag{3.4.17}$$

■ 3.5 Complexity

In this section, we consider the complexity of k-adaptability. We show that even in the restricted case of right hand side uncertainty, in fact even in the special case where $\mathcal{P}$ has the form of a generalized simplex, computing the optimal partition of $\mathcal{P}$ into two sets, $\mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2$, is NP-hard. In particular, using the terminology of Section 3.4.2, we show that computing the optimal set S^* is hard. However, the NP-hardness is in the minimum of the dimension of the uncertainty, the dimension of the problem, and the number of constraints affected. After we establish NP-hardness for the general case, we show that if any of the three quantities: dimension of the uncertainty, dimension of the problem, number of uncertain constraints, is fixed, then computing the optimal 2-adaptability is theoretically tractable.

Proposition 3.11

Obtaining the optimal split $\mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2$ is in general NP-hard.

In particular, computing 2-adaptability is NP-hard. We obtain our hardness result using a reduction from PARTITION, which is NP-complete ([69],[85]). We show that if we can find the optimal split of an uncertainty set, then we can solve any PARTITION problem.

PROOF. The data for the PARTITION problem are the positive numbers $v_1, \ldots, v_m$. The problem is to minimize $|\sum_{i \in S} v_i - \sum_{j \in S^c} v_j|$ over subsets S. Given any such collection of numbers, consider the polytope $\mathcal{P} = \text{conv}\{e_1 v_1, \ldots, e_m v_m\}$, where the e_i form the standard basis for $\mathbb{R}^m$. Thus, $\mathcal{P}$ is the simplex in $\mathbb{R}^m$, but with general intercepts v_i. Consider the static robust optimization problem:

$$\begin{array}{rl} \min: & \sum_i y_i \\ \text{s.t.}: & I\boldsymbol{y} \geq \boldsymbol{b}, \quad \forall \boldsymbol{b} \in \mathcal{P}. \end{array} \tag{3.5.18}$$

Suppose the optimal partition is $\mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2$. Let $\boldsymbol{b}_1$ be the (componentwise) smallest point that covers $\mathcal{P}_1$, and $\boldsymbol{b}_2$ the smallest point covering $\mathcal{P}_2$. Then the pair $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ covers all of $\mathcal{P}$, i.e., $(\boldsymbol{b}_1, \boldsymbol{b}_2) \in \mathcal{C}(\mathcal{P})$, and from the decomposition $\mathcal{C}(\mathcal{P}) = \bigcup_S \mathcal{C}(\mathcal{P}, S)$, we must have $(\boldsymbol{b}_1, \boldsymbol{b}_2) \in \mathcal{C}(\mathcal{P}, S)$ for some set S. Without loss of generality, we assume that $S = \{1, \ldots, k\}$, so that $(\boldsymbol{b}_1)_i = (\boldsymbol{b}_R)_i = v_i$ for $1 \leq i \leq k$, and $(\boldsymbol{b}_2)_j = (\boldsymbol{b}_R)_j = v_j$ for $k+1 \leq j \leq m$. Thus, we can write the two points as

$$\begin{aligned} \boldsymbol{b}_1 &= (v_1, v_2, \ldots, v_k, \lambda_{k+1}, \ldots, \lambda_m) \\ \boldsymbol{b}_2 &= (\mu_1, \mu_2, \ldots, \mu_k, v_{k+1}, \ldots, v_m), \end{aligned}$$

where $0 \leq \lambda_j \leq v_j$ for $k+1 \leq j \leq m$. In this case, by Lemma 3.8 we must have

$$\begin{aligned} \mu_1 &\geq \max\left\{ v_1 - \left(\frac{v_1}{v_{k+1}}\right)\lambda_{k+1}, v_1 - \left(\frac{v_1}{v_{k+2}}\right)\lambda_{k+2}, \ldots, v_1 - \left(\frac{v_1}{v_m}\right)\lambda_m \right\} \\ &\vdots \\ \mu_k &\geq \max\left\{ v_k - \left(\frac{v_k}{v_{k+1}}\right)\lambda_{k+1}, v_k - \left(\frac{v_k}{v_{k+2}}\right)\lambda_{k+2}, \ldots, v_k - \left(\frac{v_k}{v_m}\right)\lambda_m \right\}. \end{aligned}$$

Since we claim that the pair $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ corresponds to the optimal partition of $\mathcal{P}$, we can take the inequalities above to be satisfied by equality, i.e., we take the μ_i to be as small as possible. Therefore, once the $\{\lambda_j\}$ are fixed, so are the $\{\mu_i\}$, and the pair $(\boldsymbol{b}_1, \boldsymbol{b}_2)$ is determined.

Now we compute the value of the free parameters $(\lambda_{k+1}, \ldots, \lambda_m)$ that determine the pair $(\boldsymbol{b}_1, \boldsymbol{b}_2)$. For the specific form of the optimization problem we consider, given a split $\mathcal{P}_1 \cup \mathcal{P}_2$ where $\mathcal{P}_1$ is covered by $\boldsymbol{b}_1$ and $\mathcal{P}_2$ by $\boldsymbol{b}_2$, the optimization takes the simple form:

$$\left\{ \begin{array}{ll} \text{min}: & \max\left\{ \left(\sum_i y_i^{(1)}\right), \left(\sum_i y_i^{(2)}\right) \right\} \\ \text{s.t.}: & \boldsymbol{y}^{(1)} \geq \boldsymbol{b}_1 \\ & \boldsymbol{y}^{(2)} \geq \boldsymbol{b}_2 \end{array} \right\} = \max\left\{ \left(\sum_i (\boldsymbol{b}_1)_i\right), \left(\sum_i (\boldsymbol{b}_2)_i\right) \right\}.$$

Therefore, if the partition is optimal, we must have $(\sum_i (\boldsymbol{b}_1)_i) = (\sum_i (\boldsymbol{b}_2)_i)$. Thus, we have

$$v_1 + \cdots + v_k + \lambda_{k+1} + \cdots + \lambda_m = \mu_1 + \cdots + \mu_k + v_{k+1} + \cdots + v_m. \qquad (3.5.19)$$

We have $(m-k)$ parameters that are not specified. The maximizations above that determine the μ_j give $(m-k-1)$ equations. Then Eq. (3.5.19) gives the final equation to determine our parameters uniquely. From the maximizations defining $\{\mu_j\}$, we have

$$v_j - \left(\frac{v_j}{v_{k+i}}\right)\lambda_{k+i} = v_j - \left(\frac{v_j}{v_{k+i'}}\right)\lambda_{k+i'}, \qquad 1 \leq j \leq k, \quad 1 \leq i, i' \leq m-k.$$

Solving in terms of λ_m, the above equations yield $\lambda_{k+i} = \left(\frac{v_{k+i}}{v_m}\right)\lambda_m, \quad 1 \leq i \leq m-k-1$. Substituting this back into Eq. (3.5.19), we obtain an equation in the single variable λ_m:

$$v_1+\cdots+v_k+\lambda_m\left(\frac{v_{k+1}+\cdots+v_m}{v_m}\right) = \left(v_1 - v_1\frac{\lambda_m}{v_m}\right)+\cdots+\left(v_k - v_k\frac{\lambda_m}{v_m}\right)+(v_{k+1}+\cdots+v_m),$$

which gives:

$$\lambda_m\left(\frac{v_1+\cdots+v_m}{v_m}\right) = (v_{k+1}+\cdots+v_m) \Longrightarrow \lambda_{k+i} = \frac{v_i(v_{k+1}+\cdots+v_m)}{v_1+\cdots+v_m}, \quad 1 \leq i \leq m-k.$$

Using these values of $\{\lambda_{k+i}\}$, we find that the optimal value $(*)$ of the optimization is given by

$$\begin{aligned}
(*) &= v_1+\cdots+v_k+\lambda_{k+1}+\cdots+\lambda_m \\
&= \frac{(v_1+\cdots+v_k)(v_1+\cdots+v_m)+(v_{k+1}+\cdots+v_m)(v_{k+1}+\cdots+v_m)}{(v_1+\cdots+v_m)} \\
&= \frac{((v_1+\cdots v_k)+(v_{k+1}+\cdots+v_m))^2-(v_1+\cdots+v_k)(v_{k+1}+\cdots+v_m)}{(v_1+\cdots+v_m)}.
\end{aligned}$$

The first term in the numerator, and also the denominator, are invariant under choice of partition. Thus, if this is indeed the optimal solution to the optimization (3.5.18), as we assume, then the second term in the numerator must be maximized. Thus, we see that minimizing (3.5.18) is equivalent to maximizing the product $\left(\sum_{i\in S} v_i\right)\left(\sum_{j\in S^c} v_j\right)$ over $S \subseteq \{1,\ldots,m\}$. This is equivalent to the PARTITION problem. □

Note that in this example, the dimension of the uncertainty, the dimension of the problem, and the number of constraints affected by the uncertainty are all equal. Next we show that if any one of these three quantities is fixed, then computing the optimal 2-adaptability is theoretically tractable.

Proposition 3.12
We consider the static robust problem

$$\begin{aligned}
\min: \quad & \boldsymbol{d}^\top \boldsymbol{y} \\
\text{s.t.}: \quad & \boldsymbol{B}\boldsymbol{y} \geq \boldsymbol{b}, \quad \forall \boldsymbol{B} \in \mathcal{P} \\
& \boldsymbol{F}\boldsymbol{y} \geq \boldsymbol{f}.
\end{aligned}$$

Let $\mathcal{P} = \text{conv}\{\boldsymbol{B}_1, \ldots, \boldsymbol{B}_N\}$ be an uncertainty set that allows for efficient solution of the robustified linear optimization problem (note that N need not necessarily be small). Let $d = \min\{N, \dim(\mathcal{P})\}$ be the real dimension of $\mathcal{P}$, let n denote the number of optimization variables, and let m be the number of rows of $\boldsymbol{B}$, i.e., the number of uncertain constraints. Define $\kappa = \min\{d, n, m\}$. Then, we can compute the ε-optimal 2-adaptability generated by a hyperplane partition, in time $O\left(poly(d, n, m, 1/\varepsilon)e^{\kappa}\right)$. In particular, if κ is constant, then hyperplane generated 2-adaptability can be computed efficiently.

PROOF. There are three possibilities: κ is defined by d, n, or m. In the case where d or n are fixed, then the result follows immediately, since we can find the best partition, or the best two solutions $\{\boldsymbol{y}_1, \boldsymbol{y}_2\}$ by brute force discretization of the uncertainty set, or the feasible set, respectively. Indeed, the only interesting case is when d and n are possibly large, but m is a constant. In this case, however, we have the result of Proposition 3.7, which says:

$$\begin{array}{ll}\text{min}: & \max\{\boldsymbol{d}^{\top}\boldsymbol{y}_1, \boldsymbol{d}^{\top}\boldsymbol{y}_2\} \\ \text{s.t.}: & \mu_{ij}\left[(\boldsymbol{B}\boldsymbol{y}_1)_i - b_i\right] + (1-\mu_{ij})\left[(\boldsymbol{B}\boldsymbol{y}_2)_j - b_j\right] \geq 0, \quad \forall 1 \leq i, j \leq m, \quad \forall \boldsymbol{B} \in \mathcal{P} \\ & 0 \leq \mu_{ij} \leq 1, \quad \forall 1 \leq i, j \leq m.\end{array}$$

For any fixed values of $\{\mu_{ij}\}$, the resulting problem is a static robust problem with uncertainty set $\mathcal{P}$, and hence by our assumption, it can be solved efficiently. Now if m is small, we discretize the possible set of $\{\mu_{ij}\}$, and search over this set by brute force. This completes the proof. □

While in principle this result says that for κ small the problem is tractable, in large scale applications we require more than theoretical tractability. We describe one such example in Section 3.8.1. In Section 3.7, we seek to give tractable algorithms that will be practically implementable in applications.

■ 3.6 An Extended Example

In this section we consider a detailed example. Through this example, we aim to illustrate several points and aspects of the theory developed in Section 3.3 above:

1. Propositions 3.4 and 3.6 tell us how to map $\mathcal{C}_\eta$ to $\mathcal{A}_\eta$ and then to obtain necessary conditions for η-improvement. Here we illustrate this process.

2. A small set of necessary conditions (obtained as in Propositions 3.4 and 3.6) may reveal the limits of k-adaptability for some k.

3. While in general not sufficient to guarantee η-improvement, a small set of necessary conditions may even suffice to reveal the optimal structure of k-adaptability for some k.

4. Finite adaptability may improve the solution considerably, even when affine adaptability fails, i.e., even when affine adaptability is no better than the static robust solution.

5. The Continuity Assumption may not be removed from Proposition 3.3. Without it, (uncountably) infinite adaptability may be required for even arbitrarily small improvement from the static robust solution.

6. The closure of the sets $\mathcal{B}$ and $\mathcal{A}_\eta$ in Proposition 3.4 cannot be relaxed.

We consider an example with one-dimensional uncertainty set.

$$\begin{array}{ll} \min: & y_1 + y_2 + y_3 \\ \text{s.t.}: & \boldsymbol{B}\boldsymbol{y} \geq \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \qquad \forall\, \boldsymbol{B} \in \operatorname{conv}\{\boldsymbol{B}^1, \boldsymbol{B}^2\} = \operatorname{conv}\left\{ \begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 1 \\ \frac{1}{2} & \frac{1}{5} & 0 \end{pmatrix}, \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ \frac{1}{5} & \frac{1}{2} & 0 \end{pmatrix} \right\} \\ & y_1, y_2, y_3 \geq 0. \end{array} \tag{3.6.20}$$

The unique optimal solution is $\boldsymbol{y}_R = (10/7, 10/7, 1)$, with corresponding value $\text{Static}(\mathcal{P}) = 27/7$. The completely adaptable value is $\text{CompAdapt}(\mathcal{P}) = 3$. The dual to the robust problem (3.6.20) is

$$\begin{array}{ll} \max: & (\boldsymbol{p}+\boldsymbol{q})^\top \boldsymbol{b} \\ \text{s.t.}: & \boldsymbol{p}^\top \boldsymbol{B}^1 + \boldsymbol{q}^\top \boldsymbol{B}^2 \leq \boldsymbol{d} \\ & \boldsymbol{p}, \boldsymbol{q} \geq \boldsymbol{0}. \end{array}$$

There are two extreme dual optimal solutions: $\boldsymbol{p}, \boldsymbol{q} = (0, 0, 10/7), (0, 1, 10/7)$, and $\boldsymbol{p}, \boldsymbol{q} = (1, 0, 10/7)$, $(0, 0, 10/7)$. We illustrate point (1) above by mapping these two points to the corresponding necessary conditions. Each of these maps to a unique matrix $\tilde{\boldsymbol{B}}$. Recall that, considering the i^{th} component of $\boldsymbol{p}$, and the i^{th} component of $\boldsymbol{q}$, we obtain the i^{th} row of the matrix:

$$(p_i, q_i) \longmapsto (\tilde{\boldsymbol{B}})_i \overset{\triangle}{=} \frac{1}{p_i + q_i}(p_i \cdot (\boldsymbol{B}^1)_i + q_i \cdot (\boldsymbol{B}^2)_i),$$

for all i such that $p_i + q_i \neq 0$. For the first extreme dual optimal solution, this condition is met for $i = 2, 3$, and thus we have:

$$\begin{array}{lll} (p_2, q_2) & \longmapsto & \frac{1}{0+1}(0 \cdot (1,1,1) + 1 \cdot (0,0,1)) = (0,0,1) \\ (p_3, q_3) & \longmapsto & \frac{1}{10/7 + 10/7}\left[(10/7) \cdot (1/2, 1/5, 0) + (10/7) \cdot (1/5, 1/2, 0)\right] \\ & & = \frac{1}{2}(1/2, 1/5, 0) + \frac{1}{2}(1/5, 1/2, 0) = (7/20, 7/20, 0). \end{array}$$

For the second extreme dual optimal solution, the nonzero rows are $i = 1, 3$, and we get:

$$\begin{aligned}(p_1, q_1) &\longmapsto \frac{1}{1+0}(1 \cdot (0,0,1) + 0 \cdot (1,1,1)) = (0,0,1)\\ (p_3, q_3) &\longmapsto \frac{1}{20/7}[(10/7) \cdot (1/2, 1/5, 0) + (10/7) \cdot (1/5, 1/2, 0)] = (7/20, 7/20, 0).\end{aligned}$$

Next, according to Proposition 3.6, we consider the set of matrices in $\mathcal{P}$ that share one of the nonzero rows of $\tilde{B}_1$, and similarly for $\tilde{B}_2$. These are specified by the convex combination coefficients that form the non-zero rows. The two convex combinations for the first dual solution are formed by the coefficients $\alpha_2 = (0,1)$ and $\alpha_3 = (1/2, 1/2)$. The second dual solution has convex combination coefficients $\alpha_1 = (1,0)$ and $\alpha_3 = (1/2, 1/2)$. Therefore, any strictly improving partition must be such that no single region contains both matrices $\{B^2, \frac{1}{2}B^1 + \frac{1}{2}B^2\}$, nor the two matrices $\{B^1, \frac{1}{2}B^1 + \frac{1}{2}B^2\}$. Evidently, no such partition into 2 (convex) regions exists. Therefore 2-adaptability cannot satisfy these two necessary conditions, and thus (in this example) is no better than the static robust solution of (3.6.20). This illustrates point (2) above: the necessary conditions corresponding to the two extreme points of $\mathcal{C}_0$ are alone sufficient to prove that 2-adaptability is no better than the static robust solution.

Next we consider the more general case $\mathcal{C}_\eta$ and $\mathcal{A}_\eta$. We consider a few different values of η: $\eta_1 = (27/7) - 3.8, \eta_2 = (27/7) - 3.2$, and $\eta_3 = (27/7) - 2.9$. We generate the extreme points (p, q) of $\mathcal{C}_{\eta_1}$, and the points of $\mathcal{B}_{\eta_1}$ to which they map. The polytope $\mathcal{C}_{\eta_1}$ has 12 extreme points[2]. These yield four non-redundant necessary conditions:

$$\begin{array}{ll}\mathcal{N}_1 = \{B^2, \frac{10}{21}B^1 + \frac{11}{21}B^2\}; & \mathcal{N}_2 = \{B^1, \frac{11}{21}B^1 + \frac{10}{21}B^2\};\\ \mathcal{N}_3 = \{\frac{1}{50}B^1 + \frac{49}{50}B^2, \frac{1}{2}B^1 + \frac{1}{2}B^2\}; & \mathcal{N}_4 = \{\frac{49}{50}B^1 + \frac{1}{50}B^2, \frac{1}{2}B^1 + \frac{1}{2}B^2\}.\end{array}$$

While there exists no partition into only two regions that can simultaneously satisfy these four necessary conditions, the three-region split $[0,1] = [0, 1/6] \cup [1/6, 5/6] \cup [5/6, 1]$ does satisfy $\mathcal{N}_1 - \mathcal{N}_4$; we can check that none of the sets $\mathcal{N}_i$, $1 \leq i \leq 4$, are contained within any single region of the proposed partition. In fact, this partition decreases the cost by $0.4672 \geq \eta_1$. The polytope $\mathcal{C}_{\eta_2}$ has 12 vertices. The non-redundant constraints generated by points of $\mathcal{B}_{\eta_2}$ corresponding to the extreme points of $\mathcal{C}_{\eta_2}$, are

$$\begin{array}{ll}\mathcal{N}_1 = \{B^1, \frac{28}{33}B^1 + \frac{5}{33}B^2\}; & \mathcal{N}_2 = \{B^2, \frac{28}{33}B^2 + \frac{5}{33}B^1\}\\ \mathcal{N}_3 = \{\frac{77}{100}B^1 + \frac{23}{100}B^2, \frac{1}{2}B^1 + \frac{1}{2}B^2\}; & \mathcal{N}_4 = \{\frac{23}{100}B^1 + \frac{77}{100}B^2, \frac{1}{2}B^1 + \frac{1}{2}B^2\}.\end{array}$$

[2] These computations were done using the software CDD by Komei Fukuda [68]. This is an implementation of the double description method. See also http://www.cs.mcgill.ca/ fukuda/soft/cddman/node2.html for further details.

It is easy to check that these four necessary conditions are not simultaneously satisfiable by any partition with only three (convex) regions. Indeed, at least 5 are required. This is another illustration of point (2) from above: a small set of four necessary conditions suffices to prove that 3-adaptability cannot improve the static robust solution by more than $\eta_2 = (27/7) - 3.2$.

The smallest η at which the necessary conditions corresponding to the extreme points of $\mathcal{C}_\eta$ provide a certificate that at least 5 regions are required for any partition to achieve an η-improvement or greater, is $\hat{\eta} \approx 27/7 - 3.2770 \approx 0.5801$. This illustrates point (3) above: examining values of $\eta \in [0, \hat{\eta}]$, the four necessary conditions implied by the extreme points of $\mathcal{C}_\eta$ are sufficient to reveal that two-adaptability is no better than the static robust solution, and in addition, they reveal the limit of 3-adaptability. Furthermore, they reveal the optimal 3-partition to be: $[0,1] = [0, \hat{\lambda}] \cup [\hat{\lambda}, 1-\hat{\lambda}] \cup [1-\hat{\lambda}, 1]$, for $\hat{\lambda} \approx 0.797$.

Finally, let us consider $\eta_3 = (27/7) - 2.9$. In this case, we are asking for more improvement than even the completely adaptable formulation could provide (recall $\text{CompAdapt}(\mathcal{P}) = 3$). In short, such improvement is not possible within our framework of a deterministic adversary. Proposition 3.4 tells us how the polytope $\mathcal{C}_{\eta_3}$ and the set $\mathcal{B}_{\eta_3}$ witness this impossibility. The polytope $\mathcal{C}_{\eta_3}$ has 31 vertices. It is enough to consider one of these vertices in particular: $\boldsymbol{v} = (9/10, 1/10, 9/5), (0,0,0)$. The corresponding necessary condition is: $\mathcal{N} = \{\boldsymbol{B}^1, \boldsymbol{B}^1, \boldsymbol{B}^1\}$. Evidently, no number of partitions can ever satisfy this necessary condition. Indeed, this is precisely what Proposition 3.4 says: if progress η is not possible, it must be because $\bar{\mathcal{A}}_\eta \cap \mathcal{P} \neq \emptyset$.

Next we illustrate point (4), by showing that for the problem (3.6.20) above, the affine adaptability proposal of Ben-Tal et al. ([12]) is no better than the static robust formulation, even though 3-adaptability significantly improves the static robust solution, and thus outperforms affine adaptability. In Figure 3-2 on the left, we have the actual optimal solutions for the completely adaptable problem. For every $\lambda \in [0,1]$, the decision-maker has an optimal response, $\boldsymbol{y}(\lambda) = (y_1(\lambda), y_2(\lambda), y_3(\lambda))$. The figure on the right illustrates the optimal completely adaptable cost as a function of λ, as well as the optimal static robust cost (the line at the top) and then the cost when the decision-maker selects 3 and 5 contingency plans, respectively. $\text{CompAdapt}(\mathcal{P})$ is given by

$$\begin{array}{rll} \text{CompAdapt}(\mathcal{P}) = & \max: & y_1(\lambda) + y_2(\lambda) + y_3(\lambda) \\ & \text{s.t.}: & \lambda \in [0,1]. \end{array}$$

We can see from the figure that indeed this value is 3.

Next, consider the optimal affine adaptability. In (3.2.7) we define affine adaptability for the two stage problem, however we can easily apply this to single stage optimization by allowing all the decision-variables to depend affinely on the uncertainty. Here the uncertainty is one-dimensional, parameterized by $\lambda \in [0,1]$, so we let

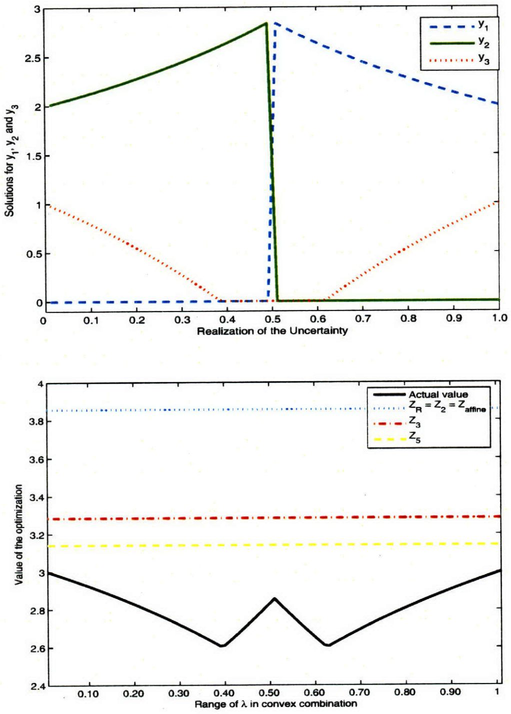

Figure 3-2. The figure on the left illustrates the optimal response policy for the decision-maker. The optimal response function is far from linear. In the figure to the right the lowest curve is the value of the nominal LP as a function of the realization of the uncertainty. The next three lines, Z_5, Z_3, Z_R, illustrate the value of 5,3-adaptability, and the static robust value, respectively. The static robust value coincides with the value of affine adaptability.

$\boldsymbol{y}^{\text{aff}}(\lambda)$ denote the optimal affine solution. The third component, $y_3^{\text{aff}}(\lambda)$ must satisfy: $y_3^{\text{aff}}(0), y_3^{\text{aff}}(1) \geq 1$. Therefore, by linearity, we must have $y_3^{\text{aff}}(\lambda) \geq 1$ for all $\lambda \in [0,1]$. Furthermore, for $\lambda = 1/2$, we must also have

$$\frac{1}{2}\left(\frac{1}{2}y_1^{\text{aff}}(1/2) + \frac{1}{5}y_2^{\text{aff}}(1/2)\right) + \frac{1}{2}\left(\frac{1}{5}y_1^{\text{aff}}(1/2) + \frac{1}{2}y_2^{\text{aff}}(1/2)\right) \geq 1,$$

which implies, in particular, that $y_1^{\text{aff}}(1/2) + y_2^{\text{aff}}(1/2) \geq \frac{20}{7}$. The cost obtained by affine adaptability is

$$\begin{aligned} \text{Affine}(\mathcal{P}) = \quad \max: &\quad y_1^{\text{aff}}(\lambda) + y_2^{\text{aff}}(\lambda) + y_3^{\text{aff}}(\lambda) \\ \text{s.t.}: &\quad \lambda \in [0,1]. \end{aligned}$$

This is at least the value at $\lambda = 1/2$. But this is: $y_1^{\text{aff}}(1/2) + y_2^{\text{aff}}(1/2) + y_3^{\text{aff}}(1/2) \geq 20/7 + 1 = 27/7$, which is the static robust value. Therefore, in this case, affine adaptability is no better than the static robust value. On the other hand, as illustrated in Figure 3-2, 3-adaptability is sufficient to significantly improve the cost to the decision-maker, and 5-adaptability is better still. Moreover, since this problem satisfies the Continuity Assumption, by Proposition 3.3, $\text{Adapt}_k(\mathcal{P}) \to \text{CompAdapt}(\mathcal{P})$ as k increases, so we can further improve the cost with more adaptability. Thus, we illustrate point (4) from above. ▲

Next we illustrate points (5) and (6) above by presenting a modification of the previous example. Consider:

$$\begin{array}{rl} \min: & y_2 \\ \text{s.t.}: & \begin{pmatrix} a_1 & a_2 \\ b_1 & b_2 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \geq \begin{pmatrix} 1 \\ -1 \\ 0 \end{pmatrix} \\ & \forall \begin{pmatrix} a_1 & a_2 \\ b_1 & b_2 \end{pmatrix} \in \text{conv} \left\{ \begin{pmatrix} \frac{1}{2} & \frac{1}{5} \\ -\frac{1}{2} & -\frac{1}{5} \end{pmatrix}, \begin{pmatrix} \frac{1}{5} & \frac{1}{2} \\ -\frac{1}{5} & -\frac{1}{2} \end{pmatrix} \right\} \end{array} \tag{3.6.21}$$

The static robust solution to (3.6.21) is $\boldsymbol{y}_R = (10/7, 10/7)$, and hence $\text{Static}(\mathcal{P}) = 10/7$. On the other hand, for any realization of the uncertain matrix,

$$\begin{pmatrix} a_1 & a_2 \\ b_1 & b_2 \end{pmatrix} = \lambda \begin{pmatrix} \frac{1}{2} & \frac{1}{5} \\ -\frac{1}{2} & -\frac{1}{5} \end{pmatrix} + (1-\lambda) \begin{pmatrix} \frac{1}{5} & \frac{1}{2} \\ -\frac{1}{5} & -\frac{1}{2} \end{pmatrix},$$

the solution $(y_1, y_2) = (5 - 3\lambda, 0)$ is feasible, and hence optimal for the nominal problem. The optimal response function in this case is affine. Here, $\text{CompAdapt}(\mathcal{P}) = 0$, and the gap is $10/7$. Consider now any partition of the uncertainty set (i.e., the interval $[0, 1]$) into finitely (or even countably many) regions. At least one region of the partition must contain more than one point of the interval, otherwise we would have uncountably many regions. Let $\tilde{\mathcal{P}}$ denote this region, with $\lambda_1 < \lambda_2$ both elements of $\tilde{\mathcal{P}}$. The static robust problem over this set $\tilde{\mathcal{P}}$, is lower bounded by

$$\begin{array}{rl} \min: & y_2 \\ \text{s.t.}: & \left[\lambda_1 \begin{pmatrix} \frac{1}{2} & \frac{1}{5} \\ -\frac{1}{2} & -\frac{1}{5} \end{pmatrix} + (1-\lambda_1) \begin{pmatrix} \frac{1}{5} & \frac{1}{2} \\ -\frac{1}{5} & -\frac{1}{2} \end{pmatrix}\right] \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \geq \begin{pmatrix} 1 \\ -1 \end{pmatrix} \\ & \left[\lambda_2 \begin{pmatrix} \frac{1}{2} & \frac{1}{5} \\ -\frac{1}{2} & -\frac{1}{5} \end{pmatrix} + (1-\lambda_2) \begin{pmatrix} \frac{1}{5} & \frac{1}{2} \\ -\frac{1}{5} & -\frac{1}{2} \end{pmatrix}\right] \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} \geq \begin{pmatrix} 1 \\ -1 \end{pmatrix} \\ & y_2 \geq 0 \end{array}$$

As $\lambda_1 \neq \lambda_2$, the point $\boldsymbol{y}_R = (10/7, 10/7)$ is the only point in the feasible region, and thus it must also be optimal; hence the value is not improved from $10/7$. Note, moreover, that this example violates the Continuity Assumption: for any two (even infinitesimally close) realizations of the uncertainty, the only common feasible point is $y_R = (10/7, 10/7)$, which is not within ε of optimality for any $\varepsilon < 10/7$. Thus, we illustrate point (5), and show that the Continuity Assumption may not be removed from Proposition 3.3. Recall that Proposition 3.4 says that finite adaptability can strictly improve the solution if and only if $\mathcal{P} \cap \bar{\mathcal{B}} = \emptyset$. Here, we can indeed check that $\mathcal{P} \cap \mathcal{B} = \emptyset$. However, the set of dual optimal solutions to (3.6.21) is unbounded, and the set $\mathcal{B}$ is not closed. With some work, we can check that, e.g.,

$$\begin{pmatrix} \frac{1}{2} & \frac{1}{5} \\ -\frac{1}{2} & -\frac{1}{5} \\ 0 & 1 \end{pmatrix} \in \bar{\mathcal{B}}.$$

Thus, the conclusion of Proposition 3.4 holds, and in particular, as we point out in (6) above, taking the closure of $\mathcal{A}_\eta$ cannot be relaxed. ▲

We take up this example again in Chapter 4. There, we pursue a different hierarchy of adaptability, and we show that with quartic polynomial adaptability, or with piecewise affine adaptability, one can recover the optimal solution.

■ 3.7 Heuristic Algorithms

In Section 3.4, we have formulated the k-adaptability problem as a bilinear optimization problem, which, in general, is difficult to solve. Moreover, in Section 3.5 we have established that even the 2-adaptability problem is NP-hard to solve. There, we give a sufficient condition for theoretical tractability. However, this sufficient condition may not apply, and in any case, in large scale optimization problems such as the one discussed in Section 3.8.1, we seek practically efficient and implementable solutions. In this section, we propose a heuristic tractable algorithm. We restrict ourselves to an infinite class of partitions from which selecting the optimal partition can be done efficiently.

The algorithm is motivated by the results of Section 3.3. There, the necessary conditions we derive say that good partitions divide points of $\mathcal{P}$ which must be separated. We try to do exactly that. The algorithm is based on the following observation.

Lemma 3.13 *Consider the set of partitions $\mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2$ given by a hyperplane division of $\mathcal{P}$. If the orientation (i.e., the normal vector) of the hyperplane is given, then selecting the optimal partitioning hyperplane with this normal can be done efficiently.*

PROOF. Consider the one-dimensional subspace parallel to the given normal. Parameterize the set of hyperplanes with given normal by their point of intersection, $t \in \mathbb{R}$, with this subspace (the location of intersection completely specifies the hyperplane). Let $\mathcal{P} = \mathcal{P}_1(t) \cup \mathcal{P}_2(t)$ be the resulting partition for any value of t, and let $z(t)$ denote the value of the resulting 2-adaptability problem. Then $z(t)$ is quasi-convex in t, for if $t_1 < t_2 < t_3$, then either $\mathcal{P}_1(t_1) \subseteq \mathcal{P}_1(t_2) \subseteq \mathcal{P}_1(t_3)$, or $\mathcal{P}_1(t_1) \supseteq \mathcal{P}_1(t_2) \supseteq \mathcal{P}_1(t_3)$, and vice versa for $\mathcal{P}_2(t)$. Therefore, the values of $\text{Static}(\mathcal{P}_1(t))$ are either increasing, or decreasing, and vice versa for $\text{Static}(\mathcal{P}_2(t))$. Quasi-convexity then follows, concluding the proof. □

Algorithm 1: Let $\mathcal{P} = \text{conv}\{\boldsymbol{B}^1, \dots, \boldsymbol{B}^K\}$.

1. For every pair (i, j), $1 \leq i \neq j \leq K$, let $\boldsymbol{v}_{ij} = \boldsymbol{B}^j - \boldsymbol{B}^i$ be the unique vector they define.

2. Consider the family of hyperplanes with normal $\boldsymbol{v}_{ij}$.

3. Solve the quasi-convex problem, and let $\boldsymbol{H}_{ij}$ be the hyperplane that defines the optimal hyperplane partition of $\mathcal{P}$ within this family.

4. Select the optimal pair (i, j) and the corresponding optimal hyperplane partition of $\mathcal{P}$.

This algorithm can be applied iteratively as a heuristic approach to computing 2^d-adaptability. In Section 3.8, we implement this algorithm to compute 2,4-adaptability.

Section 3.3.2 provides an approach to strengthen the above algorithm. The algorithm selects the optimal hyperplane from the set of hyperplanes that have normal vector defined by a pair of extreme points of $\mathcal{P}$. By adding explicitly more points that are in the interior of $\mathcal{P}$, we enlarge the space of hyperplanes over which the algorithm searches. In Section 3.3.2, we illustrate a procedure for obtaining necessary conditions that any "good" partition must satisfy. These conditions essentially contain requirements that certain collections of points of $\mathcal{P}$ should not be contained within any single region of the partition. By including (a partial set of) the points corresponding to a list of necessary conditions, we guarantee that the set of partitions considered include partitions that meet the necessary conditions. In effect, this gives a structured approach to increasing the size of the family of partitions considered.

Algorithm 2: Let the uncertainty set be given by inequalities: $\mathcal{P} = \{\boldsymbol{B} : \boldsymbol{\alpha}_i'\text{vec}(\boldsymbol{B}) \leq 1,\ 1 \leq i \leq N\}$, where $\text{vec}(\boldsymbol{B})$ is vector consisting of the rows of $\boldsymbol{B}$.

1. For every defining facet $\boldsymbol{\alpha}_i$ of $\mathcal{P}$, let $\boldsymbol{v}$ be the unique normal vector.

2. Consider the family of hyperplanes with normal $\boldsymbol{v}$.

3. Solve the quasi-convex problem, and let $\boldsymbol{H}_i$ be the hyperplane that defines the optimal hyperplane partition of $\mathcal{P}$ within this family.

4. Select the optimal index i and the corresponding optimal hyperplane partition of $\mathcal{P}$.

3.8 Computational Examples

In this section, we report on the performance of the heuristic algorithm of Section 3.7. First, we discuss the problem of Air Traffic Control. We discuss why finite adaptability may be an appropriate framework for adaptability, both in terms of theoretical and practical considerations. The full details of the model, and the numerical computations are in Chapter 5. To illustrate the main idea and the applicability of finite adaptability, we consider a network flow problem with uncertain capacity constraints, to model a very small air traffic control problem. Next, in Section 3.8.2, we consider a minimum cost robust scheduling problem with integer constraints. These randomly generated examples are meant to illustrate the applicability of k-adaptability, and some types of problems that can be considered. In the final part of this section, 3.8.3, we explore a large collection of randomly generated instances of the scheduling problem without integer constraints. We consider different problem size, and types and level of uncertainty, in an effort to obtain some appreciation in the generic case, for the benefit of the first few levels of the adaptability hierarchy, and for the behavior of the algorithm of Section 3.7.

3.8.1 An Example from Air Traffic Control

The cost of the delay to the airline industry is staggering, to the point where passenger safety and industry growth threaten to become competing objectives ([30]). Over 70% of delay is caused by convective weather, as weather patterns reduce the takeoff and landing capacity of airports, as well as the capacity of air traffic corridors. The air traffic control and scheduling problem, including managing ground holding times, dynamic rerouting, air holding times, and aircraft and crew continuation, is a large-scale, multi-stage optimization problem with sequentially revealed uncertainty, namely, the weather update and forecast. Because the uncertainty generated by the weather is primarily in the capacity constraints, the uncertainty is largely non-constraintwise. In particular, a static robust formulation cannot capture correlation among the capacity constraints. Therefore, a naive robust implementation for the uncertainty in the capacity constraints results in a significantly overly-conservative formulation. In [23], we approach this problem using finite adaptability. A key feature of the problem is the fact that even the deterministic formulation (i.e., with perfect forecast) has millions of variables and constraints. There are 30-50 thousand commercial flights that take off and land daily in the United States, and the numbers for European Air Space are similar. Thus, practical computational considerations are of highest importance. In addition, integrality constraints enter naturally; therefore any framework for adaptability must be able to accommodate integer constraints, without a significant increase in the size of the resulting problem. For small k, k-adaptability does not increase the size of the problem beyond our computational means. Furthermore, we can treat integer constraints. The computational complexity of k-adaptability lies largely in the partitioning of the uncertainty set. We believe that this particular large-scale application is an example that illustrates well the applicability of finite adaptability, because of the nature of the uncertainty.

Indeed, the 0-60 minute forecast of the weather is essentially deterministic as far as the impact on capacity is concerned. Moreover, the nature of the capacity-affecting weather uncertainty is particularly suitable for application of finite adaptability, because the uncertainty in longer term weather prediction, say, 0-6 hours, can be modeled successfully using low-dimensional uncertainty sets. While the capacity vector is potentially very high dimensional (roughly it is equal in size to the number of airports and air traffic sectors) the weather uncertainty that drives the capacity uncertainty generates an uncertainty set where the resulting capacities are highly correlated. Current weather prediction capabilities are very successful at identifying the existence, and general trajectory of a storm system large enough to significantly affect capacity. If a storm front follows a projected path, but moves more quickly or slowly than originally forecast, the effect on the entire capacity vector is highly correlated, since multiple sectors are affected by the same storm system. Similarly, if a single storm front breaks into two or three smaller storms, the vector of capacities changes in a tightly correlated fashion. It is precisely because of this correlation that the robust formulation is overly conservative, while k-adaptability with small k can offer significant gains.

Of particular importance for the finite adaptability approach, is that the low dimensionality of the capacity uncertainty is independent of the number of planes, routes, airports of origin and destination; in particular, the dimensionality of the uncertainty does not scale with the size of the problem. Thus, while the problem itself is naturally very high dimensional, the positive complexity results of Proposition 3.12 promise a tractable problem. More important that this theoretical guarantee, however, is the fact that empirically, the heuristic algorithm of Section 3.7 seems to perform well in practical problems of realistic size. For more details on the characteristics of the weather uncertainty, and tools for weather prediction, we refer the reader to [111] and references therein. For more information on the particular robust models, and full numerical details about the problem, we refer the interested reader to Chapter 5, and also [23]. For further background on the problem, and for other approaches to the air traffic control problem, see, e.g., [30], [95], [99], and references therein.

For the purpose of illustrating the application, and the behavior of the algorithm of Section 3.7, we consider a small example of a network flow problem subject to capacity uncertainty, which is modeled as RHS-uncertainty. The idea comes from a model of air traffic control where capacity is affected by an arriving storm front as discussed above. The exact time and location where a storm front affects an air traffic corridor are subject to some uncertainty. In Figure 3-3 we show a simple network. The edges denote air traffic corridors, whose capacity in clear weather conditions is 20. The corridors represented by edges 1,4,8 and 2,5,9 correspond to the more direct routes, and the cost of using each edge is nominally set to 1. The edges 3,5,7, however, represent a geographically much longer path that involves possibly rerouting or delaying aircraft with other origin and destination, and hence their cost is set to 6. The decision-maker (the air traffic controller) must schedule 30 units of flow (airplanes) from the source, s, to the sink, t, without exceeding the capacity of any link. Suppose that in the considered time-frame, the capacity of the central corridor marked with a solid line remains unaffected (as it is geographically distant from the top and bottom paths) while the capacity of the exterior corridors may be decreased due to convective weather. We consider an uncertainty region described by four extreme points, each representing some capacity degradation. The first extreme point represents loss of 24% of capacity in edge 1 and 28% in edge 8, the next 6% in edge 1 and 22% in 4, the third 36% in edge 2 and 44% in 5, and the fourth 11% in edge 5 and 12% in 9. In the completely adaptable formulation where the decision-maker has exact information about the precise impact of the storm, the flight controller can reroute flights so as to completely avoid incurring any additional cost due to the storm. Under the static robust formulation, however, the threatening storm increases the cost by over 49%. The 2-adaptability computed by our heuristic algorithm outperforms the static robust cost by over 36%. Thus, under 2-adaptability, the cost of the storm is at most 13% above the clear-weather cost, down from 49%.

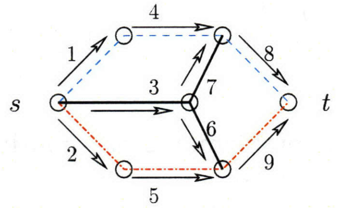

Figure 3-3. Here we show a small network illustrating a potential application of k-adaptability to network flows.

3.8.2 Robust Scheduling: Integer Constraints

Suppose we have m products, and each product can be completed partially or fully at one of n stations, and the stations work on many products simultaneously so that no product blocks another. Thus the decision variables, y_j, are for how long to operate station j. The matrix $\boldsymbol{B} = \{b_{ij}\}$ gives the rate of completion of product i at station j. Running station j for one hour we incur a cost c_j. To minimize the cost subject to the constraint that the work on all products is completed, we solve:

$$\begin{array}{rlr} \min: & \sum_{j=1}^{n} c_j y_j & \\ \text{s.t.}: & \sum_{j=1}^{n} b_{ij} y_j \geq 1, & 1 \leq i \leq m \\ & y_j \geq 0, & 1 \leq j \leq n. \end{array}$$

In the static robust version of the problem, the rate matrix $\boldsymbol{B}$ is only known to lie in some set $\mathcal{P}$. How much can we reduce our cost if we can formulate 2 (in general k) schedules rather than just one? Particularly in the case where we have to make binary decisions about which stations to use, there may be some cost in having k contingency plans prepared, as opposed to just one. It is therefore natural to seek to understand the value of k-adaptability, so the optimal trade-off may be selected.

In Section 3.8.3, we generate a large ensemble of these problems, varying the size and the generation procedure, and we report average results. Here, we consider only one instance from one of the families below, and impose binary constraints, so that each station must be either on or off: $y_i \in \{0, 1\}$.

The heuristic algorithms proposed in Section 3.7 are tractable because of the quasi-convexity of the search for the optimal dividing hyperplane and by the limited set of normal directions considered. Both these factors are independent of the continuous or discrete nature of the underlying problem. Indeed, all that is required for the algorithms is a method to solve the static robust problem.

We consider an instance with 6 products and 6 stations, where the uncertainty set is the convex hull of six randomly generated rate matrices. Without the integer constraints, the value of the static robust problem is 3.2697, and the completely adaptable value is bounded below by 2.8485. The value of the 2-adaptability solution is 3.1610, and for 4-adaptability the value is 3.0978. Thus, 2-adaptability covers 25.8% of the gap, and 4-adaptability covers just over 40% of the gap. As we see from the results of the next section, these numbers are typical in our ensemble. When we add integer constraints, the static robust cost is 5, i.e., 5 stations must be turned on. The completely adaptable value is 4. The 2-adaptability solution also improves the static robust cost, lowering it to 4. Thus, in this case a single split of the uncertainty region reduces the cost as much as the full completely adaptable formulation.

■ 3.8.3 Robust Scheduling

We consider a large collection of randomly generated instances of the scheduling problem above, without integer constraints. First, we suppose that the extreme points of $\mathcal{P}$ are generated uniformly at random, their elements drawn *iid* from a uniform distribution. Next, we consider another random instance generation procedure, where the extreme points of $\mathcal{P}$ come from a specific degrading of some number of products. That is, we may have nominal values $\{b_{ij}\}$, but in actuality some collection (typically small) of the m products may take longer to complete on each machine, than indicated by the nominal values. Here each extreme point of $\mathcal{P}$ would be constructed from the nominal matrix $\boldsymbol{B}$, degraded at some small number of rows. We generate random instances of this problem by generating a nominal matrix $\boldsymbol{B}$, and then degrading each row individually. This corresponds to choosing robustness that protects against a single product being problematic and requiring more time at the stations.

We are interested in several figures of merit. We consider the gap between the static robust problem and complete adaptability. As we have remarked above, we note that complete adaptability is typically difficult to compute exactly ([12]). Therefore for all the computations in this section, we compute upper bounds on the gap between the static robust and the completely adaptable values. Thus, we present lower bounds on the benefit of adaptability and the performance of the heuristic algorithm. We obtain upper bounds on the gap by approximating the completely adaptable value by random sampling. We sample 500 points independently and uniformly at random from the uncertainty set. Since the truly worst case may not be close to one of these sampled points, the completely adaptable value may in fact be worse than reported, thus making the gap *smaller*. Thus our random approximation gives a conservative bound on the true gap. Next, we compute the extent to which 2- and 4-adaptability, as computed by the algorithm of Section 3.7, close this gap.

We summarize the computational examples by reporting the size of the instances and some statistics of the simulations. In each category, every number represents the average of 50 independently generated problem instances of size as shown. These results are contained in Table 3.1. There, we give the average, minimum, and maximum gap between the static robust and the completely adaptable values. We give this as a fraction of the static robust value, that is, GAP $=$ $(\text{Static}(\mathcal{P})-\text{CompAdapt}(\mathcal{P}))/\,\text{Static}(\mathcal{P})$. Then we report the average percentage of this gap covered by 2-adaptability and 4-adaptability, as computed by the heuristic algorithm.

The table illustrates several properties of the gap, and of adaptability. We have considered several examples where we fix the number of products and the number of stations (i.e., we fix the size of the matrices) and then vary the size of the uncertainty set, i.e., the number of extreme points. In all such examples, we see that the average gap increases as the level of the uncertainty grows. Indeed, this is as one would expect. Furthermore, we see that the quality of 2,4-adaptability decreases as the size of the uncertainty set grows. Again this is as one would expect, as we are keeping the amount of adaptability, and the problem dimension constant, while increasing the number of extreme points of the uncertainty set. For the 6×6 matrices, 4-adaptability covers, on average, over 70% of the gap. That is, with only 4 contingency plans, on average we do over 70% as well as the best possible attainable by any amount of adaptability. When we double the size of $\mathcal{P}$, the average performance of 2-adaptability drops from over 63% to just over 42%, while the performance of 4-adaptability drops from over 70% to

Matrix Size	Size of $\mathcal{P}$	Avg Gap %	2-Adapt %	4-Adapt %
6×6	$K = 3$	10.10	63.22	70.70
6×6	$K = 6$	14.75	42.72	52.33
6×6	$K = 8$	18.45	39.15	47.42
10×10	$K = 3$	10.12	50.67	63.29
10×10	$K = 5$	14.22	38.58	49.36
10×10	$K = 10$	18.27	31.17	40.18
15×25	$K = 3$	8.06	39.27	54.53
15×25	$K = 5$	10.73	25.12	35.52
15×25	$K = 7$	13.15	18.21	26.84

Table 3.1. The matrices in these instances were generated independently. The first group of two columns identifies the size of the problem, where by matrix size we mean the "number of products by number of stations," and by size of $\mathcal{P}$ we indicate the number of extreme points. We note that the average gap between the static and adaptable formulations increases with the size of the uncertainty set $\mathcal{P}$. Also, the benefit of 2,4-adaptability decreases as the size of the set $\mathcal{P}$ increases.

about 52%. A similar phenomenon occurs in the other examples as well.

We also report the results of the computations for the case where the uncertainty set $\mathcal{P}$ corresponds to the case where at most one product is degraded. That is, we form $\mathcal{P}$ by degrading each row of a matrix B individually. The results from this random generation procedure are comparable to the first procedure. The results are reported in Table 3.2.

Matrix Size	Size of $\mathcal{P}$	Avg Gap %	2-Adapt %	4-Adapt %
3×6	$K = 3$	9.08	70.61	78.36
6×6	$K = 6$	14.24	54.62	66.34
3×20	$K = 3$	10.61	28.67	45.16
5×20	$K = 5$	15.90	33.78	47.60
10×20	$K = 10$	21.17	22.35	31.50
3×25	$K = 3$	10.92	52.94	65.16
5×25	$K = 5$	15.81	32.83	45.90
3×50	$K = 3$	10.66	44.04	59.06

Table 3.2. The matrices in these instances were generated with dependent matrices, as explained above. In this example again we note the same trends as for the first example: The gap between the static and the adaptable increases with the size of the uncertainty set, and the value of 2,4-adaptability is better for low-dimensional uncertainty sets than for high-dimensional uncertainty.

■ 3.9 Conclusion

We have proposed a notion of finite adaptability. This corresponds to choosing a finite number of contingency plans, as opposed to a single static robust solution. We have shown that this is equivalent to partitioning the uncertainty space, and receiving ahead of time coarse information about the realization of the uncertainty, corresponding to one of the chosen partitions.

The structure of this adaptability is designed to reduce the geometric gap between $\mathcal{P}$ and $(\mathcal{P})_R$, which is exactly the reason the static robust solution may be conservative. In this chapter, we have focused on exploiting non-constraintwise uncertainty. We consider elsewhere the value of adaptability in the face of non-convex uncertainty sets. This notion of finite adaptability establishes a hierarchy of adaptability that bridges the gap between the static robust formulation, and the completely adaptable formulation. Thus, we introduce the concept of the value of adaptability. We believe that the finiteness of the proposal, as well as the hierarchy of increasing adaptability, are central to the chapter. The finiteness of the adaptability is appropriate in many application areas where infinite adjustability, and infinitesimal sensitivity, are either impossible due to the constraints of the problem, or undesirable because of the structure of the optimization, i.e., the cost. In addition to this, the inherent finiteness, and hence discrete nature of the proposal, makes it suitable to address adaptability problems with discrete variables. We expect that this benefit should extend to problems with non-convex constraints.

In problems where adaptability, or information is the scarce resource, the hierarchy of finite adaptability provides an opportunity to trade off the benefits of increased adaptability, versus its cost.

On the other hand, as we demonstrate, obtaining optimal partitions of the uncertainty space can be hard. Thus, there is a need for efficient algorithms. We have proposed a tractable algorithm for adaptability. Numerical evidence indicates that its behavior is good.

CHAPTER 4

Adaptability via Sampling: Integer and Higher Order Models

In the last chapter, we considered a finite scheme for adaptability, explicitly constructing the regions for the piecewise constant adaptability functions. The focus was primarily on two stage models. Indeed, the finite nature of the adaptability seems to have inherent a combinatorial explosion in the number of stages, for multi-stage problems. Here we are interested in two-stage models as well, but we explicitly consider extensions to the multi-stage case as well. As in Chapter 3, we are interested in structuring a hierarchy of adaptability, bridging the gap between the static approach on the one extreme, and the fully adaptable approach on the other. The main contribution of this chapter is a polynomial time scheme for structuring such a hierarchy, that is also polynomial in the number of stages in the model.

We consider both continuous and discrete models for adaptability. Affine adaptability, as described in Chapter 2, was introduced in [12], and is a continuous model for adaptability that extends to multiple stages without a combinatorial explosion in the number of variables. Yet that framework does not provide a hierarchy of adaptability. Finite adaptability, as proposed in Chapter 3, does provide such a hierarchy, but does not extend well to multiple stages. Both of these two approaches take a Robust Optimization view of the uncertainty, assuming a deterministic set-based model. Here, we take a different approach, and assume that the behavior of the uncertainty is governed by an underlying distribution. Furthermore, we assume that we have black-box access to this distribution, i.e., we can sample from it, but otherwise we may know nothing further. This conceptual framework is similar to some work from the Stochastic Optimization literature, and in our context is closest to the recent work in [43]. We show that if we accept a reliability probability $(1 - \delta)$, and relax our feasibility requirement to being feasible with probability at least $(1 - \varepsilon)$, then we can construct higher order adaptability (affine, quadratic, and so on) simply by solving LPs. In addition, we propose a method for adaptability where the second stage variables are integral. On the way to proving probabilistic guarantees for this sampling-based proposal, we provide an extension of the results of [43] to the case of integer constraints. Our results make no assumption of complete, or constant recourse. Furthermore, for a fixed ε and δ, and a fixed order of adaptability, our results yield a polynomial time approach for multistage sampling. That is, the number of samples required grows polynomially in the number of stages.

■ 4.1 Introduction

The main focus of this chapter is on adaptability for multistage linear optimization problems, and also linear optimization problems with integer constraints. As we have done throughout the thesis, we consider multistage optimization problems with parameter uncertainty, where the uncertainty is realized sequentially, and therefore, future stage decisions are allowed to depend on the past realizations of the uncertainty. We consider a problem that has $K+1$ stages; we denote by $\boldsymbol{x}$ the first stage variable (the decision to be implemented now), and $\boldsymbol{y}_k$ the decision to be implemented at stage $1 \leq k+1 \leq K+1$. The uncertainty $\boldsymbol{z} = (\omega_1, \ldots, \omega_K)$ is revealed in stages, so that at stage $k+1$, the decision-maker sees the vector $\boldsymbol{z}_k \triangleq (\omega_1, \ldots, \omega_k)$. Thus the problem we consider is of the form:

$$\begin{aligned} \min: \quad & \boldsymbol{c}^{\top}\boldsymbol{x} + \textstyle\sum_{i=1}^{K} \boldsymbol{d}^{\top}\boldsymbol{y}_i(\boldsymbol{z}_i) \\ \text{s.t.}: \quad & \boldsymbol{A}(\boldsymbol{z})\boldsymbol{x} + \textstyle\sum_{i=1}^{K} \boldsymbol{A}_i(\boldsymbol{z})\boldsymbol{y}_i(\boldsymbol{z}_i) \leq \boldsymbol{b}, \end{aligned}$$

We note, in particular, that the stage $k+1$ decision $\boldsymbol{y}_k(\boldsymbol{z}_k)$ is a function of the uncertainty revealed up to that time. In general, the parameters of the problem (i.e., the matrices) may depend on the full realization of the uncertainty, and so may not be completely (deterministically) known to the decision-maker until the final stage. The dependence, therefore, reflects the causality of the decision-making sequence. Note further that we have not made any independence assumptions on the uncertainty revealed at different times i and j.

We also pay specific attention to the two-stage model:

$$\begin{aligned} \min: \quad & \boldsymbol{c}^{\top}\boldsymbol{x} + \boldsymbol{d}^{\top}\boldsymbol{y}(\omega) \\ \text{s.t.}: \quad & \boldsymbol{A}(\omega)\boldsymbol{x} + \boldsymbol{B}(\omega)\boldsymbol{y}(\omega) \leq \boldsymbol{b}. \end{aligned}$$

■ 4.2 A Brief Review & Our Contributions

Multistage formulations of optimization under uncertainty have a rich history in the research literature, and have attracted attention of researchers in a broad area of disciplines, including stochastic optimization, control, computer science, and many others. Chapter 2 contains a more thorough exposition of what has been done, and the pertinent references.

In this section, our aim is to provide only a brief overview of some of the different approaches to the multi-stage problem, and a few details about the results that are relevant to the material of this chapter. We mention only the high-level ideas, both with the intention of broadly introducing what has been done in the past, and also to contextualize the results of the present chapter.

■ 4.2.1 Robust Optimization

Robust Optimization has been a central theme of this thesis, and it is reviewed in depth in Chapter 2. For the purposes of this chapter, there are several aspects that bear further mentioning.

Recall that in Robust Optimization, we take a deterministic set-based view of uncertainty. A generic formulation, then, is:

$$\begin{array}{rl} \min: & \boldsymbol{c}^{\top}\boldsymbol{x} \\ \text{s.t.}: & f(\boldsymbol{x},\boldsymbol{\omega}) \leq 0 \quad \forall \boldsymbol{\omega} \in \Omega, \end{array} \tag{4.2.1}$$

where Ω is the uncertainty set.

The success of Robust Optimization is due to the tractability of this formulation for a large and useful class of problems, including LP and SOCP. This tractability of the single stage formulation, has not been successfully extended to the multi-stage case.

Consider the robust formulation of two-stage linear optimization:

$$\begin{array}{rl} \min: & \boldsymbol{c}^{\top}\boldsymbol{x} + \boldsymbol{d}^{\top}\boldsymbol{y} \\ \text{s.t.}: & \boldsymbol{A}(\boldsymbol{\omega})\boldsymbol{x} + \boldsymbol{B}(\boldsymbol{\omega})\boldsymbol{y}(\boldsymbol{\omega}) \leq b, \quad \forall \boldsymbol{\omega} \in \Omega. \end{array}$$

When $\boldsymbol{y}(\boldsymbol{\omega})$ is an arbitrary function of $\boldsymbol{\omega}$, the problem becomes:

$$\min: \quad \boldsymbol{c}^{\top}\boldsymbol{x} + \max_{\boldsymbol{\omega}\in\Omega} J(\boldsymbol{x},\boldsymbol{\omega}),$$

where we can think of J as the realization of the cost-to-go function when the first-stage decision is $\boldsymbol{x}$ and the uncertainty realization is $\boldsymbol{\omega}$. Thus we have:

$$J(\boldsymbol{x},\boldsymbol{\omega}) \triangleq \left[\begin{array}{rl} \min: & \boldsymbol{d}^{\top}\boldsymbol{y} \\ \text{s.t.}: & \boldsymbol{B}(\boldsymbol{\omega})\boldsymbol{y} \leq (\boldsymbol{b} - \boldsymbol{A}(\boldsymbol{\omega})\boldsymbol{x}) \end{array} \right].$$

Evaluation of $\max\{J(\boldsymbol{x},\boldsymbol{\omega}) \mid \boldsymbol{\omega} \in \Omega\}$ is typically intractable, because $J(\boldsymbol{x},\boldsymbol{\omega})$ is not concave in the optimization argument, $\boldsymbol{\omega}$.

The approach we follow in this chapter is to restrict the second stage function $\boldsymbol{y}(\boldsymbol{\omega})$ to have particular functional structure. The first approach along these lines within the Robust Optimization paradigm is due to Ben-Tal et al. ([12]). They restrict the function

$\boldsymbol{y}(\boldsymbol{\omega})$ to be an affine function of the uncertainty: $\boldsymbol{y}(\boldsymbol{\omega}) = \boldsymbol{Q}\boldsymbol{\omega} + \boldsymbol{q}$. Thus they have:

$$\begin{array}{rl} \min: & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{s.t.}: & \boldsymbol{A}(\boldsymbol{\omega}) + \boldsymbol{B}(\boldsymbol{\omega})[\boldsymbol{Q}\boldsymbol{\omega} + \boldsymbol{q}], \quad \forall \boldsymbol{\omega} \in \Omega. \end{array}$$

This reduces the problem to a single-stage robust optimization problem. Recall that the tractability of a single stage robust optimization problem, depends on the tractability of the family of constraints

$$f(\boldsymbol{x}, \boldsymbol{\omega}) \leq 0, \quad \forall \boldsymbol{\omega} \in \Omega,$$

which are equivalent to what we have called the **inner problem**:

$$\left[\begin{array}{rl} \max: & f(\boldsymbol{x}, \boldsymbol{\omega}) \\ \text{s.t.}: & \boldsymbol{\omega} \in \Omega \end{array} \right] \leq 0.$$

For affine adaptability, we have an inner problem for every constraint, i. Here the i^{th} inner problem becomes:

$$\begin{array}{rl} \max: & \boldsymbol{A}_i(\boldsymbol{\omega})\boldsymbol{x} + \boldsymbol{B}_i(\boldsymbol{\omega})[\boldsymbol{Q}\boldsymbol{\omega} + \boldsymbol{q}] \\ \text{s.t.}: & \boldsymbol{\omega} \in \Omega. \end{array}$$

This is an indefinite quadratic optimization problem, and thus is nonconvex. In general, this problem is NP-hard, with the notable exception the case where Ω is an ellipse (see Chapter 2). In [12], the authors use some previous results from [17] to obtain approximation results.

In addition to issues of computational complexity, a central question is the benefit of employing a particular adaptability scheme. As we have pointed out by example in Chapter 3, even in simple problems such as low-dimensional linear problems with linear dependence on the uncertainty, and with simple (e.g., one-dimensional) uncertainty sets, affine adaptability may fail (i.e., may be no better than the no adaptability case). Indeed, we saw a one-stage version of this phenomenon in Chapter 3, in Section 3.6. There, the example served as an illustration of when affine adaptability can fail, yet finite adaptability can significantly improve the solution. It is precisely in such a case when we need a hierarchy of adaptability, as presented in Chapter 3, or as presented here now. In Chapter 3 we saw that we can approach the optimal completely adaptable solution by finite adaptability, but doing so requires increasing without bound the number of regions in the partition. We carry this example again through this chapter, illustrating the benefit of nonlinear adaptability, and also piecewise affine adaptability. We are able to formulate such families of adaptability because of our sampling approach. We reproduce the example from Chapter 3, here in the context of a two-stage problem:

$$\begin{array}{lll} \min: & x & \\ \text{s.t.}: & x \geq y_1(\boldsymbol{\omega}) + y_2(\boldsymbol{\omega}) + y_3(\boldsymbol{\omega}) & \\ & \boldsymbol{B}(\boldsymbol{\omega})\boldsymbol{y}(\boldsymbol{\omega}) \geq \mathbf{1}, & \forall \boldsymbol{\omega} \in [0,1] \\ & x, \boldsymbol{y} \geq 0, & \end{array} \tag{4.2.2}$$

where $\boldsymbol{B}(\boldsymbol{\omega}) = \boldsymbol{\omega}\boldsymbol{B}_1 + (1-\boldsymbol{\omega})\boldsymbol{B}_2$, and

$$\boldsymbol{B}_1 = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ \frac{1}{2} & \frac{1}{5} & 0 \end{pmatrix} \qquad \boldsymbol{B}_1 = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 1 \\ \frac{1}{5} & \frac{1}{2} & 0 \end{pmatrix}.$$

The optimal second stage solution $\boldsymbol{y}(\boldsymbol{\omega})$ is plotted in Figure 4-1, illustrating that it is very nonlinear. To the best of our knowledge, there has not been any other continuous proposal for adaptability, that may improve on the affine model. A primary reason for this is that since the affine model already presents tractability problems because of the resulting quadratic dependence on the uncertainty, higher order models, or other continuous nonlinear models for adaptability would only exacerbate this problem. We revisit this example in Section 4.3.2, where we show that quadratic adaptability nearly closes the gap between the static and dynamic formulations. We also show that higher order polynomial adaptability closes the gap completely, as does a piecewise affine adaptability with only two pieces.

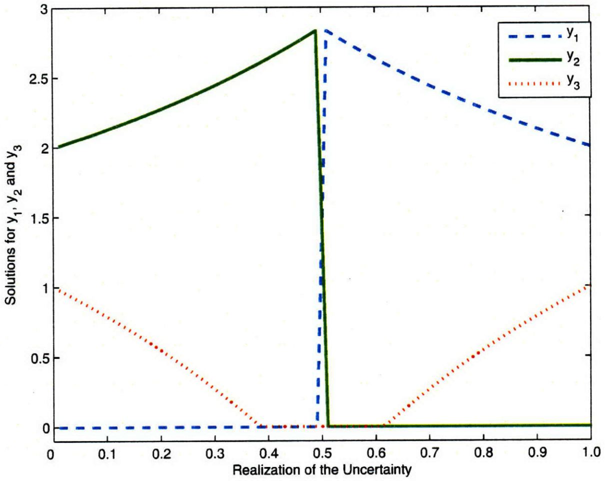

Figure 4-1. This figure gives a plot of optimal second stage solutions, as a function of the realization of the uncertainty. The figure shows that even in a simple example, the optimal solutions can be very non-linear.

■ 4.2.2 Stochastic Optimization

As discussed at greater length in Chapter 2, the Stochastic Optimization paradigm explicitly takes into account the probabilistic nature of the uncertain constraints (we refer the reader to the textbooks [81],[108], [35], and the references therein). Unlike the focus of Robust Optimization which is typically feasibility two-stage stochastic problem assumes complete recourse. That is, the assumption is made that for any first-stage decision $\boldsymbol{x}$, and any possible realization of the uncertainty, $\boldsymbol{\omega}$, there exists some second stage decision $\boldsymbol{y}$ such that the pair $(\boldsymbol{x}, \boldsymbol{y})$ is feasible for the particular realization, $\boldsymbol{\omega}$. Therefore the issue becomes one of minimizing the expected cost, and thus a typical formulation is:

$$\begin{array}{rl} \min: & f(\boldsymbol{x}) \stackrel{\triangle}{=} \mathbb{E}_{\boldsymbol{\omega}}[F(\boldsymbol{x}, \boldsymbol{\omega})] \\ \text{s.t.}: & \boldsymbol{x} \in \mathcal{X}. \end{array}$$

Analytic solutions require us to be able to express the objective function $f(\boldsymbol{x})$ in closed form. This is rarely possible. A common way out, then, is to approximate the expectation via some version of Monte Carlo sampling (see, e.g., [117], [88]). In [119], Shapiro considers the sample average approximation method, and shows that the "total number of scenarios needed to solve the true problem with a reasonable accuracy grows *exponentially* with increase of the number of stages." In fact, even for a constant number of stages, the bounds he gives for the sample complexity given a particular error parameter ε and reliability parameter δ are not necessarily polynomial in the description length of the problem. This is because his bounds involve a parameter capturing the variability of the function F, and this may not be polynomial. Recently, in [126] the authors showed that for a fairly broad class of multi-stage problems, the Sample Average Approximation algorithm converges in polynomial time for any fixed number of stages. Nevertheless, the sample complexity does grow exponentially in the number of stages of the problem.[1]

Indeed, the fundamental issue with respect to multi-stage problems, appears to be how we can evaluate the quality (or even feasibility, for problems without complete recourse) of a first stage solution $\boldsymbol{x}$, in a K-stage problem. In a two-stage problem, a straightforward approach might be to simply sample realizations $\boldsymbol{\omega}^{(i)}$ of the uncertainty, and for each $\boldsymbol{\omega}^{(i)}$ compute a second stage variable $\boldsymbol{y}^{(i)}$ to optimally complete the (fixed) first-stage solution $\boldsymbol{x}$. Consider, however, a three-stage problem:

$$\begin{array}{rl} \min: & \boldsymbol{c}^{\top}\boldsymbol{x} + \boldsymbol{y}(\omega_1) + \boldsymbol{v}(\omega_1, \omega_2) \\ \text{s.t.}: & \boldsymbol{A}(\omega_1, \omega_2)\boldsymbol{x} + \boldsymbol{B}(\omega_1, \omega_2)\boldsymbol{y}(\omega_1) + \boldsymbol{C}(\omega_1, \omega_2)\boldsymbol{v}(\omega_1, \omega_2) \leq \boldsymbol{b}. \end{array}$$

[1] See Chapter 2 for further discussion of sampling results for Stochastic Optimization, and integration via Monte Carlo sampling.

Given x, we cannot simply generate a collection of samples $\{(\omega_1^{(1)}, \omega_2^{(1)}), \ldots, (\omega_1^{(N)}, \omega_2^{(N)})\}$ and seek a pair $(y^{(i)}, v^{(i)})$ for each $(\omega_1^{(i)}, \omega_2^{(i)})$. Indeed, such an approach does not preserve the causality structure of the three-stage problem, as now both y and v are effectively functions of the second and third stage uncertainty realizations. Essentially the result of Shapiro shows that this problem forces an exponential explosion in the number of samples required, since for each realization of the second stage uncertainty, ω_1, we must take an independent collection of samples of the third stage uncertainty, ω_2.

4.2.3 Chance Constraints and Sampling

Robust Optimization provides tractable results, but only for a limited class of convex problems. Intractability stems in part from the worst-case view, which creates the inner problem that proves to be the (computational resource) bottleneck. Stochastic Optimization also presents complexity issues, requiring the solution of challenging nonconvex constraints, and difficult-to-compute integrations. The difficult integrations stem directly from an attempt to compute expected values of the cost-to-go functions, to use language of dynamic programming. There is a middle road, however, that avoids both the tractability pitfalls of worst-case interpretation of constraints, and expected value performance measure.

Chance constraints require the uncertain constraints to be satisfied with some (high) probability at least $(1-\varepsilon)$. It has long been observed in stochastic programming, that a solution that is feasible with high probability, can dramatically outperform a solution that is required to be feasible with probability one. This is a primary motivation for the so-called chance constraint, where given a stochastic constraints $f(x, \omega) \leq 0$, we write

$$\mathbb{P}(f(x, \omega) \leq 0) \geq 1 - \varepsilon.$$

Such constraints are typically nonconvex, and thus difficult to deal with directly ([104],[107]) except for some special classes of distribution, and form of the constraints. This is discussed further in Chapter 2.

Analogously to the Monte Carlo approach to integration, Monte Carlo sampling again becomes of interest here. The main focus, however, is feasibility, as opposed to an expectation minimization, as in the standard stochastic programming setup. Feasibility evaluation essentially amounts to integration of an indicator function. For this reason, variability parameters of the integrand – something that, as discussed above, has presented difficulties for sample-based multi-stage stochastic optimization approximations – does not appear to be an issue.

The basic approach is to consider the sampled robust problem (SRP), where the chance constraint is replaced by the sampled version of that constraint, and so we replace the true feasible set

$$\mathcal{X}_\varepsilon \triangleq \{x \mid \mathbb{P}(f(x,\omega) \leq 0) \geq 1-\varepsilon\},$$

by the sampled feasible set

$$\mathcal{X}_N = \left\{ x : \begin{array}{c} f(x,\omega^{(1)}) \leq 0, \\ \vdots \\ f(x,\omega^{(N)}) \leq 0 \end{array} \right\}.$$

The central question related to the sampling approach, is to understand the sample complexity, i.e., the number of samples N required, so that an element $x \in \mathcal{X}_N$, is in $\mathcal{X}_\varepsilon$ with at least some reliability $(1-\delta)$.

This approach relaxes the notion of robust feasibility, requiring the optimal solution to be feasible only to the finite collection of samples seen. Furthermore, the optimality criterion is not one of expected value, but rather a worst-case criterion. Thus we can think of this as a relaxation of the robust framework.

We are particularly interested in the work of Calafiore and Campi ([43]) and de Farias and Van Roy ([51]). The authors assume that the uncertain data are generated by some (possibly unknown) distribution. Subject to certain convexity constraints, they show that if one samples $N(\delta,\varepsilon)$ constraints, then with probability at least $(1-\delta)$, the solution x^* to the resulting deterministic problem will be feasible to the next sampled constraint with probability at least $(1-\varepsilon)$. The main result of [43] is to show that choosing

$$N(\delta,\varepsilon) \geq 2[n\frac{1}{\varepsilon}\ln\frac{1}{\varepsilon} + \frac{1}{\varepsilon}\ln\frac{1}{\delta} + n], \tag{4.2.3}$$

is sufficient to obtain the above reliability and feasibility guarantees. The result of de Farias and Van Roy ([51]) is similar, although it relies on results from learning theory, rather than convexity properties of $\mathbb{R}^n$, as is discussed in detail in the sequel.

The sample complexity results have been specialized, and the sample complexity greatly improved for particular special classes of problems, in [92] and [62]. There, they consider linear optimization problems with affine dependence on the uncertainty parameters, and subject to some additional conditions, they prove sample complexity bounds independent of the dimension, and logarithmic in ε^{-1}. However, such an approach is restricted to affine dependence on the uncertainty, and in particular cannot handle problems such as the two-stage example given above. As we see below, higher order models for continuous and discrete adaptability can be reformulated as linear optimization problems where the parameters depend in a nonconvex (and in the integer case noncontinuous) manner on the uncertainty. Therefore the results of [92] and [62] do not apply to our situation.

Moreover, in [92] and [62], the authors apply their results to two stage problems, but because of the affineness requirements, they only consider two-stage problems with constant recourse. This is the case when the second stage matrix, B in our notation, is constant and known in advance. In this chapter we consider the more general setting where the recourse matrix is not fixed.

■ 4.2.4 Other Directions

The work done in Stochastic Optimization goes far beyond the works that we are able to mention here. We bring up only the modern versions of the results that are most important to contextualize and support the work in this chapter. One additional recent advance that is worth noting and commenting on, is that of Dean, Goemans, and Vondrák (see, e.g., [52],[72]). In [52], the authors consider the stochastic knapsack problem with deterministic item values, but sizes that are instantiated only after the item is placed in the knapsack. Remarkably, they are able to show that there exists a nonadaptive algorithm that performs within a factor of four of the optimal adaptive algorithm. This work differs in primarily two ways from what we consider here. First, feasibility is the main focus of our work, and thus the constraints are treated as hard, and cannot be violated. As is typical in optimization, violation incurs an infinite cost. In contrast, in the knapsack formulation, once the knapsack is full or overflows, the game ends, and the optimizer retains the current value of the knapsack. The bounds they obtain would not be possible in a more restrictive framework. In addition to this, the main focus there is on the gap between the adaptive policies, and entirely non-adaptive policies.

For many applications, however, a receding horizon approach is in fact practical, and in many cases such approaches have been used with very favorable results. We are here most interested in the comparison of the quality of only the first-stage solution obtained through different levels of adaptability. It is unclear from the analysis in [52] and [72], that there even is a gap between the receding-horizon non-adaptive policy, and the optimal adaptive policy.

Indeed, the difficult aspect of adaptability is capturing the fact that we have adaptability in future stages, in our computation for an optimal, or good, first-stage decision.

■ 4.2.5 Outline and Contributions

The central topic of this chapter is to use sampling techniques to structure adaptability, in multi-stage, and in particular in two-stage optimization problems.

We take advantage of the fact that by sampling constraints, one entirely side-steps any difficulty with non-linear dependence of the constraints on the uncertain parameters (as further explained below). The only requirement is that we are able to sample from the distribution. The price to pay, of course, is that rather than deterministic feasibility to the uncertainty, we have to accept a reliability parameter δ, and a probability of infeasibility ε.

Specifically, our contributions in this chapter and the chapter outline are as follows:

(1) Finite Adaptability: In [22] we introduced a finite hierarchical model for adaptability, termed Finite Adaptability, that provided a geometric approach to structuring piecewise constant second-stage solutions. This proposal presented a hierarchy of adaptability, and because of its inherent finiteness, was able to accommodate discrete variables in the second stage. The approach was limited to polyhedral uncertainty sets specified by their extreme points. Here we show that by sampling the uncertainty set, we can greatly increase the class of uncertainty sets that the finite adaptability approach can address. This is the subject of Section 4.3.1.

(2) Nonlinear and Higher Order Adaptability: In [12], the authors propose the notion of affine adaptability for multi-stage linear optimization problems. As discussed in Chapter 2, the resulting problem is NP-hard, but they use some past results ([17]) to develop approximate approaches via Semidefinite Optimization. The difficulty they face is due entirely to the fact that if the second stage variable is an affine function of the uncertainty, the resulting robust optimization problem takes on a quadratic dependence on the uncertainty. This results in an intractable inner problem (see Chapter 2). By using the sampling techniques of [43] we are able to circumvent this problem. This allows us to structure not only affine, but also nonlinear models of adaptability, such as quadratic and, if we care to do so, higher order polynomial adaptability, all by solving deterministic linear optimization problems. This involves a *feature* map, that maps the uncertainty in potentially a nonlinear manner, to a possibly higher dimensional space. Affine and polynomial adaptability are just two instances of this general procedure, and in fact Finite Adaptability introduced in Chapter 3 can also be viewed in this light. This gives us an important outlet when affine adaptability fails. As illustrated in [22], even for simple and low-dimensional linear optimization problems, the affine adaptability paradigm may be no better than the original robust approach. In Section 4.3.2 we give the proposed sampled approach, along with computational examples of its effectiveness.

(3) Integer Adaptability: A significant shortcoming of the affine adaptability framework of [12], and indeed any continuous-adaptability proposal, is its inability to address any problem with integer constraints on the second-stage variables. In such a case, the adaptability must necessarily be piecewise constant. The finite adaptability proposal of [22] had such a setup in mind, and developed a framework for piecewise constant adaptability with a very small number of pieces. The focus there was on the explicit construction of the partition of the uncertainty set, into a small number of pieces. Here, we take a different approach

made possible by sampling the uncertainty. Rather than structuring the pieces of the adaptability explicitly, we develop a new proposal that allows for an exponential number of pieces to be implicitly constructed.

Along the way, we prove an extension to the results of [43], to the case of convex sampled constraints, and nonconvex, specifically integer, constraints imposed in addition.

The framework for the integer adaptability, as well as the proof of this result, is the subject of Section 4.3.3.

(4) Multistage Optimization Under Uncertainty: We consider here extensions to the case of many stages. Here, sampling approaches in the past have proven ineffective, as the schemes proposed require a number of samples that grows exponentially in the number of stages. We circumvent this problem by restricting ourselves to structured adaptability functions, and then using the sampling ideas of [43]. To the best of our knowledge, this is the first proposal that offers a hierarchy of higher order adaptability, that does not require complete recourse, and furthermore is applicable to the multi-stage optimization, yet does not suffer an exponential explosion of the complexity, in the number of stages.

(5) In Section 4.5, we consider the reliability, and probability of feasibility, uniformly over the set of all points that are feasible for the sampled problem. We introduce a notion of robustness into the sampling procedure. This allows us to avoid relying on the VC-dimension to obtain bounds on the sample complexity. It allows us to use much finer complexity measures, such as covering numbers, and also fat shattering dimension. The finer complexity measures can be particularly useful in the context of the rich class of adaptability introduced in Section 4.3.2.

In addition, the new notion of robustness in the samples, allows us to introduce a third parameter into the sample complexity. Thus, rather than sample complexity that is only a function of the reliability, δ, and the feasibility level ε, we now are able to trade-off with a robustness parameter η. This further allows us to control the growth of the number of samples we require to obtain feasibility and reliability guarantees.

(6) Finally, in Section 4.6, we offer more extensive computational examples of our approach, and we compare it with other proposals for adaptability. We consider two specific applications. First, in Section 4.6.1, we consider a network design problem. We use this to not only compare the benefit in terms of cost of adaptability over the static robust formulation, but in addition we formulate the problem of maximizing robustness subject to resource constraints, and here too we demonstrate the advantages of using adaptability. Then in Section 4.6.2, we con-

sider a multistage portfolio optimization considered by Ben-Tal, Margalit, and Nemirovski ([13]).

4.3 A Proposal: Sampling Structured Adaptability

In this section we propose a sample-based approach to structured adaptability. Both the sampling aspect, and the structured nature of the proposal, are important to our results.

The sampling itself, accomplishes a simplification of the uncertainty set. We show in Section 4.3.1 that while the convex hull of the sampled points may make up only an exponentially small fraction of the volume, nevertheless we maintain strong feasibility guarantees, and in addition, the central benefit is that we work with a much more tractable uncertainty set description, namely, one given by extreme points. In Section 4.3.1, we show that such a description is crucial to the efficient computation of certain parameters, in particular, the worst-case uncertainty realization point of a single-stage robust optimization problem. We use this observation to broaden the scope and applicability of finite adaptability, as considered in Chapter 3.

Even more significantly, sampling uncertainty realizations eliminates non-linearities in the way the parameter uncertainty enters into the problem. By sampling a finite set of realizations of the uncertainty parameter, ω, we explicitly include each sampled constraint individually in the optimization. Therefore a potentially non-linear uncertain inequality $f(x,\omega) \leq 0$ which must be satisfied for all realizations of ω in a continuous set Ω, we deal with individual constraints $f(x,\omega^{(i)}) \leq 0$. Thus effectively, the impact of the uncertainty becomes affine. As usual for sampling methods, this simplification comes at the expense of a nonzero probability of error, and of infeasibility; that is, we introduce the reliability and feasibility parameters (ε, δ); another expense, naturally, is the increase in the number of inequalities. Bounding this increase is of central importance.

This sampling approach allows us to consider higher order and non-linear adaptability models. In turn, this structure we introduce, allows us to control the number of variables, and reduce multistage problems into single stage problems with only polynomially many variables. This should be contrasted to the exponential growth of the number of variables and samples required, in other multistage approaches (see [119], [120]).

Since this reduces the problem to an explicit finite dimensional single stage problem, we are able to apply other tools we have developed. In particular, employing ideas of finite adaptability, we are able to structure piecewise polynomial adaptability.

In addition to simplifying polynomial nonlinearities and allowing us to structure higher order continuous adaptability models, the same approach allows us to deal with much more nonlinear functions: rounding functions. This observation allows us to implement rounding functions, and thus structure adaptability with integral second (and later) stage variables, by solving integer linear optimization problems. This is the topic of Section 4.3.3.

4.3.1 Finite Adaptability

For convex robust optimization problems with component-wise uncertainty, it is well-known that there is a single realization of the uncertainty, $\boldsymbol{\omega}^* \in \Omega$, such that the nominal problem with realization $\boldsymbol{\omega}^*$ is equivalent to the full robust problem, in the sense that the robust solution is feasible and optimal for both. When the uncertainty is not component-wise, then this point $\boldsymbol{\omega}^*$ may not lie in Ω, but it does lie in the smallest hypercube containing Ω, (this is pursued in greater detail in Chapter 3, and [22]). Let Ω_i denote the projection of Ω onto the components corresponding to the i^{th} constraint, that is, the components that affect the i^{th} constraint of the optimization. Then the hypercube is defined as

$$(\Omega)_R \overset{\triangle}{=} \Omega_1 \times \cdots \times \Omega_m,$$

where m is the number of constraints.

Computing these points $\boldsymbol{\omega}^* \in (\Omega)_R$ is important for understanding why the static solution (i.e., where $\boldsymbol{y}(\boldsymbol{\omega}) = \boldsymbol{y}$ has no adaptability to the realized uncertainty) is conservative with respect to the optimal adaptable solution. In particular, it is very important for obtaining guidance in structuring finite adaptability, in the sense of the content of Chapter 3.

Computing these points $\boldsymbol{\omega}^*$ is typically difficult. Indeed, it is at least as difficult as solving an inverse optimization subject to an additional feasibility constraint, but where we are inverting with respect to the matrix:

1. Given vectors $\boldsymbol{c}, \boldsymbol{b}$, the optimal target point $\boldsymbol{x}^*$, and a set of matrices $\{\boldsymbol{A}(\boldsymbol{\omega}) \ : \ \boldsymbol{\omega} \in (\Omega)_R\}$;

2. Find $\boldsymbol{\omega}^* \in (\Omega)_R$ such that:

$$\begin{array}{rl} \min: & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{s.t.}: & \boldsymbol{A}(\boldsymbol{\omega}^*)\boldsymbol{x} \leq \boldsymbol{b}, \end{array}$$

   has optimal solution $\boldsymbol{x}^*$.

Typically, inverse optimization problems are solved by considering the dual problem, and the optimality conditions for linear optimization. For inverse optimization problems with respect to the objective function (or the right hand side vector) this becomes a simple feasibility problem. In the case of matrix inversion, however, the problem becomes difficult, because of the bilinearity of the resulting feasibility problem (we must simultaneously search for a matrix $\boldsymbol{A}(\boldsymbol{\omega})$, and also for a dual variable that appears in a product with $A(\omega)$. As the following example shows, the resulting bilinear problem cannot be easily solved, as the feasible set of points ω^* for which the given solution x^* is indeed optimal, may be nonconvex.

An Example: Inverse set may be nonconvex:

We consider the example (4.2.2). It is not difficult to show that, according to the definition above, the two matrices:

$$D_1 = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ \frac{7}{20} & \frac{7}{20} & 0 \end{pmatrix} \qquad D_2 = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 1 \\ \frac{7}{20} & \frac{7}{20} & 0 \end{pmatrix},$$

both satisfy $D_1, D_2 \in (\Omega)_R$, and the point $x^* = (10/7, 10/7, 1)$ is optimal for both nominal problems. However the nominal problem with a matrix from the convex hull of $\{D_1, D_2\}$ need not have x^* as an optimal solution. For instance, $D = (D_1 + D_2)/2$ has optimal solution $x = (10/7, 10/7, 0)$. ■

However, the problem of computing $\omega^* \in (\Omega)_R$ is tractable (and straightforward) when Ω is given as a convex hull of extreme points.

The sampling approach gives us a technique to compute some of the points ω^*, and thus allows us to extend the partitioning tools of Chapter 3 (and [22]) to more general uncertainty sets, including, for instance, norm-bounded sets.

■ 4.3.2 Nonlinear Adaptability: Feature Space

Here we use sampling to structure higher order adaptability, and also to consider piecewise-continuous models for adaptability. The general robust adaptable problem is:

$$\begin{aligned} \min: \quad & c^\top x \\ \text{s.t.}: \quad & A(\omega)x + B(\omega)y(\omega) \leq b, \quad \forall \omega \in \Omega. \end{aligned}$$

Sampling and Higher Order Adaptability

In the affine adaptability framework of [12], the second stage variable y is allowed an affine dependence on the uncertainty ω, thus we have:

$$y(\omega) = Q\omega + q.$$

Thus the affine adaptability problem becomes:

$$\begin{aligned} \min: \quad & c^\top x \\ \text{s.t.}: \quad & A(\omega)x + B(\omega)\left[Q\omega + q\right] \leq b, \quad \forall \omega \in \Omega. \end{aligned} \tag{4.3.4}$$

This is now a single stage linear optimization problem with uncertain parameters, where the uncertainty takes values in precisely the same set Ω as in the original problem. The decision variables are now $(\boldsymbol{x}, \boldsymbol{Q}, \boldsymbol{q})$. The fundamental difference with the single stage problems solved in [15], and [26], is that the uncertainty, while it takes values in the original uncertainty set Ω, affects the problem parameters in a nonlinear (quadratic, here) manner. This takes us outside the realm where the tractability results of Robust Optimization hold. Indeed, as shown in [12], in general this problem is NP-hard. For higher order adaptability, for instance, quadratic adaptability, the problem is further exacerbated. In the quadratic adaptability case, we have:

$$\boldsymbol{y}(\omega) = \boldsymbol{Q}_2 \boldsymbol{w} + \boldsymbol{Q}_1 \omega + \boldsymbol{q},$$

where $\boldsymbol{w}$ is the vector of all pairwise products $u_i u_j$. Then the resulting problem is a single stage robust linear optimization, but where now the parameters have a cubic dependence on the uncertainty. In [12], the authors propose an SDP approximation scheme for affine adaptability, using an approximate S-lemma proved in [17]. For quadratic and higher order adaptability, there is not even a proposal that attempts to address this problem.

The theme of this chapter is that if we are willing to accept reliability probability $(1-\delta)$, and feasibility probability $(1-\varepsilon)$, then we can effectively circumvent the computational intractability arising from nonlinear dependence of the parameters on the uncertainty.

Thus, for affine adaptability, rather than attempting to solve a hard subproblem exactly, or solving it approximately via SDP (as in [12]), instead we sample Ω according to the distribution $\mathbb{P}$ N times independently, to obtain samples $\{\omega^{(1)}, \ldots, \omega^{(N)}\}$, and then solve the sampled robust problem:

$$(\boldsymbol{SRP}^N) \left\{ \begin{array}{ll} \min: & \boldsymbol{c}^{\mathrm{T}} \boldsymbol{x} \\ \text{s.t.}: & \boldsymbol{A}(\omega^{(1)})\boldsymbol{x} + \boldsymbol{B}(\omega^{(1)}) \left[\boldsymbol{Q}\omega^{(1)} + \boldsymbol{q}\right] \leq \boldsymbol{b} \\ & \vdots \\ & \boldsymbol{A}(\omega^{(N)})\boldsymbol{x} + \boldsymbol{B}(\omega^{(N)}) \left[\boldsymbol{Q}\omega^{(N)} + \boldsymbol{q}\right] \leq \boldsymbol{b} \end{array} \right\}.$$

Letting $\boldsymbol{w}_r$ denote the vector of all r-fold products of the $\{u_i\}$ (so that, e.g., $\boldsymbol{w}_1 = \omega$), this procedure allows us to structure higher order adaptability in precisely the same manner. Thus, for degree-r-adaptability, we have adaptability function:

$$\boldsymbol{y}(\omega) = \boldsymbol{Q}_r \boldsymbol{w}_r + \boldsymbol{Q}_{r-1} \boldsymbol{w}_{r-1} + \cdots + \boldsymbol{Q}_2 \boldsymbol{w}_2 + \boldsymbol{Q}_1 \omega + \boldsymbol{q}.$$

Then, the sampled robust problem is again simply an LP, where the variables are $\boldsymbol{x}$

and the coefficient matrices Q_i, q:

$$(SRP^N) \left\{ \begin{array}{ll} \min: & c^\top x \\ \text{s.t.}: & A(\omega^{(1)})x + B(\omega^{(1)}) \left[\sum_{i=2}^r Q_i w_i^{(1)} + Q_1\omega^{(1)} + q\right] \le b \\ & \vdots \\ & A(\omega^{(N)})x + B(\omega^{(N)}) \left[\sum_{i=2}^r Q_i w_i^{(N)} + Q_1\omega^{(N)} + q\right] \le b \end{array} \right\}.$$

Indeed, by the linearity of polynomials in their coefficients, we have the straightforward result:

Proposition 4.1
We can obtain the optimal adaptability function of order r, for the sampled robust LP, by solving a linear optimization problem with polynomially increased number of variables.

In Section 4.4 we show that given any reliability parameter $\delta > 0$, and feasibility parameter $\varepsilon > 0$, the sample complexity $N(\delta, \varepsilon)$ required to guarantee feasibility within the given parameters, of the sampled solution with degree r adaptability, is polynomial in ε and δ. Furthermore, for the two-stage problem, this sample complexity is independent of the degree r of the adaptability.

A Nonlinear View

The central fact that we exploit is that using samples frees us from dealing with nonlinearities in the uncertainty parameter ω. Indeed, this, along with the fact that our adaptability is ultimately linear in the decision variables, allows us to structure nonlinear adaptability by solving linear optimization problems. The higher order polynomial adaptability above illustrates this principle, but we can view such adaptability in a more general light. Consider again the sampled robust optimization problem, given samples $\Omega_N = \{\omega^{(1)}, \dots, \omega^{(N)}\}$:

$$\begin{array}{ll} \min: & c^\top x \\ \text{s.t.}: & A(\omega^{(1)})x + B(\omega^{(1)})y(\omega^{(1)}) \le b \\ & \vdots \\ & A(\omega^{(N)})x + B(\omega^{(N)})y(\omega^{(N)}) \le b, \end{array}$$

In this formulation, the adaptable function y is some function: $y : \mathbb{R}^m \to \mathbb{R}^{n_2}$.

We propose the following approach to structuring nonlinear adaptability for multistage optimization. Suppose we have $\omega \in \mathbb{R}^m$ for some m. Now consider any function

$$F : \mathbb{R}^m \to \mathbb{R}^r.$$

The only requirement we place on F is that it is, in some appropriate sense, easy to compute. Note that the dimension of the image space, r, may be larger or smaller than m. Now consider the optimization problem:

$$\begin{array}{rl} \min: & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{s.t.}: & \boldsymbol{A}(\boldsymbol{\omega}^{(1)})\boldsymbol{x} + \boldsymbol{B}(\boldsymbol{\omega}^{(1)})\boldsymbol{y}(F(\boldsymbol{\omega}^{(1)})) \leq \boldsymbol{b} \\ & \vdots \\ & \boldsymbol{A}(\boldsymbol{\omega}^{(N)})\boldsymbol{x} + \boldsymbol{B}(\boldsymbol{\omega}^{(N)})\boldsymbol{y}(F(\boldsymbol{\omega}^{(N)})) \leq \boldsymbol{b}. \end{array}$$

Now, the second stage function $\boldsymbol{y}$ is a map: $\boldsymbol{y} : \mathbb{R}^r \rightarrow \mathbb{R}^{n_2}$. If we restrict $\boldsymbol{y}$ to be an affine map, then the sampled problem becomes a linear optimization problem. A few comments are in order.

1. For affine adaptability, the function F is the identity.

2. For polynomial adaptability, the map F is the nonlinear moment map, mapping from a vector $\boldsymbol{\omega} = (\omega_1, \ldots, \omega_m)$ to the vector of monomials of bounded degree: $\boldsymbol{\omega} \mapsto (\boldsymbol{\omega}^\alpha)_{|\alpha| \leq d}$.

3. Finite adaptability as well can be viewed in this light: given a partition $\Omega = \Omega_1 \cup \Omega_2$, of the uncertainty set, the function F becomes a piecewise constant function on that partition. Note that in this Chapter 3, the effort is to perform an optimization over a restricted class of such functions F. Here, on the other hand, F is fixed, and we optimize with respect to the affine function $\boldsymbol{y}(\cdot)$.

To use the language of approximate dynamic programming, the mapping F can be thought of as mapping to a feature space. Choosing a good mapping F, i.e., choosing the right features, is an important and difficult task. As mentioned above, Chapter 3 addresses this problem in a very limited context. This is essentially the issue of choosing a proper model for the adaptability. In addition to providing a hierarchy of adaptability, the flexibility of this approach offers the opportunity to capture additional outside information the decision-maker may have about the problem. That is, this is not the issue of choosing what degree polynomial to use, but rather of much richer question of what class of nonlinear feature functions F are appropriate for the problem at hand. In particular when faced with the choice of several feature functions $F \in \mathcal{F}$, sample complexity bounds becomes very important. In Sections 4.4 and 4.5, we discuss sample complexity. Particularly in Section 4.5, we consider sample complexity bounds related directly to the complexity of the class of adaptability functions. In the scheme proposed above, where $\boldsymbol{y}$ is restricted to be affine on the image of F, it is the feature function F that determines the complexity of the class of the adaptability functions, and hence controls the bounds on sample complexity. The possibility of also "training" and performing some search of a limited class of functions is also of interest. This line of research is pursued elsewhere (but see Chapter 6).

The extension to the multi-stage case is straightforward. Given K-stage uncertainty, $\boldsymbol{z} = (\boldsymbol{\omega}_1, \dots, \boldsymbol{\omega}_K) \in \mathbb{R}^{m_1+\cdots+m_K}$, and $\boldsymbol{z}_k = (\boldsymbol{\omega}_1, \dots, \boldsymbol{\omega}_k) \in \mathbb{R}^{m_1+\cdots+m_k}$, and a K-stage problem

$$\begin{array}{rl} \min: & \boldsymbol{c}^\top \boldsymbol{x} + \sum_{i=1}^K \boldsymbol{d}^\top \boldsymbol{y}_i(\boldsymbol{z}_i) \\ \text{s.t.}: & \boldsymbol{A}(\boldsymbol{z})\boldsymbol{x} + \sum_{i=1}^K \boldsymbol{A}_i(\boldsymbol{z})\boldsymbol{y}_i(\boldsymbol{z}_i) \le \boldsymbol{b}, \end{array}$$

then the adaptability is defined by K nonlinear functions:

$$\begin{array}{c} F_1 : \mathbb{R}^{m_1} \longrightarrow \mathbb{R}^{r_1} \\ F_2 : \mathbb{R}^{m_1+m_2} \longrightarrow \mathbb{R}^{r_2} \\ \vdots \\ F_K : \mathbb{R}^{m_1+\cdots+m_K} \longrightarrow \mathbb{R}^{r_K}. \end{array}$$

Then we show that the sample complexity will be polynomial in the numbers $(r_1, \dots, r_K)$. That is, the dimension of the mappings F_i effectively determines the sample complexity. In particular, if F_i are given by the moment maps of fixed degree d, then the dimensions r_k grow in a polynomial fashion, and hence the degree k adaptability has polynomial sample complexity in the number of stages to the problem. In Section 4.5, we propose a modified sampled problem that allows us to give more sophisticated sample complexity bounds that can be independent of the dimension of the mappings F_i, but instead depend on other regularity properties.

Piecewise-continuous Adaptability

Once we form the sampled problem, we have a robust optimization problem where the uncertainty set is given by extreme points; the extreme points are the sampled uncertainty points. Therefore we have the required structure for the results developed in Chapter 3. Applying the results of Chapter 3, we can obtain a partition of the uncertainty set Ω, and then structure affine, higher order, or other nonlinear adaptability on each region of the partition, thus yielding piecewise polynomial adaptability.

Some Illustrative Examples

We now consider a small example. Larger-scale examples are considered in Section 4.6.

$$\begin{array}{rl}
\min: & x \\
\text{s.t.}: & x \geq y_1 + y_2 + y_3 \\
 & \boldsymbol{B}\boldsymbol{y} \geq \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} \qquad \forall\ \boldsymbol{B} \in \operatorname{conv}\{\boldsymbol{B}^1, \boldsymbol{B}^2\} = \operatorname{conv}\left\{ \begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 1 \\ \frac{1}{2} & \frac{1}{5} & 0 \end{pmatrix}, \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ \frac{1}{5} & \frac{1}{2} & 0 \end{pmatrix} \right\} \\
 & y_1, y_2, y_3 \geq 0.
\end{array} \tag{4.3.5}$$

The static value is 27/7, obtained by the solution $x = 27/7$, $\boldsymbol{y} = (10/7, 10/7, 1)$. The optimal affine adaptable solution coincides with the static solution. By sampling uniformly from the unit interval $[0, 1]$ (the uncertainty set) we compute the sampled version of both affine adaptability, and also quadratic adaptability and higher order adaptability (up to sextic). We see from the plots in Figure 4-2 that the affine adaptability indeed approaches the static value, while quadratic adaptability is almost as good as the optimal adaptable value, and this further illustrates the quick convergence for this simple example. Cubic adaptability seems to perform similarly as quadratic adaptability, but by the time we get to quartic adaptability, the solution achieves the same value as the optimal adaptable solution. In Figure 4-4 we plot the optimal quadratic solutions in the figure on the left, and, to illustrate the hierarchy of adaptability, we plot the sextic solutions on the right (like the quartic solution, the sextic achieves the optimal cost).

We can also consider partitioning the set, and then employing a continuous adaptability scheme on each partition. For this simple example, while affine adaptability is unable to improve the solution, piecewise affine adaptability with only two pieces, is enough to obtain the optimal adaptable value. The graph of the piecewise constant solution is given in Figure 4-3.

The key point we illustrate here is that sampling allows us to work harder to obtain better solutions when affine adaptability fails. Previously, we could not resort to working harder in order to improve the adaptability.

■ 4.3.3 Integer Adaptability

In this section, we use the sampling techniques to address the situation where the second stage variables $\boldsymbol{y}(\boldsymbol{\omega})$ must be integral.

We prove a simple extension to the sample complexity result of Calafiore and Campi in Section 4.4.1. We apply that sampling result here, to the case of integer

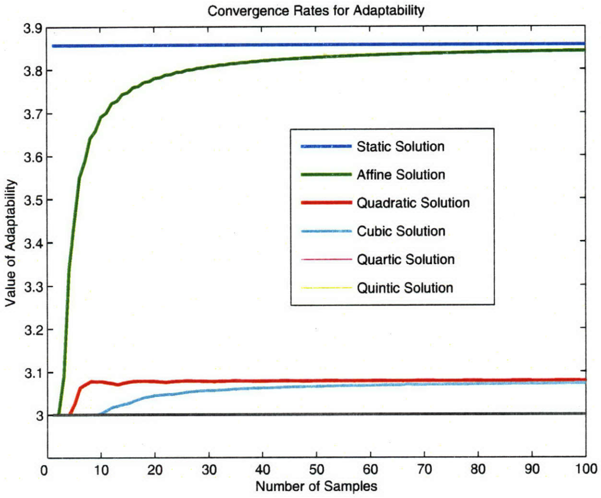

Figure 4-2. This figure illustrates the difference, in particular the improvement, between affine adaptability and quadratic adaptability, and also higher orders of adaptability, for the sampled version of the problem in Example 4.3.5. Note that beyond quartic adaptability, we recover the optimal solution.

adaptability for the second stage variables. We consider the problem:

$$\begin{aligned} \min: \quad & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{s.t.}: \quad & \boldsymbol{A}(\boldsymbol{\omega})\boldsymbol{x} + \boldsymbol{B}(\boldsymbol{\omega})\boldsymbol{y}(\boldsymbol{\omega}) \geq \boldsymbol{b}, \quad \forall \boldsymbol{\omega} \in \Omega, \\ & \boldsymbol{x} \in \mathcal{X}, \boldsymbol{y} \in \mathbb{Z}^n. \end{aligned}$$

Similar to the affine adaptability scheme with sampling that we use above, we now introduce the following integer adaptability scheme:

$$\boldsymbol{y}(\boldsymbol{\omega}) = \boldsymbol{Q}\lceil \boldsymbol{\omega} \rceil + \boldsymbol{q},$$

where $\lceil \boldsymbol{\omega} \rceil$ indicates the component-wise ceiling function, so for instance $\lceil (0.2, 2.1, 1.9) \rceil = (1, 3, 2)$. Then the sampled problem we solve is:

$$\begin{aligned} \min: \quad & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{s.t.}: \quad & \boldsymbol{A}(\boldsymbol{\omega}^{(1)})\boldsymbol{x} + \boldsymbol{B}(\boldsymbol{\omega}^{(1)}) \left[\boldsymbol{Q}\lceil \boldsymbol{\omega}^{(1)} \rceil + \boldsymbol{q} \right] \geq \boldsymbol{b} \\ & \quad \vdots \\ & \boldsymbol{A}(\boldsymbol{\omega}^{(N)})\boldsymbol{x} + \boldsymbol{B}(\boldsymbol{\omega}^{(N)}) \left[\boldsymbol{Q}\lceil \boldsymbol{\omega}^{(N)} \rceil + \boldsymbol{q} \right] \geq \boldsymbol{b} \\ & \boldsymbol{x} \in \mathcal{X} \\ & Q_{ij} \in \mathbb{Z}, \quad \forall i, j \\ & \boldsymbol{q} \in \mathbb{Z}^n. \end{aligned}$$

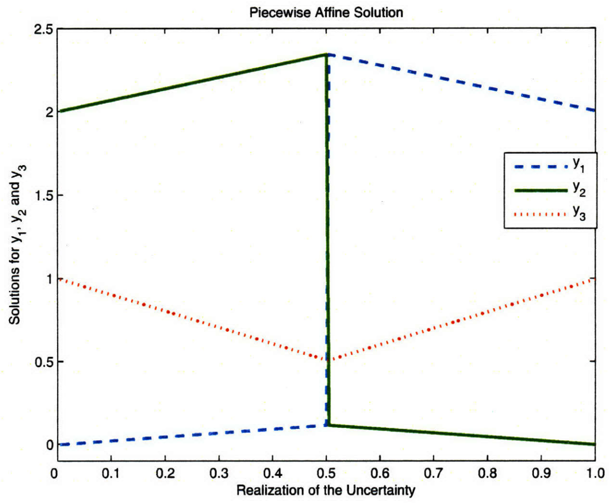

Figure 4-3. This figure gives the piecewise affine adaptable second stage solutions. The affine scheme alone is no better than the static robust. For polynomial adaptability, we need at least quartic adaptability to approach the optimal value. By partitioning into two regions, however, we achieve the optimal value with affine adaptability in each region of the partition. For higher dimensional problems, therefore, partitioning could lead to potentially large reductions in the number of variables. What constitutes a good partition for piecewise affine adaptability is a challenging issue that will be addressed elsewhere (see Chapter 6.

We note that as in the affine adaptability case, the number of variables has increased from $2n$ to $2n + n^2$. However the number of variables is fixed, and independent of the number of samples N. In particular, this means that we have a sampled linear integer problem, and therefore the results we prove below in Section 4.4.1 apply directly.

We remark that the ceiling function we use is highly nonlinear, and it would pose severe computational tractability issues were we not exploiting the power of sampling. Indeed, as stressed before, sampling the constraints allows us to circumvent any issues arising from the manner in which the uncertainty affects the problem parameters (as long as it is easily computable, which it certainly is here).

As in the results of Section 4.3.2, we can in a completely analogous way consider higher order adaptability.

Remark 4.1

We have used here the least-integer function to map from Ω to $\mathbb{Z}^m$. In fact we can use any function here. The key that allows us to obtain a computationally tractable scheme is that this function is fixed *a priori*, and then on top of that we chose the integer matrix $\boldsymbol{Q}$.

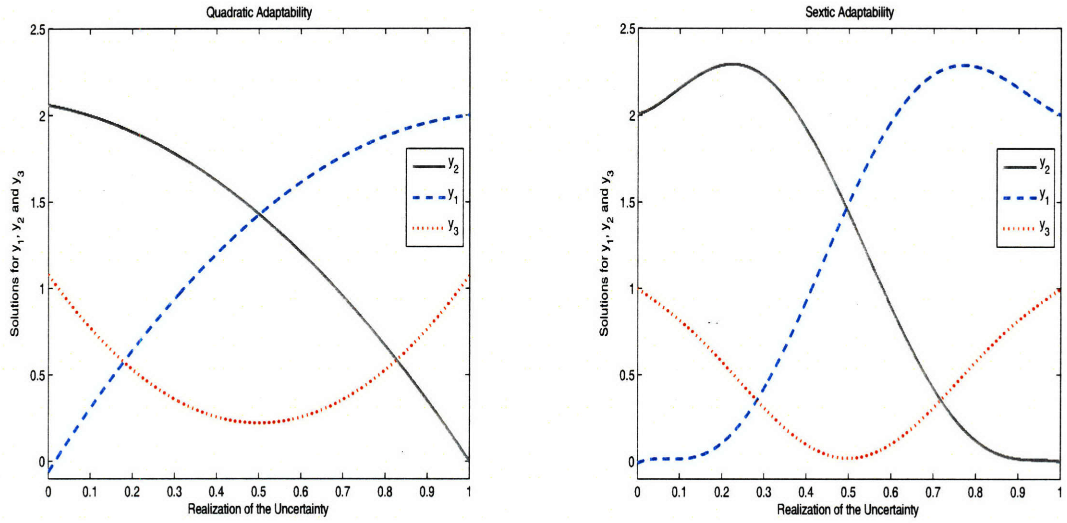

Figure 4-4. This figure plots the optimal quadratic and sextic adaptable solutions as a function of the realized uncertainty. The figure on the left represents the optimal quadratic solutions, and the figure on the right, the optimal sextic solutions. Comparing to Figure 4-1, we see how the adaptability here tries to match the nonlinear optimal adaptability shown there. Of course, for problems with higher dimensional uncertainty, sextic adaptability would require a very large number of additional variables, and thus would typically not be practical.

■ 4.4 Sample Complexity Results I

In this section, we focus on the sample complexity required to ensure feasibility of the optimal solution to the sampled robust problem. We consider the feasibility of the full multi-stage solution, and also the feasibility of the first-stage solution. The latter is the quantity of interest in the receding or folding horizon approach.

■ 4.4.1 Integer Extension to Calafiore and Campi

Here we consider the sample complexity of obtaining (ε, δ) guarantees for the case of integer variables, as in the framework introduced in Section 4.3.3.

First, we need to extend the results of Calafiore and Campi, to deal with the case of integer variables. We care about linear optimization problems with integer constraints:

$$\begin{aligned} \min: \quad & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{s.t.}: \quad & \mathbb{P}_{\boldsymbol{\omega}}(\boldsymbol{A}(\boldsymbol{\omega})\boldsymbol{x} \leq \boldsymbol{b}) \geq 1 - \varepsilon, \\ & \boldsymbol{x} \in \mathcal{X}. \end{aligned}$$

We are interested in the case where the deterministic set constraint $\boldsymbol{x} \in \mathcal{X}$, may include constraints such as $x_i \in \mathbb{N}$, or $x_i \in \mathbb{Z}$.

We note that the proof technique of [43] does not immediately extend to our situation. In the convex continuous case, the key part of the proof relies on showing that the number of support constraints is at most n. In the face of integer constraints, this may no longer be true. Indeed, the following example shows that we can have exponentially many support constraints.

Example: Let $\mathcal{X} = \{0,1\}^n$. Let the constraint set be finite, and consist of constraints $\boldsymbol{a}_\alpha \boldsymbol{x} \leq 1$, such that this constraint slices off corner $\alpha \in \{0,1\}^n$ off of the hypercube $[0,1]^n$. Consider now the problem:

$$\begin{aligned} \max: \;& x_1 + \cdots + x_n \\ \text{s.t.}: \;& \boldsymbol{a}_\alpha \boldsymbol{x} \leq 1 \\ & \boldsymbol{x} \in \mathcal{X}. \end{aligned}$$

Consider the situation where all $(2^n - 1)$ constraints $\boldsymbol{a}_\alpha$ for every $\alpha \in \{0,1\}^n \setminus \{\mathbf{0}\}$ have been sampled. Then, the optimal integer solution is $\boldsymbol{x}^*_{\text{int}} = \mathbf{0}$. If any one of the $2^n - 1$ constraints $\boldsymbol{a}_\alpha \boldsymbol{x} \leq 1$ is removed, the optimal solution becomes $\boldsymbol{x}^*_{\text{int}} = \alpha$. Therefore this problem indeed has $(2^n - 1)$ support constraints. ▲

Proposition 4.2

Given $\delta, \varepsilon > 0$. Consider the uncertain optimization problem whose robust formulation is:

$$(\boldsymbol{RP}) \quad \left\{ \begin{array}{rl} \min: & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{s.t.}: & \boldsymbol{A}(\boldsymbol{\omega})\boldsymbol{x} \leq \boldsymbol{b}, \quad \boldsymbol{\omega} \in \Omega \\ & \boldsymbol{x} \in \mathcal{X}. \end{array} \right\}$$

Let $\mathbb{P}$ be the distribution of $\boldsymbol{\omega}$ in Ω. Further, consider the sampled robust problem:

$$(\boldsymbol{SRP}^N) \quad \left\{ \begin{array}{rl} \min: & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{s.t.}: & \boldsymbol{A}(\boldsymbol{\omega}^{(1)})\boldsymbol{x} \leq \boldsymbol{b} \\ & \vdots \\ & \boldsymbol{A}(\boldsymbol{\omega}^{(N)})\boldsymbol{x} \leq \boldsymbol{b} \\ & \boldsymbol{x} \in \mathcal{X}. \end{array} \right\}$$

Let $N \geq N(\delta, \varepsilon)$, where this is the same sample complexity function of [43] as given above in Eq. (4.2.3). If $\boldsymbol{x}^*_{\text{int}}$ is an optimal (integer) solution to $\boldsymbol{SRP}^N$, then with probability at least $(1 - 2\delta)$, it satisfies

$$\mathbb{P}_{\boldsymbol{\omega}}(\boldsymbol{A}(\boldsymbol{\omega})\boldsymbol{x}^*_{\text{int}} \leq \boldsymbol{b}) \geq 1 - 2\varepsilon.$$

PROOF. While we cannot mimic the proof itself of [43] due to the possibly exponential number of support constraints, we can in fact use the result. Let $\mathcal{X}_0^N$ denote the feasible set of the sampled problem *without* the constraints $\mathcal{X}$, so that the true feasible set is $\mathcal{X}_0^N \cap \mathcal{X}$. The set $\mathcal{X}_0^N$ is polyhedral, and in particular, it is convex. Now let $\boldsymbol{x}^*_{\text{int}}$ be an optimal integer solution to the sampled problem $\boldsymbol{SRP}^N$. If this happens to be an extreme point of $\mathcal{X}_0^N$, then the results of [43] apply directly and we are done. If not, then there must be two points $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \mathcal{X}_0^N$ that are extreme points, and $\lambda \in (0,1)$, such that $\boldsymbol{x}^*_{\text{int}} = \lambda \boldsymbol{x}_1 + (1-\lambda)\boldsymbol{x}_2$. Now, by direct application of the results of [43], with probability at least $(1-\delta)$,

$$\mathbb{P}_\omega(\boldsymbol{A}(\omega)\boldsymbol{x}_1 \leq \boldsymbol{b}) \geq 1 - \varepsilon,$$

and also with probability $(1-\delta)$,

$$\mathbb{P}_\omega(\boldsymbol{A}(\omega)\boldsymbol{x}_2 \leq \boldsymbol{b}) \geq 1 - \varepsilon.$$

If $\boldsymbol{x}^*_{\text{int}}$ is infeasible for some constraint, then by convexity, at least one of $\boldsymbol{x}_1, \boldsymbol{x}_2$ must also be infeasible. Therefore the following inclusion holds:

$$\{\omega \in \Omega \, : \, \boldsymbol{A}_\omega \boldsymbol{x}^*_{\text{int}} \not\leq \boldsymbol{b}\} \subseteq \{\omega \in \Omega \, : \, \boldsymbol{A}_\omega \boldsymbol{x}_1 \not\leq \boldsymbol{b}\} \cup \{\omega \in \Omega \, : \, \boldsymbol{A}_\omega \boldsymbol{x}_2 \not\leq \boldsymbol{b}\},$$

and therefore we have

$$\begin{aligned}
\mathbb{P}_\omega(\boldsymbol{A}_\omega \boldsymbol{x}^*_{\text{int}} \leq \boldsymbol{b}) &= 1 - \mathbb{P}_\omega(\boldsymbol{A}_\omega \boldsymbol{x}^*_{\text{int}} \not\leq \boldsymbol{b}) \\
&\geq 1 - \mathbb{P}(\boldsymbol{A}_\omega \boldsymbol{x}_1 \not\leq \boldsymbol{b} \text{ or } \boldsymbol{A}_\omega \boldsymbol{x}_2 \not\leq \boldsymbol{b}) \\
&\geq 1 - [\mathbb{P}(\boldsymbol{A}_\omega \boldsymbol{x}_1 \not\leq \boldsymbol{b}) + \mathbb{P}(\boldsymbol{A}_\omega \boldsymbol{x}_2 \not\leq \boldsymbol{b})] \\
&\geq 1 - 2\varepsilon,
\end{aligned}$$

by the union bound. □

Remark 4.2
The result we prove is much more general than simply the case where $\mathcal{X}$ contains integer constraints, but for our purposes this is all we require.

■ 4.4.2 Sample Complexity for Two-Stage Problems: Projection

In this section we consider the feasibility only of the first stage decision, $\boldsymbol{x}$. This is in contrast to what we do in the subsequent section on multistage problems. The motivation is the receding horizon formulation. There, we are interested only in the quality (feasibility) of the first-stage variables.

Theorem 4.3
Consider an arbitrary stochastically uncertain two-stage linear program:

$$\begin{array}{rl} \min: & \boldsymbol{c}^{\mathsf{T}}\boldsymbol{x} + \boldsymbol{d}^{\mathsf{T}}\boldsymbol{y}(\boldsymbol{\omega}) \\ \text{s.t.}: & \boldsymbol{A}(\boldsymbol{\omega})\boldsymbol{x} + \boldsymbol{B}(\boldsymbol{\omega})\boldsymbol{y}(\boldsymbol{\omega}) \leq \boldsymbol{b}. \end{array} \tag{4.4.6}$$

Let $\boldsymbol{y}(\cdot)$ be a second stage adaptability function drawn from an arbitrary class of functions, $\mathcal{Y}$. Then, given a collection of N samples $\Omega_N = \{\boldsymbol{\omega}^{(1)}, \ldots, \boldsymbol{\omega}^{(N)}\}$ drawn iid according to the underlying distribution μ, and if $(x^*, y^*(\cdot)) \in \mathcal{X} \times \mathcal{Y}$ is an optimal solution with respect to these N samples,[2] *then with probability $(1-\delta)$, x^* is feasible to the next sample drawn with probability at least $(1-\varepsilon)$, as long as we have:*

$$N \triangleq N(\delta, \varepsilon) \geq 2[n\frac{1}{\varepsilon}\ln\frac{1}{\varepsilon} + \frac{1}{\varepsilon}\ln\frac{1}{\delta} + n].$$

PROOF. Even though the results of Calafiore and Campi ([43]) say nothing about multistage optimization problems, we can directly apply their results here. Define the function:

$$f(\boldsymbol{x}, \boldsymbol{\omega}) = \begin{cases} 0, & \text{if there exists a vector } \boldsymbol{y} \text{ such that } (\boldsymbol{x}, \boldsymbol{y}) \text{ is feasible for } \boldsymbol{\omega}, \\ 1, & \text{otherwise.} \end{cases}$$

Then the set $\{\boldsymbol{x} : f(\boldsymbol{x}, \boldsymbol{\omega}) \leq 0\}$ is polyhedral, as it is the projection of a polyhedral set, and thus it is indeed convex for all $\boldsymbol{\omega} \in \Omega$. Therefore we can rewrite the problem in (4.4.6) as

$$\begin{array}{rl} \min: & \boldsymbol{c}^{\mathsf{T}}\boldsymbol{x} \\ \text{s.t.}: & f(\boldsymbol{x}, \boldsymbol{\omega}) \leq 0, \quad \forall \boldsymbol{\omega} \in \Omega, \end{array}$$

and then by sampling $\boldsymbol{\omega}$, we obtain the required probabilistic guarantees. Note that by the convexity of f, their sample complexity bound holds without considering the additional variables introduced in the second stage. This is crucial for obtaining sample complexity bounds independent of the nature or degree of the complexity. □

■ 4.4.3 Sample Complexity for Multistage Problems

In the previous section, we are able to show that if we care only about the feasibility of a projection of the variables onto the first-stage decisions, then the sample complexity required to provide particular (ε, δ) feasibility and reliability guarantees, can be taken independent of the complexity of the second stage adaptability. We are unable to extend a similar line of analysis to the multistage case. The essential reason for this is that in the multistage case, we must be careful to preserve the causality. Namely, in a three stage problem with variables $(\boldsymbol{x}, \boldsymbol{y}_1, \boldsymbol{y}_2)$, and uncertainty (ω_1, ω_2), by definition

[2] An optimal solution is one that is feasible to all the samples, and minimizes the objective function.

y_1 may be taken to be a function of ω but not of ξ, while y_2 may have unrestricted dependence. If we try to generalize the result above directly, we would write down the function f defining feasibility of x as follows:

$$f(x, \omega, \xi) = \begin{cases} 0, & \text{if there exists vectors } y_1, y_2 \text{ such that } (x, y_1, y_2) \text{ is feasible for } (\omega, \xi). \\ 1, & \text{otherwise.} \end{cases}$$

In the two-stage example, we are able to avoid the complexity of the second stage function because we can form a convex function f, defined based on a single sample. In the three-stage cases such as the above, formulations considering only a single sample do not respect the required causality; namely, in the function f above, y_1 is implicitly a function of the second stage uncertainty ω_2.

Thus it seems that the dimension and adaptability-free sample complexity results obtained in the previous section for two-stage problems, cannot be directly replicated in the multistage case. At least, it seems not in a straightforward manner.

Instead, we go down a different avenue. We show here that by fixing the structure of the adaptability using a feature function F, we are able to reduce to a single-stage problem where the number of samples required in order to obtain the (ε, δ) reliability-feasibility guarantees, is controlled by the choice of function F. In particular, F is the map to degree-d monomials, that is, if we consider degree-d polynomial adaptability for a K-stage problem and (at most) n variables at each stage, the sample complexity will be bounded by a polynomial in: $(n, K, (1/\varepsilon), \log(1/\delta))$.

Theorem 4.4

Consider a K-stage stochastic linear optimization problem:

$$\begin{aligned} \min: \quad & c^{\mathrm{T}} x + \textstyle\sum_{i=1}^{K} d^{\mathrm{T}} y(z_i) \\ \text{s.t.}: \quad & A(z)x + \textstyle\sum_{i=1}^{K} A_i(z) y(z_i) \le b, \end{aligned}$$

where $z = (\omega_1, \ldots, \omega_K)$, and $z_k = (\omega_1, \ldots, \omega_k)$. Furthermore, let the adaptability functions $y_i(\cdot)$ be restricted to be polynomial of degree d in their respective argument, z_i. Let n and l be the number of variables and dimension of the uncertainty vectors ω, respectively. Then if we sample:

$$N \triangleq N(K, n, l, d, \varepsilon, \delta) \ge 2\left[K\binom{n+d}{d}\frac{1}{\varepsilon}\ln\frac{1}{\varepsilon} + \frac{1}{\varepsilon}\ln\frac{1}{\delta} + K\binom{n+d}{d}\right],$$

then with probability at least $(1-\delta)$, the optimal solution we obtain from solving the sampled linear optimization, $(x^*, y_1^*(\cdot), \ldots, y_K^*(\cdot))$, is feasible with probability at least $(1-\varepsilon)$.

PROOF. Once we fix the functional form (i.e., d-degree polynomial) of the adaptability functions $y_i(\cdot)$, the K-stage problem reduces to a single stage *linear* optimization with stochastic uncertainty. Then the remainder of the proof follows directly from the proof of Calafiore and Campi ([43]), and by counting variables. Indeed, we have an explicit linear program with a finite number of variables, and thus we can appeal to their results on sample complexity. Note that we crucially use the structured aspect of the adaptability, in order to maintain a number of variables that is independent of the number of samples. Also, the *a priori* specified structure allow us to avoid requiring exponentially many samples, while also respecting the causality constraints. □

We remark that one may essentially follow the reasoning in [119] to show that such a result is not possible in this generality, for arbitrary adaptability.

■ 4.5 Sample Complexity Results II

In this last section, our complexity results were based on convex optimization considerations such as support constraints, as first developed by Calafiore and Campi. The complexity bounds obtained by such methods are tied to the dimensionality of the adaptability. It is possible, however, to limit the complexity of the adaptability in ways not directly linked to the dimensionality of the parameter space defining the possible adaptability functions. Adding regularization constraints, and controlling the size of the coefficients, for instance, may allow us to obtain good bounds on the sample complexity, even for classes of adaptability with very many parameters.[3] In this section, we seek more general complexity results of this nature. Therefore we move away from the Calafiore and Campi approach, and consider a learning theory approach.

Remark 4.3
As an additional consequence of using results from learning theory, the complexity bounds now provide feasibility and reliability guarantees that are uniform over the set of points feasible to the samples drawn, Ω_N. Such uniform results are typically stronger than what is required. However, we see that for linear families of adaptability, such as those that we consider, they come at little extra cost, compared to what is required to obtain feasibility and reliability guarantees for only the optimal point of the solution. In addition, such uniform results may themselves be of interest. For instance, in [78], the authors develop a parametric programming scheme that maps the Pareto frontier of robustness and performance. Doing this in a sampling context would require some uniform guarantees of feasibility.

Using this kind of learning theory setup in optimization, was first proposed in de Farias and Van Roy [51], in the context of single stage linear programs designed to solve a reduced version of the approximate linear program coming from a Markov Decision Problem.

[3] In the classification and regression context, this is quite common. See, e.g., [115],[64], and references therein.

In addition to obtaining refined sample complexity estimates, the main contribution of this section is the introduction of an additional parameter, that allows us to trade off sample complexity for robustness.

Let us first consider the generic case, where our feasible set is defined by the constraints $f(\boldsymbol{x}, \boldsymbol{\omega}) \leq 0$, for $\boldsymbol{\omega} \in \Omega$, and $\boldsymbol{\omega}$ is a random parameter. While we may not have an explicit expression for the distribution of $\boldsymbol{\omega}$, as in the previous section, we assume that we are able to generate *iid* samples of $\boldsymbol{\omega}$. We note that the case where we have access only to samples that are "approximately" generated by the correct distribution, is a generalization that can be addressed without much trouble (see, for instance, [61]).

Define the set of uncertainty realizations $\boldsymbol{\omega} \in \Omega$ for which a given point $\boldsymbol{x}$ is feasible:

$$C_{\boldsymbol{x}} \overset{\triangle}{=} \{\boldsymbol{\omega} \mid f(\boldsymbol{x}, \boldsymbol{\omega}) \leq 0\}.$$

Let $\mathcal{C} = \{C_{\boldsymbol{x}} : \boldsymbol{x} \in \mathcal{X}\}$ then denote the collection of these sets. Consider N samples, $\boldsymbol{\omega}^{(1)}, \ldots, \boldsymbol{\omega}^{(N)}$, drawn independently and at random, from the generating distribution. As we have done above, let $\mathcal{X}_N \subset \mathcal{X}$ denote the resulting set of feasible elements. Then the lowest probability of feasibility of any point $\boldsymbol{x} \in \mathcal{X}_N$, can be expressed as:

$$\inf_{\{C_{\boldsymbol{x}} : \boldsymbol{\omega}^{(i)} \in C_{\boldsymbol{x}}\}} \mathbb{P}(C_{\boldsymbol{x}}).$$

That is, this is the lowest measure set, $C_{\boldsymbol{x}}$ that happens to contain all of the sampled points, $\boldsymbol{\omega} \in \Omega_N$. The larger the collection of sets $\{C_{\boldsymbol{x}}\}$ is, the more likely it is that there will be a set that contains the sampled points, that has low measure. This is illustrated in Figure 4-5 below. The size of the collection of sets $\{C_{\boldsymbol{x}}\}$ is controlled by some measure of complexity of the classes.

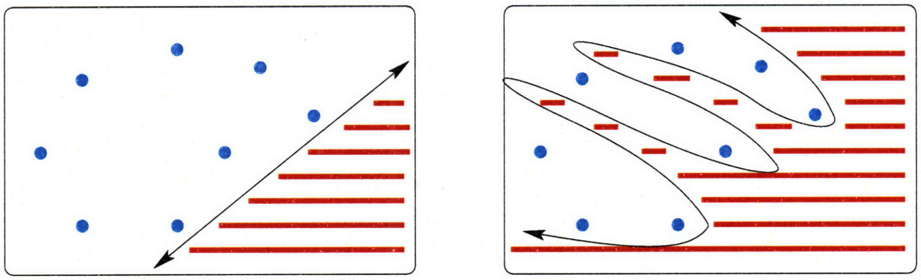

Figure 4-5. This figure illustrates the connection between the complexity of the sets $C_{\boldsymbol{x}}$, and the probability of error. In the figure on the left, the sets $C_{\boldsymbol{x}}$ are restricted to be hyperplanes. Any hyperplane that contains all the points on one side, must necessarily also contain most of the probability mass. On the figure on the right, we see that if the set $C_{\boldsymbol{x}}$ comes from a much richer class with nonlinear boundaries, then a set that contains all the sampled points may contain a much smaller fraction of the mass of the distribution.

Recall from Chapter 2, that one measure of the complexity of a set of functions, or sets, is the so-called VC-dimension. We can use this in a direct way to bound the quantity of interest. A simple result from learning theory (see, e.g., [4]), but appearing for the first time (to the best of our knowledge) in the optimization literature in [51], gives us the following:

Proposition 4.5
If we sample $N(\varepsilon, \delta)$ times with

$$N(\varepsilon, \delta) \geq \frac{4}{\varepsilon}\left(V_{\mathcal{C}} \ln \frac{12}{\varepsilon} + \ln \frac{2}{\delta}\right),$$

where $V_{\mathcal{C}}$ denotes the VC-dimension of the class $\mathcal{C}$, then with probability at least $(1-\delta)$, we have

$$\inf_{\{C_{\boldsymbol{x}} \,:\, \boldsymbol{\omega}^{(i)} \in C_{\boldsymbol{x}}\}} \mathbb{P}(C_{\boldsymbol{x}}) \geq 1 - \varepsilon.$$

Naturally, the usefulness of this result depends on how the VC-dimension, $V_{\mathcal{C}}$ of $\mathcal{C}$ scales. In [51], the authors considered only functions $f(\boldsymbol{x}, \boldsymbol{\omega})$ affine in $\boldsymbol{\omega}$ and in $\boldsymbol{x}$. Such functions can be written as $f(\boldsymbol{x}, \boldsymbol{\omega}) = \boldsymbol{a}(\boldsymbol{\omega})\boldsymbol{x} + b(\boldsymbol{\omega})$, and hence for such functions, the sets $C_{\boldsymbol{x}}$ can be written as:

$$C_{\boldsymbol{x}} = \{\boldsymbol{\omega} \in \Omega \,:\, \langle(\boldsymbol{a}(\boldsymbol{\omega}), b(\boldsymbol{\omega})), (\boldsymbol{x}, 1)\rangle \leq 0\},$$

which in turn can be written (using the adjoint of $\boldsymbol{a}$ and b) as affine functions of ω, and thus are half-spaces. The family of these half-spaces is parameterized (linearly) by the values of $\boldsymbol{x}$, and therefore their VC-dimension is at most $n = \dim(\boldsymbol{x})$ (this is an old result; see, e.g., [56]).

But now let us consider again our structured adaptability, where we have higher order adaptability functions. Then for the two-stage linear optimization problem, we can find a sample complexity bound that is valid for the worst-case feasible $\boldsymbol{x}$, yet is not far off the guarantee we derive in the previous section that holds only for the optimizing points $\boldsymbol{x}^*$.

Our framework of nonlinear adaptability, controls the nonlinearity through the use of the feature function F. Because the decision-variables are the coefficients of the linear map from the feature space, i.e., the image of the function F, we can again use the VC-dimension results for hyperplanes. Thus we have:

Proposition 4.6
Consider a two-stage linear optimization problem, with affine dependence on the uncertainty, and feature function any[4] function $F : \mathbb{R}^m \to \mathbb{R}^s$, and second stage adaptability affine in the

[4] Again by "any" we have the implicit requirement that evaluations of this function can be performed cheaply enough so as not to affect the computational complexity of the problem.

feature space:

$$\begin{array}{rl} \min: & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{s.t.}: & \boldsymbol{A}(\omega)\boldsymbol{x} + \boldsymbol{B}(\omega)\boldsymbol{y}(F(\omega)) \leq \boldsymbol{b}. \end{array}$$

Then as long as

$$N(\varepsilon,\delta) \geq \frac{4}{\varepsilon}\left(V_C \ln\frac{12}{\varepsilon} + \ln\frac{2}{\delta}\right),$$

where V_C is the VC-dimension of the class of sets C, then with probability at least $(1-\delta)$,

$$\inf_{\{C_{(\boldsymbol{x},\boldsymbol{y}(\cdot))}\,:\,\boldsymbol{\omega}^{(i)} \in C_{(\boldsymbol{x},\boldsymbol{y}(\cdot))}\}} \mathbb{P}(C_{(\boldsymbol{x},\boldsymbol{y}(\cdot))}) \geq 1-\varepsilon,$$

Moreover, since $\boldsymbol{y}$ is affine, we have: $V_C \leq s$.

For the case of polynomial adaptability, we have:

Corollary 4.7
If F maps to the space of monomials of degree at most d, and hence we have structured polynomial adaptability of degree at most d, then the above reliability and feasibility guarantees hold for the given sample complexity, where now

$$V_C = n + \binom{n+r}{r+1} + 1.$$

PROOF. The sets $C_{(\boldsymbol{x},\boldsymbol{y}(\cdot))}$ are defined as subsets of Ω, much like the single stage case. We give the definition for a single constraint:

$$C_{(\boldsymbol{x},\boldsymbol{y}(\cdot))} \triangleq \{\omega \in \Omega \,:\, \boldsymbol{A}_j(\omega)\boldsymbol{x} + \boldsymbol{B}_j(\omega)\boldsymbol{y}(F(\omega)) \leq b_j\}.$$

By our construction, our adaptability need not be affine in ω, but it is, however, affine in $F(\omega)$, and thus we can write the set $C_{(\boldsymbol{x},\boldsymbol{y}(\cdot))}$, as we did in the single-stage case, as a half-space in the feature space (the image of F), parameterized in an affine manner by the values of the first-stage decisions $\boldsymbol{x}$, and the coefficients $\boldsymbol{y}$. The result of the proposition then follows.

For the corollary, the proof follows from the fact that the sets $C_{(\boldsymbol{x},\boldsymbol{y}(\cdot))}$ are half-spaces in the (finite dimensional) feature (or kernel) space defined by the (non-homogeneous) polynomial kernel of degree $(r+1)$. The VC-dimension of this set is as given (see, e.g., [42]). □

We note that the VC-dimension of this class is quite close to the actual dimension of the problem, and therefore again, as in the linear case, these bounds very closely match the sample complexity bounds that guarantee only the feasibility of the optimizing point $(\boldsymbol{x}^*, \boldsymbol{y}^*(\cdot))$. Indeed, this is the case because, roughly speaking, the VC-dimension of families affinely parameterized (such as polynomials) is closely related to the dimension of the space in which the parameters live. In this sense, for the class of nonlinear adaptability functions we consider in our framework, the price we pay for the added strength of uniform feasibility guarantees, is essentially zero.

The bounds we have seen thus far, then, including the ones obtained by convexity considerations, as well as the uniform bounds, essentially use the dimension of the parameter space of the adaptability functions as a proxy for the complexity of the adaptability. This dimension translates directly into the sample complexity bounds. There are, however, other ways to control the complexity of a class of functions, that are not dependent on the dimension. Indeed, many notions of regularity, such as smoothness, slow variation, and parameter size, are able to control the complexity of a function class despite potentially large (in some cases infinite) dimension. The nonlinear mapping F introduced in Section 4.3.2 controls the dimension of the parameter space. In addition to this, we can regularize the space of adaptability functions by, for instance, controlling the size of the coefficients. We need, therefore, complexity measures that reflect such additional regularity properties, and are not just a function of the dimension. For this purpose, we introduce some more refined notions of complexity. In the process, we introduce an additional parameter that relates to how close we allow the sampled constraints to be violated.

Fat Shattering and Covering Numbers

The main quantity we want to bound, is

$$\inf_{\{C_{\boldsymbol{x}} : \boldsymbol{\omega}^{(i)} \in C_{\boldsymbol{x}} \; 1 \leq i \leq N\}} \mathbb{P}(C_{\boldsymbol{x}}).$$

This quantity depends on a notion of the complexity of the class of sets $\mathcal{C}$. The more complex the class, the more likely there that a set with low probability will happen to contain all the samples $(\boldsymbol{\omega}^{(1)}, \ldots, \boldsymbol{\omega}^{(N)})$.

The VC-dimension of a particular class of sets or functions, is a combinatorial quantity that is convenient to use in that it provides universal upper bounds on the complexity of the class. While it can be useful, in general it is quite loose as a measure of complexity, and thus the bounds obtained by using it are also quite loose. Furthermore, particularly for linearly parameterized classes of functions, such as those that are important in our context (namely, maps from the image space of the nonlinear mapping F) VC-dimension does not capture certain regularity properties that may in fact limit the complexity. Indeed, there are many classes of functions which have infinite VC-dimension, yet whose complexity can be controlled in terms of other quantities. There are other notions of complexity, such as covering numbers, and other combinatorial dimensions, that are more refined measures of complexity than the VC-dimension. Using these allows us to exercise more careful control of the upper bounds, and thus obtain more useful bounds, that also introduce new parameters of interest.

There is a large body of work related to covering numbers, both in learning theory, and also in functional analysis. See, for example, [5], [57], [136], and references therein. The purpose of this section is twofold: to illustrate that more sophisticated complexity estimates can provide better bounds than VC-dimension, and also to show that working directly with scale sensitive quantities like covering numbers, or the so-called fat shattering dimension ([1]), allow the introduction of new parameters that have useful interpretations directly in terms of the sampled robust optimization problem. We postpone a more thorough discussion of this area, and also of some more results, to Appendix B.

Here, we restrict ourselves, then, to only a brief discussion involving a refined complexity measure known as the *fat shattering dimension* of a class $\mathcal{F}$ of functions, and denoted by: $\mathrm{fat}_{\mathcal{F}}(\gamma)$. It is defined as follows:

Definition 4.1

The **fat shattering dimension** *of a class of* $\mathbb{R}$*-valued functions* $\mathcal{F}$ *mapping* Ω *to* $\mathbb{R}$*, is denoted by* $\mathrm{fat}_{\mathcal{F}}(\gamma)$ *and it is defined as follows. Let* $\Omega_N = \{\omega^{(1)}, \dots, \omega^{(N)}\} \subseteq \Omega$*, and* $\gamma > 0$*. Then the subset* Ω_N *is* γ*-shattered by the functions* $\mathcal{F}$ *if it is shattered with a margin* γ*, i.e., if there are numbers* $r_1, \dots, r_N$ *such that for any subset* $S \subseteq \{1, \dots, N\}$*, there is some function* $f_S \in \mathcal{F}$ *such that*

$$\begin{array}{rcll} f_S(\omega^{(i)}) & \geq & r_i + \gamma & \forall i \in S \\ f_S(\omega^{(i)}) & \leq & r_i - \gamma & \forall i \notin S. \end{array}$$

The parameter γ *defines the coarseness of the complexity measure. Because the quantity* $\mathrm{fat}_{\mathcal{F}}(\gamma)$ *depends on this additional parameter, the fat-shattering dimension is a scale-sensitive dimension.*

Sampling and Robust Optimization

We can now apply the above ideas to our present context. The above results suggest that there is much to be gained from using sample complexity results involving finer complexity estimates obtained from covering numbers and fat shattering dimension. We define the class of functions:

$$\mathcal{F} \triangleq \{f_{\boldsymbol{x}}(\omega) \triangleq f(\boldsymbol{x}, \omega) \, : \, \boldsymbol{x} \in \mathcal{X}\}.$$

Following the notation introduced above, let $\mathrm{fat}_{\mathcal{F}}(\gamma)$ denote the fat shattering dimension of $\mathcal{F}$ with scale parameter γ. We refer to this scale parameter γ as a margin parameter. The margin will take a concrete interpretation in the next definition, and also in the results below.

Definition 4.2
Given a parameter γ, and samples $\Omega_N = \{\boldsymbol{\omega}^{(1)}, \ldots, \boldsymbol{\omega}^{(N)}\}$, we define the feasible set with respect to the sample, and the margin parameter γ, as

$$\mathcal{X}_N(\gamma) \triangleq \left\{ \boldsymbol{x} : \begin{array}{c} f(\boldsymbol{x}, \boldsymbol{\omega}) \leq 0, \quad \forall \boldsymbol{\omega} \in B_\gamma(\boldsymbol{\omega}^{(1)}) \\ \vdots \\ f(\boldsymbol{x}, \boldsymbol{\omega}) \leq 0, \quad \forall \boldsymbol{\omega} \in B_\gamma(\boldsymbol{\omega}^{(N)}) \end{array} \right\},$$

where $B_\gamma(\boldsymbol{\omega})$ denotes the γ-ball about the point $\boldsymbol{\omega} \in \Omega$.

For the multi-stage problem with decision variables $(\boldsymbol{x}, \boldsymbol{y}_1(\cdot), \ldots, \boldsymbol{y}_K(\cdot))$, we denote the corresponding set of solutions feasible to this robustified sampled problem, by $\mathcal{XY}_N(\gamma)$.

Recall from Chapter 2, and also from the illustration in Figure 4-5, that VC-dimension, given by the number of points a family of sets (or functions) can shatter, provides a bound on the minimum probability a set must have, if it contains all the points sampled, i.e., if it contains Ω_N. Consider the definition of $\mathcal{X}_N(\gamma)$ given above. This corresponds to the family of all sets that not only contain the points $\boldsymbol{\omega}^{(i)} \in \Omega_N$, but rather contain the γ-balls around each of those points. We see below, that just as the VC-dimension controls the minimum weight of a set containing all sampled points $\boldsymbol{\omega}^{(i)} \in \Omega_N$, the fat shattering dimension controls the minimum probability of the smaller family of sets that contains all γ-balls about each sampled point. To get an intuitive idea of why the fat-shattering dimension should be no larger than the VC-dimension, consider Figure 4-6.

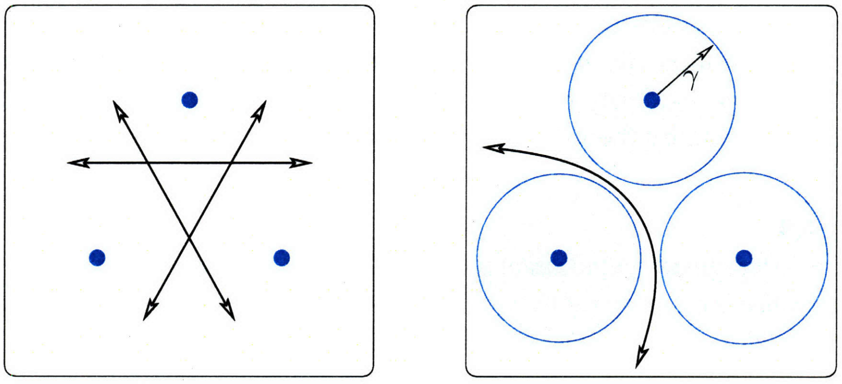

Figure 4-6. This figure shows that it is more difficult to shatter a set of γ-balls, than it is to shatter a set of points. This is particularly important when the balls and points are restricted to lie in a compact set, as then the regularity of the separating surfaces, meaning their curvature and smoothness, play an important role.

Proposition 4.8
Consider a two-stage linear optimization problem, and structured adaptability function $\boldsymbol{y}(\boldsymbol{\omega})$:

$$\begin{aligned} \min: \quad & \boldsymbol{c}^\top \boldsymbol{x} \\ \text{s.t.}: \quad & \boldsymbol{A}(\omega)\boldsymbol{x} + \boldsymbol{B}(\boldsymbol{\omega})\boldsymbol{y}(\boldsymbol{\omega}) \le \boldsymbol{b}. \end{aligned}$$

Let Ω_N denote N random samples of $\boldsymbol{\omega}$, and let $\mathcal{XY}_N(\gamma)$ denote the set defined above, of pairs of solutions feasible for the robustified sampled. Then with probability at least $(1-\delta)$,

$$\sup_{\{(\boldsymbol{x},\boldsymbol{y}(\cdot))\in\mathcal{XY}_N(\gamma)\}} \mathbb{P}(\boldsymbol{A}(\boldsymbol{\omega})\boldsymbol{x} + \boldsymbol{B}(\boldsymbol{\omega})\boldsymbol{y}(\boldsymbol{\omega}) - \boldsymbol{b} > 0) \le \varepsilon,$$

as long as the number of samples $N = N(\varepsilon, \delta, \gamma)$ satisfies

$$N(\varepsilon, \delta, \gamma) \ge \frac{4}{\varepsilon}\left(V_\gamma \ln \frac{12}{\varepsilon} + \ln \frac{2}{\delta}\right),$$

where V_γ denotes the fat shattering dimension of the collection of sets

$$\hat{C}_{(\boldsymbol{x},\boldsymbol{y}(\cdot))} = \{\boldsymbol{\omega} \in \Omega : \boldsymbol{A}(\boldsymbol{\omega})\boldsymbol{x} + \boldsymbol{B}(\boldsymbol{\omega})\boldsymbol{y}(\boldsymbol{\omega}) - \boldsymbol{b} \le 0\}.$$

PROOF. We defer the proof to Appendix B. We note that while we present the 2-stage case, the K-stage case is not appreciably different. □

Because generally the fat shattering dimension can be significantly sharper (i.e., lower) than the VC-dimension, the above proposition offers the potential of greatly improved sample complexity estimates. In particular, the fat-shattering dimension can be controlled by regularity conditions that are independent of the dimension of the function space, i.e., by the number of free parameters. For example, from [5] we have:

Proposition 4.9
If we have a two-stage linear optimization problem where the uncertain matrices $\boldsymbol{A}(\boldsymbol{\omega}), \boldsymbol{B}(\boldsymbol{\omega})$ have affine dependence on $\boldsymbol{\omega}$, and if we restrict ourselves to continuous functions of bounded variation, with total variation at most TV, then we have:

$$V_\gamma \le n\left(1 + \left\lfloor \frac{TV}{2\gamma} \right\rfloor\right).$$

Covering numbers and fat shattering dimension complexity measures are scale sensitive. That is, both produce a complexity measure with respect to a parameter; for covering numbers, this parameter is the size of the covering balls, and for fat shattering it is precisely the γ in the definition. Indeed, in addition to the improved sample complexity bounds of the proposition above, the new ingredient which has not appeared before, is an additional notion of robustness which we are building in to the process. That is, we build an optimization problem over the feasible set $\mathcal{X}_N(\gamma)$. Each point in this set is not just feasible to the N sampled points Ω_N, but in fact it is, by definition, feasible for every point in a γ-ball about each sampled point $\boldsymbol{\omega}^{(i)}$. As the bounds of Proposition 4.9 show, this robustness can play a roll in reducing the sample complexity. Indeed, the fact that γ explicitly appears in the sample complexity bounds, underscores the fact that there is a trade-off not only between the sample complexity, and then the reliability and feasibility parameters, ε and δ, but that this notion of robustness parameterized by γ, is also a factor.

In spirit, this bears a similarity to the approach of Nemirovski and Shapiro ([92]), where in order to obtain improved sample complexity bounds, they sample from a "worse" distribution, i.e., one that puts more weight on less favorable realizations of the uncertainty. Here, in order to improve our sample complexity estimates, we sample from the correct distribution, but impose a harsher condition on the sample feasible solution. In symbols, rather than computing sample complexity bounds required to control the quantity:

$$\inf_{\{C_{(\boldsymbol{x},\boldsymbol{y}(\cdot))}\,:\,\boldsymbol{\omega}^{(i)}\in C_{(\boldsymbol{x},\boldsymbol{y}(\cdot))}\}} \mathbb{P}(C_{(\boldsymbol{x},\boldsymbol{y}(\cdot))}) \geq 1-\varepsilon,$$

we control the quantity:

$$\inf_{\{C_{(\boldsymbol{x},\boldsymbol{y}(\cdot))}\,:\,B_\gamma(\boldsymbol{\omega}^{(i)})\subset C_{(\boldsymbol{x},\boldsymbol{y}(\cdot))}\}} \mathbb{P}(C_{(\boldsymbol{x},\boldsymbol{y}(\cdot))}) \geq 1-\varepsilon.$$

While the introduction of this additional parameter may reduce sample complexity dramatically, we have not yet addressed how its introduction affects the solvability of the sampled robust problem. Indeed, where as in the original sampled proposal we must solve a problem with N constraints, as formulated, we now must solve a problem with N robustified constraints, where the robust set is the Cartesian product of the γ-balls about the sampled points. As a consequence of the S-lemma, we immediately have:

Proposition 4.10

For the case of affine adaptability, the resulting robust sampled problem can be solved efficiently and exactly, via SDP.

PROOF. The proof is immediate. If the adaptability function is affine, then the inner problem is an optimization of a (possibly indefinite) quadratic function, subject to a single ellipsoidal constraint (the γ-ball). Therefore, by the S-lemma and by hidden convexity, (see, e.g., [39], [10], and the discussion in Chapter 2), this can be reformulated exactly as a Semidefinite Optimization. □

This avenue has two immediate drawbacks. First, our original sampling proposal had the distinct advantage that it involved solving only linear optimization problems. Semidefinite optimization, while still convex, is, practically speaking, of a different order of tractability than linear optimization. Furthermore, as the above proposition indicates, the use of the S-lemma is limited to affine adaptability. We would have no such recourse for the case of higher order models for adaptability.

For continuous adaptability models (that is, when the nonlinear mapping F is continuous) such as polynomial adaptability models, we can circumvent this problem by using the continuity to translate the margin in the ω-space, to a margin in the output space, or space of the inequalities. If the function $f(\boldsymbol{x}, \omega)$ is Lipschitz continuous in ω, and uniformly so with respect to $\boldsymbol{x} \in \mathcal{X}$, then we can translate the robustness parameter designating the size of the balls around each sampled point, into a margin that has a much more physical, intuitive, and in fact computable definition and meaning.[5] In particular, we will show that as long as we have this continuity property, then we can use the improved fat shattering results of Proposition 4.8, while still solving only linear optimization problems. And this continues to be true for case of general polynomial adaptability, and not just affine adaptability.

We define the sets

$$C_{\boldsymbol{x}}^{\eta} \triangleq \{\omega \, : \, f(\boldsymbol{x}, \omega) \leq -\eta\}.$$

Here, we can interpret the parameter η as being a measure of safety, or the distance from violation of a constraint. Then, the quantity we want to understand is:

$$\inf_{\{\boldsymbol{x} \, : \, \omega^{(i)} \in C_{\boldsymbol{x}}^{\eta}\}} \mathbb{P}(C_{\boldsymbol{x}}).$$

Note that we are taking the infimum over all $\boldsymbol{x}$ such that all the samples $\omega^{(i)}$ are contained in the sets $C_{\boldsymbol{x}}^{\eta}$, but then are considering the measure of the larger set $C_{\boldsymbol{x}}$. The following result is immediate.

Lemma 4.11 *Assume that the function $f(\boldsymbol{x}, \omega)$ is Lipschitz continuous, uniformly in $\boldsymbol{x}$, with constant L. Then, in particular, this means:*

$$||\omega - \omega'|| \leq \eta \Longrightarrow |f(\boldsymbol{x}, \omega) - f(\boldsymbol{x}, \omega')| \leq \eta L,$$

and therefore,

$$\omega \in C_{\boldsymbol{x}}^{\eta'} \Longrightarrow \omega' \in C_{\boldsymbol{x}},$$

as long as we have $\eta' \geq \eta L$.

[5] In fact, the Lipschitz constant need not be uniform over $\boldsymbol{x}$, since we can instead use the local Lipschitz constant. This greatly expands the class of functions we can consider. Furthermore, using local Lipschitz constants more properly captures the sensitivity of the constraint in question to the uncertainty sampled.

Therefore we are able to translate the robustness parameter, η, into a covering parameter. Specifically, we have:

$$\omega \in C_{\boldsymbol{x}}^{\eta} \Longrightarrow B_{\eta'}(\omega) \subset C_{\boldsymbol{x}}.$$

Therefore, analogously to Proposition 4.8, we now have:

Proposition 4.12

Assume the setup of Proposition 4.8, and assume further that the functions $f(\boldsymbol{x}, \boldsymbol{y}(\cdot), \omega)$ are Lipschitz continuous in ω, uniformly in $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{X}\mathcal{Y}$. Let Ω_N denote N random samples of ω. Then, with probability at least $(1 - \delta)$,

$$\sup_{\{(\boldsymbol{x},\boldsymbol{y}(\cdot)\,:\,\omega^{(i)} \in C_{\boldsymbol{x}}^{\eta}\}} \mathbb{P}(C_{\boldsymbol{x}}) \geq 1 - \varepsilon,$$

as long as the number of samples $N = N(\varepsilon, \delta, \gamma)$ satisfies:

$$N(\varepsilon, \delta, \gamma) \geq \frac{4}{\varepsilon}\left(V_\gamma \ln \frac{12}{\varepsilon} + \ln \frac{2}{\delta}\right).$$

4.6 Computational Examples

Here we give several examples to illustrate both the procedures described above, and also to demonstrate numerically their effectiveness. We are particularly interested in examples that illustrate the benefit to the first-stage decisions. Indeed, for multistage problems, the robust optimization formulation may (and in fact ought to) be implemented in a receding horizon manner. For practical scenarios and problems where re-optimization, and the receding horizon approach can be viably implemented, the value of adaptability must be judged in the improved quality (in terms of both objective function, and robustness) of the first-stage decisions. As we have remarked in the introduction, this focus is somewhat different from other work in adaptability, e.g., in [52] and [72].

4.6.1 Network Design

We consider first a network design problem:

$$\begin{array}{rl} \min: & \boldsymbol{c}^\top \boldsymbol{x} + \boldsymbol{d}^\top \boldsymbol{y} \\ \text{s.t.}: & \boldsymbol{F}\boldsymbol{y} \geq \boldsymbol{b} \\ & \boldsymbol{0} \leq \boldsymbol{y} \leq \boldsymbol{x} \\ & \boldsymbol{0} \leq \boldsymbol{x} \leq \boldsymbol{u}. \end{array}$$

Here, the first stage variables x are the capacity of each arc in the network. The second stage variables y are the flow variables, i.e., how much flow there is on each arc. The constraint $\boldsymbol{F}\boldsymbol{y} \geq \boldsymbol{b}$ gives the flow constraints that say that the amount flowing into any node minus the flow out of that node, must be at least equal (not necessarily strictly equal) to the demand at that node.

We consider the situation where the demand at each node is uncertain. We consider the following network model:

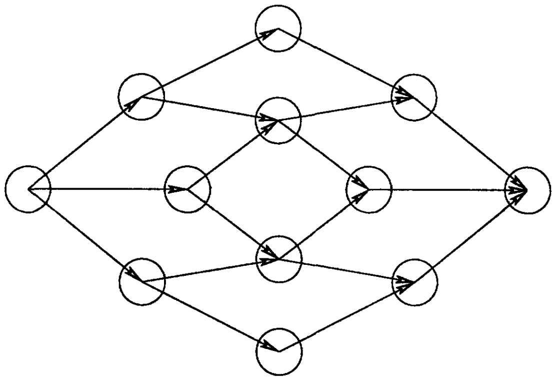

Figure 4-7. This figure shows the topology of the network we consider in Example 1.

The supply is essentially infinite at the source node. Nominally, the demand is set to zero at all intermediate nodes, and at 5 for the sink node. However, this vector of demands is not known exactly. We consider a simple random model, where the demand vector varies according to an additive normal perturbation about the nominal values. The Gaussian perturbations at each node may be correlated.

We compare the cases where the second stage variables have no dependence on the uncertainty, and when they have affine dependence, and finally when they have quadratic dependence on the uncertainty. We also consider the case of complete adaptability. Because in this section we focus on the two-stage case, by the projection results, the sample complexity is controlled. In particular, the upper bounds are no worse than for affine or quadratic adaptability. We consider later a three-stage network design problem, where this is no longer the case.

We find that, as one would expect, the completely adaptable model greatly outperforms the static case. However, the affine and quadratic models seem to perform no worse than the completely adaptable model.

Numerical Results

We report numerical results from the following experiments:

1. We generate 20 samples of the uncertainty: in this case, the demand at each node.

This is generated according to two different distributions:

(a) In the first, we generate the data uniformly at random from the scaled simplex. In this case, the data always sum to the scaling level of the simplex.

(b) We also generate the demand at each node from *iid* samples of a uniform random variable. In this case, the samples are uncorrelated. We adjust the support of the uniform random variable so that the two methods produce a demand vector with the same mean.

2. We solve using: static robust (no adaptability), affine adaptability, quadratic adaptability, complete adaptability.
3. All numbers represent averages over 50 trials.

Random Trial	Static Robust	Aff. Adapt	Quad. Adapt	Comp. Adapt
(a): $\beta = 1$	312.9	294.2	294.2	294.2
(b): $\beta = 1$	309.7	298.1	298.1	298.1
(a): $\beta = 5$	534.2	430.9	430.9	430.9
(b): $\beta = 5$	507.8	448.9	448.9	448.9
(a): $\beta = 10$	785.7	598.5	598.5	598.5
(b): $\beta = 10$	757.0	601.1	601.1	601.1
(a): $\beta = 15$	1070.5	777.0	777.0	777.0
(b): $\beta = 15$	1005.0	825.8	825.8	825.8
(a): $\beta = 20$	1339.5	949.7	949.7	949.7
(b): $\beta = 20$	1254.4	1004.1	1004.1	1004.1

Table 4.1. This table shows values for the two-stage network design problem, generated for the two different sources of uncertainty detailed above, for different normalization constants, and for the static, affinely adaptable, as well as the quadratic and complete adaptability cases. Here, β is the normalization parameter. The entries indexed by (a) denote uncertainty generated according to the first procedure (and thus are correlated) while those marked with (b) are generated according to the second procedure (and hence are independent).

It is helpful to see some of the numbers comparing the static and affine adaptability in a graph. In Figure 4-8, we plot the ratio of the value of the static robust solution, to the value of the affine adaptable solution, for the case where the uncertain demand is generated from the scaled simplex. Thus the sum of the demand vectors is always equal to the normalization constant. We plot this ratio as we increase this normalization.

Note that by drawing the demand realizations at each node, from the scaled simplex, the demand uncertainties become correlated, since their sum is fixed *a priori*. As discussed at length in this thesis, a failure of the robust framework is that it is unable to capture correlations across constraints. Thus we would expect the ratio between the adaptable and the robust to be greater in the case of correlated data, than for the case of independent data. We plot both curves in Figure 4-9, where we see that the data indeed bear out this general claim.

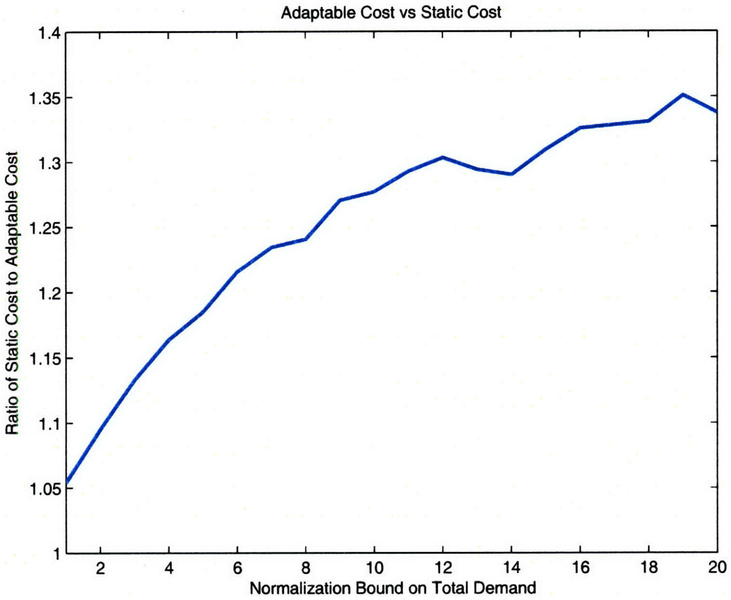

Figure 4-8. This figure gives the ratio of the static robust to the affine adaptable case (for this problem affine is as good as the completely adaptable case) when the demand at each node is chosen at random from a uniform distribution on the simplex scaled by β. Thus the sum total demand is always equal to β. The horizontal axis of the graph gives the value of this β. We see that the ratio between the static and the affine grows, as β increases.

Maximizing Robustness for a Fixed Budget

In the optimization formulation of the above adaptability models, the objective is not feasibility, but rather to minimize the cost. We can also ask, for a fixed first-stage budget, how the feasibility of the first-stage solution changes as we introduce adaptability in the second stage (in general, in future stages).

The upper bounds on sample complexity discussed above, give bounds on the the sample complexity that are directly related to the complexity of the adaptability functions. Indeed, for the problems formulated, if the size of the sample Ω_N is fixed, then the (δ, ε) guarantees we obtain become weaker as the complexity of adaptability increases. We use the network design example introduced in this section, in order to further examine the interplay of feasibility and adaptability. We show that, counter to the intuition provided by the sample complexity bounds, adaptability can in fact

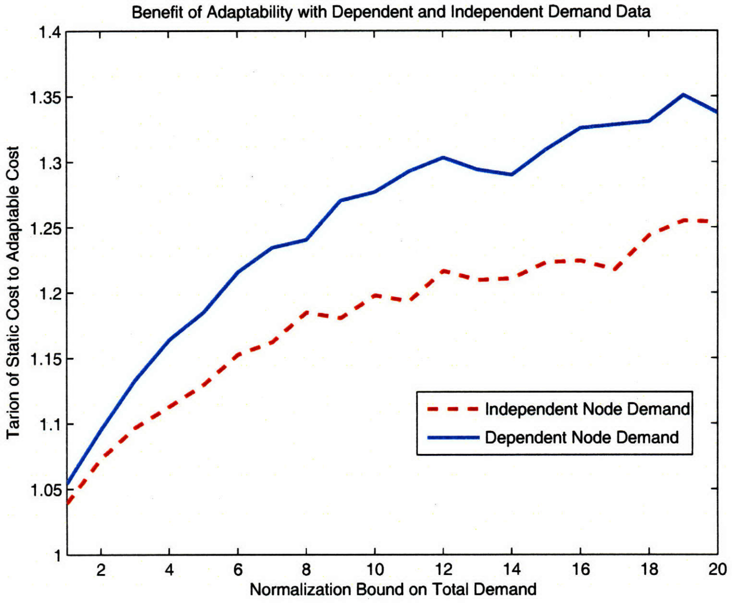

Figure 4-9. This figure illustrates the increased benefits of adaptability in the presence of correlation in the data. Here, we compare the ratio of the static robust case, and affine adaptability, in two scenarios: first, when the demand at each node is chosen from the scaled simplex, and second, when each demand is chosen independently and at random, from a range $[0, \hat{\beta}]$. Here, $\hat{\beta}$ is chosen so that the expected total demand equals the expected total demand in the normalized case, namely, the expected total demand should equal β. We see that the ratio between the static and adaptable cases is larger when the data have this correlation.

provide increased feasibility of the first-stage decisions, at a lower first-stage cost than the static solution trained on the same sample set Ω_N. The intuition is that without adaptability, the optimization problem fails to capture important correlation information of the data. That is, as discussed above (and more extensively in [22]), the static robust formulation effectively replaces the uncertainty set with the smallest hypercube that contains it. As we demonstrate by example below, this can lead to a misallocation of resources, causing the decision-maker to add protection where it is unnecessary. Adaptability allows correlations in the data to be captured and exploited, and thus the resulting first-stage decisions can be much more robust and feasible. On the other hand, if the number of samples is fixed, then increasing the adaptability can lead to essentially *over-fitting*, thus deteriorating the robustness and feasibility of the first-stage decision. This calls for a Structural Risk Minimization approach to optimization. We take up a full discussion of this issue elsewhere, and for now we settle for a brief discussion, and some illustrative examples.

Given a multistage optimization problem

$$\begin{array}{rl} \min: & \boldsymbol{c}^\top \boldsymbol{x} + \sum_{i=1}^K \boldsymbol{d}^\top \boldsymbol{y}_i(\boldsymbol{z}_i) \\ \text{s.t.}: & \boldsymbol{A}(\boldsymbol{z})\boldsymbol{x} + \sum_{i=1}^K \boldsymbol{A}_i(\boldsymbol{z})\boldsymbol{y}_i(\boldsymbol{z}_i) \le \boldsymbol{b}, \end{array}$$

the problem we would like to solve is then as follows:

Feasibility Maximization

1. Given a fixed data sample $\Omega_N = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_N\}$, assumed to be drawn *iid* from the generating distribution;

2. Given a budget β_1 on the first stage cost;

3. Find:

$$\begin{array}{rl} \max: & \mathbb{P}(\boldsymbol{x} \text{ is feasible}) \\ \text{s.t.}: & \boldsymbol{c}^\top \boldsymbol{x} \le \beta_1 \\ & \boldsymbol{A}(\boldsymbol{z})\boldsymbol{x} + \sum_{i=1}^K \boldsymbol{A}_i(\boldsymbol{z})\boldsymbol{y}_i(\boldsymbol{z}_i) \le \boldsymbol{b}. \end{array}$$

Adding a restriction to future stage actions is also possible under precisely the same framework. For the remainder of the discussion and the examples, we consider only a bound on the first-stage costs.

The objective function, i.e. the probability of feasibility, is difficult to handle. It is not clear how to even formulate this problem exactly as a finite dimensional optimization problem. Indeed, as posed, the problem is not well defined, since the true distribution of the uncertainty is unknown to us. The only information we have is the samples, and thus the objective function must have an interpretation either with respect to some reliability parameters (ε, δ), or as a worst-case result over the set of distributions that satisfy some level of consistency with the data sample, Ω_N.

The issue of proper definition aside, the objective function is typically non-convex, regardless of its definition, and this renders any optimization problem of this form difficult to solve. Indeed, even empirical evaluation of the objective function is difficult and not well-defined if we have more than two stages in the optimization.

Instead, then, we propose a robustness maximization formulation, that makes sense for a limited class of problems where the there is a physical meaning associated to scaling the uncertainty by a positive scaler. We consider the following formulation:

$$\begin{array}{rl} \max: & s \\ \text{s.t.}: & \boldsymbol{c}^\top \boldsymbol{x} \le \beta_1 \\ & \boldsymbol{A}(\boldsymbol{z})\boldsymbol{x} + \sum_{i=1}^K \boldsymbol{A}_i(\boldsymbol{z})\boldsymbol{y}_i(\boldsymbol{z}_i) \le \boldsymbol{b}, \quad \forall \boldsymbol{\omega} \in s\Omega_N. \end{array} \tag{4.6.7}$$

Here, $s\Omega_N = \{s\boldsymbol{\omega}^{(1)}, \ldots, s\boldsymbol{\omega}^{(N)}\}$ denotes the given set of samples, scaled by the parameter s. The optimization given in (4.6.7) can be easily solved by bisection, by computing the solution to a small number of feasibility problems.

We have yet to say anything about the nature of the adaptability of the future stage decision. It is clear that as written, for a fixed and specified sample Ω_N, increasing the adaptability increases the value of the optimization, that is, our proxy for robustness and feasibility increases. This does not mean, however, that the first stage decision is truly more robust, or feasible. Since the number of samples is fixed to be N, increasing the level of adaptability decreases the (δ, ε) guarantees we get from the sample complexity results in the first part of this chapter. Thus, there must be a trade-off between the increased value of the above optimization problem in (4.6.7), and the reduced feasibility guarantees. Intuitively, the trade-off is between more flexibility in the adaptability to capture correlations in the data, versus over-fitting due too high a level of adaptability for the given level N of samples. A full discussion of these trade-offs and the connection to SRM is discussed elsewhere.

We next give some examples of this phenomenon, in the context of a three-stage network design problem. Consider the network design problem, with the same network topology and uncertainty, as given above. But now consider the three-stage problem, where in the first stage the decision-maker builds a capacity for each link, and then assigns the flow (not violating the capacity constraints) to satisfy the demand at each node for the second and third constraint. In the adaptable case, the flow for the second stage may be a function of the realization of the first demand vector. Similarly, the third stage flow vector may be a function of both the first and the second flow vectors. The, setting $\boldsymbol{x}$ as the capacity vector, $\boldsymbol{y}$ as the second stage flow vector, and $\boldsymbol{v}$ the third stage flow vector, problem then, is:

$$\begin{array}{rl} \min: & \boldsymbol{c}^\top \boldsymbol{x} + \boldsymbol{d}^\top \boldsymbol{y} + \boldsymbol{f}^\top \boldsymbol{v} \\ \text{s.t.}: & \boldsymbol{F}\boldsymbol{y} \geq \boldsymbol{b}^1 \\ & \boldsymbol{F}\boldsymbol{v} \geq \boldsymbol{b}^2 - (\boldsymbol{b}^1 - \boldsymbol{F}\boldsymbol{y}) \\ & \boldsymbol{0} \leq \boldsymbol{y} \leq \boldsymbol{x} \\ & \boldsymbol{0} \leq \boldsymbol{v} \leq \boldsymbol{x} - \boldsymbol{y} \\ & \boldsymbol{0} \leq \boldsymbol{x} \leq \boldsymbol{u}. \end{array}$$

We consider the the static robust and dynamic formulations. In Figure 4-10, we draw the the capacity assigned in the first stage of the optimization problem. The thickness of links is meant to give a pictorial representation of the level of capacity assigned. A missing link signifies that the solution assigned zero capacity to that link. Indeed, we see that the robust solution actually removes several links from the network, i.e., assigns them zero capacity. The adaptable solution, meanwhile, assigns nonzero capacity to every link of the network. It is not difficult to see that this behavior of the robust solution is generic. This is a first indication that adaptability may in certain cases increase the robustness of the first-stage solution.

Next, we consider the approximation to the feasibility maximization problem given above, where we maximize the robustness level s subject to a fixed budget for the first

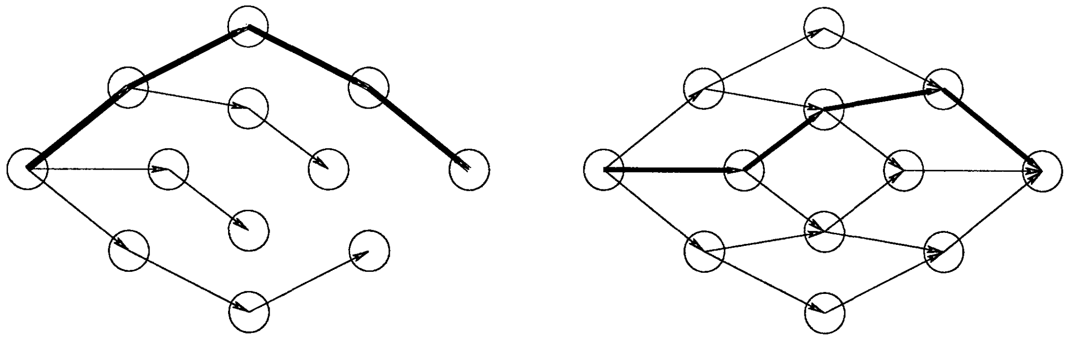

Figure 4-10. This figure gives a pictorial representation of the static and dynamic (i.e., robust, and adaptable) first-stage solution for the network described at the beginning of this section. The figure on the left represents the first stage solution for the robust formulation, and the figure on the right gives the adaptable solution. Note that indeed, the adaptable solution is more robust.

stage. We let the budget vary from the cost of the optimal adaptable solution as its lower bound, up to the cost of the optimal robust solution. There is only a static feasible solution at the upper bound of this interval. Because of this, the problem we solve gives us lower bounds to the feasibility improvement one obtains by introducing adaptability. This is because there is a single feasible solution to the robust formulation, and therefore the structure of the objective function does not affect the optimization. On the other hand, for the case of the adaptable solution, the problem we formulate may not produce the optimal (in the sense of feasibility) adaptable solution.

We obtain an adaptable solution by further approximating the formulation given in (4.6.7), by solving the adaptable problem on the same sample set as the static problem, and then scaling up the first stage solution, until it uses up the full level of allowable budget.

Then, for each solution obtained, we empirically check its feasibility. In the two-stage formulation, checking feasibility of a first-stage solution is simple, since this requires only generating demand realizations and checking for flows that satisfy those demands, while respecting the capacity constraints imposed by the first-stage solution. In the three stage problem, we must simulate the remaining two stages. That is, we must solve a sampled two-stage problem. For each first-stage solution, $\boldsymbol{x}$, we then test its feasibility as follows:

1. Generate a second stage demand vector, $\boldsymbol{b}^1$.

2. Generate N_2 third stage demand vectors, $\boldsymbol{b}^2_1, \dots, \boldsymbol{b}^2_N$.

3. We call $\boldsymbol{x}$ feasible if there exists a single flow vector $\boldsymbol{y}$, and N flow vectors $\boldsymbol{v}_1, \dots, \boldsymbol{v}_N$, such that the pair $(\boldsymbol{y}, \boldsymbol{v}_i)$ is feasible for the demand vectors $(\boldsymbol{b}^1, \boldsymbol{b}^2_i)$.

4. Repeat this procedure and compute the fraction of feasible outcomes for the first-

stage solution, $\boldsymbol{x}$.

This gives us an empirical feasibility for a particular first-stage capacity solution $\boldsymbol{x}$. We then repeat this entire procedure, generating different first-stage solutions $\boldsymbol{x}$, and finally averaging to obtain results for the feasibility.

In the results reported in the graphs below, we use a data sample of size 50 in order to train the adaptability functions and obtain a first-stage capacity vector. Then, for a given capacity vector, we generate a single second stage demand, and $N_2 = 25$ third-stage demand vectors in order to test the feasibility of the now fixed capacity vector. We repeat this procedure for the same capacity testing vector 100 times and compute an empirical feasibility percentage. We then repeat this entire procedure, generating new training data and a new capacity vector, and thus new feasibility statistics, 50 times. The average of these 50 numbers represents the empirical feasibility of a particular adaptability scheme.

The adaptability scheme we use is **affine** for the second stage, and **affine** for the third stage.[6]

The graphs are generated as follows:

1. Solve the static robust problem, and the affine problem. Let D_{static} denote the cost of the static, and D_{affine} the cost of the affine/affine implementations, and let $\boldsymbol{x}_{\text{static}}$ and $\boldsymbol{x}_{\text{affine}}$ denote the respective first-stage solutions.

2. Test the feasibility, according to the procedure outlined above, of $\boldsymbol{x}_{\text{static}}$, and also $\lambda \cdot \boldsymbol{x}_{\text{affine}}$, for $\lambda \in [1, D_{\text{affine}}/D_{\text{static}}]$. Thus for λ equal to the right endpoint, the scaled-up affine solution $\lambda \cdot \boldsymbol{x}_{\text{affine}}$ has the same cost (i.e., uses the same budget) as the static solution, $\boldsymbol{x}_{\text{static}}$.

We vary λ from 1 to $D_{\text{affine}}/D_{\text{static}}$ in 20 steps. Furthermore, we test feasibility not only against noise drawn from a distribution identical to the one generating the training data, but also against amplified distributions with more noise.

In Figure 4-11 we plot the empirical feasibility when we test the first stage solutions of the static robust, and the scaled affine adaptable solution, against noise generated by the same distribution, and the same strength as the noise generating the samples used to construct the static and affine solutions.

In Figure 4-12 we test feasibility against stronger noise, amplified by 110% of the noise generating the samples we use to construct the first-stage solutions. In Figure 4-13, we have the same, but the noise is now amplified by 130%. Finally, in Figure 4-14 the noise is amplified by 150%. We see that while the empirical feasibility decreases as the noise level increases (as one would expect) the scaled adaptable solution greatly exceeds the performance of the static robust solution at the extreme where the two

[6] We tried also using quadratic first second, third, and both second and third stages, but did not find any improvement over the affine/affine scheme. Therefore we report here only the affine/affine results.

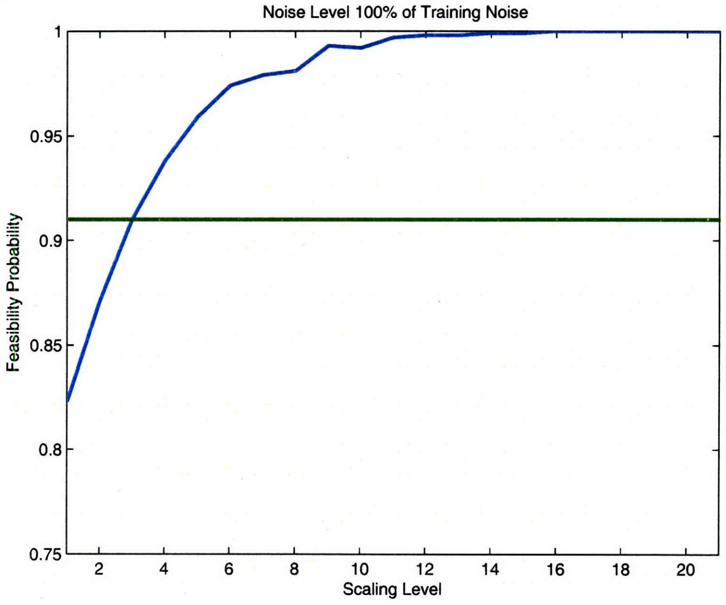

Figure 4-11. This figure gives the empirical feasibility for the static and affinely adaptable solutions for the three-stage network design problem described above. The horizontal axis represents the budget available for the adaptable solution, as a percentage of the minimum budget required to make the adaptable problem feasible. The scaling occurs in 20 steps, and these are indicated on the horizontal axis. Thus at the right-most point, the static robust, and affine adaptable first-stage solutions use the same level of resources. The vertical axis represents empirical feasibility, based on the procedure explained above. The feasibility of the affine adaptable solution, as expected, increases as the solution is scaled. It quickly surpasses the feasibility level of the static robust solution, and converges to essentially 100% feasibility.

solutions use the same level of first-stage resources, i.e., when we have: $\boldsymbol{c}^\top \boldsymbol{x}_{\text{static}} = \boldsymbol{c}^\top \boldsymbol{x}_{\text{affine}}$.

Finally, for the sake of comparison, we show the levels of feasibility for the amplified noise on the same graph, in Figure 4-15. This particular comparison particularly illustrates the feasibility advantages of adaptability. For the case of 150% amplified noise, the static solution exhibits very poor robustness to the noise. It is more often infeasible than feasible. The scaled affine solution, on the other hand, greatly outperforms it. Indeed, with the exception of the case of unamplified noise, the adaptable solution dominates the feasibility of the static robust solution throughout the full range of the scaling. This is significant, because at the left endpoint of the scaling, namely, at no scaling, the adaptable solution minimizes the first stage cost subject to satisfying only the sampled constraints, with no effort to build in further adaptability. Nevertheless, it exhibits significantly better feasibility than the static robust solution.

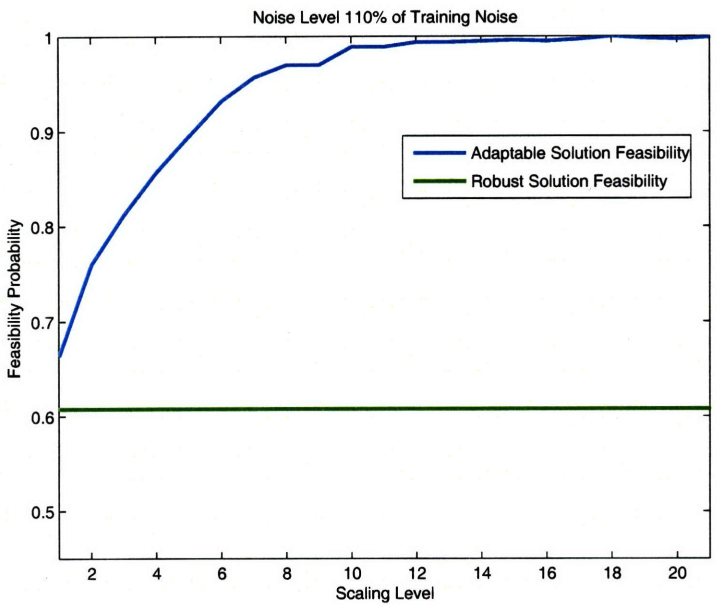

Figure 4-12. This figure gives the empirical feasibility for the static and affinely adaptable solutions for the three-stage network design problem when the noise is amplified by 110%.

■ 4.6.2 A Multistage Portfolio Problem

Here we consider a multistage portfolio problem. Our starting point is the formulation of Ben-Tal, Margalit, and Nemirovski in [13]. For other papers with robust optimization approaches to portfolio optimization, see [73], [60], and references therein.

We give only a brief overview of their development, skipping quickly to the formulation itself. For the full details and motivation, see [13]. The basic problem is to maximize the value of a portfolio of n assets, plus cash (an $(n+1)^{st}$ asset) after L investment stages. At each stage the decision-maker decides the amount to sell or buy for each commodity, and he faces a transaction cost as a percentage of the total sale or purchase. The returns for each commodity for each future period are uncertain, except for the returns for cash which, without loss of generality, and following [13], we treat as having certain return. Let x_i^l denote the amount of asset i held at stage l, y_i^l the amount of asset i sold at stage l, z_i^l the amount bought, and r_i^l the period l return. Let μ_i^l reflect the transaction cost of selling asset i at stage l, and ν_i^l that of selling. Then the

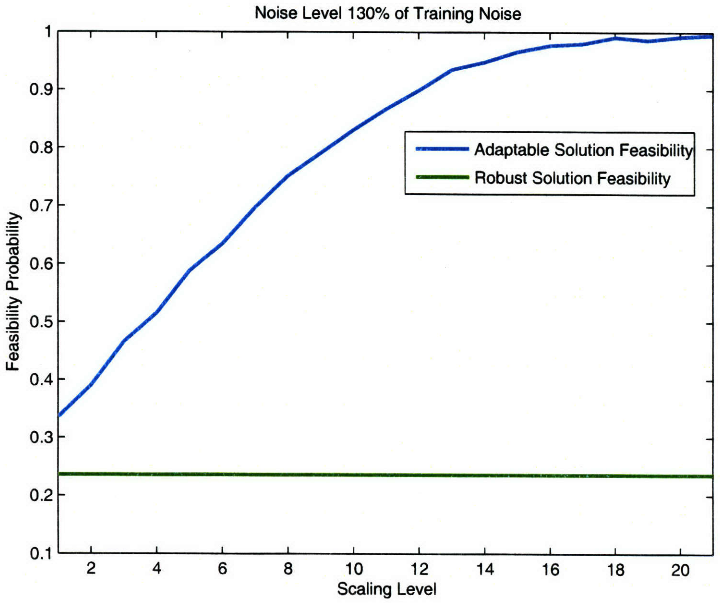

Figure 4-13. This figure gives the empirical feasibility for the case where we have 130% noise amplification. At the right endpoint, the scaled adaptable solution is almost 100% feasible, while the static robust solution is feasible less than 25% of the time. The cost of the two capacity solutions is the same at the right endpoint of the graph.

portfolio maximization problem becomes a linear optimization:

$$\begin{array}{rl} \max: & \sum_{i=1}^{n+1} r_i^L x_i^L \\ \text{s.t.}: & x_i^l = r_i^{l-1} x_i^{l-1} - y_i^l + z_i^l, \quad \imath = 1,\dots,n, \; l = 1,\dots,L \\ & x_{n+1}^l = r_{n+1}^{l-1} x_{n+1}^{l-1} + \sum_{i=1}^n (1-\mu_i^l) y_i^l - \sum_{i=1}^n (1+\nu_i^l) z_i^l \\ & y_i^l, z_i^l, x_i^l \geq 0, \quad \forall i, \; \forall l. \end{array} \tag{4.6.8}$$

One of the interesting contributions of [13], is an equivalent reformulation of this linear optimization, so that in the robustified version, there is in fact some adaptability to the realization of the returns in previous stages. To some extent, this allows us to take advantage of correlations in the uncertain returns across the different assets, but also over time.

$$\begin{array}{l} R_i^l = r_i^0 r_i^1 \cdots r_i^{l-1} \\ \xi_i^l = (R_i^l)^{-1} x_i^l \\ \eta_i^l = (R_i^l)^{-1} y_i^l \\ \zeta_i^l = (R_i^l)^{-1} z_i^l \\ a_i^l = (1-\mu_i^l) R_i^l / R_{n+1}^l \\ b_i^l = (1+\nu_i^l) R_i^l / R_{n+1}^l, \end{array}$$

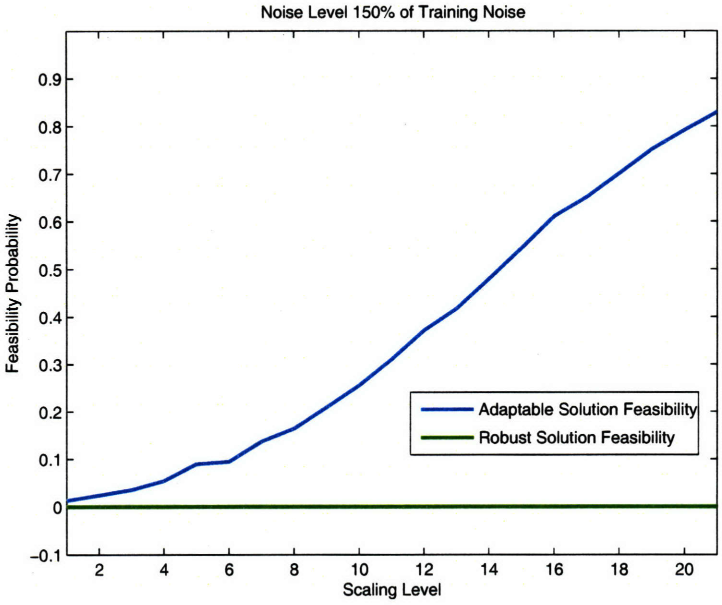

Figure 4-14. Here, the noise is amplified by 150%, in the same framework as the other figures above. Note here that the static robust solution is essentially useless for this level of noise, as it is almost always infeasible. The scaled adaptable solution, however, approaches 80% feasibility at the right endpoint of its scaling. Thus the gap between the feasibility for the scaled affine, and the static robust first-stage solutions, is over 80%, while the cost of the two solutions is the same.

the above formulation becomes equivalent to (again, for the full details and consequences of the reformulation we refer the reader to [13])

$$\begin{array}{rl} \max: & w \\ \text{s.t.}: & w \leq \sum_{i=1}^{n+1} R_i^{L+1} \xi_i^L \\ & \boldsymbol{\xi}^l = \boldsymbol{\xi}^{l-1} - \boldsymbol{\eta}^l + \boldsymbol{\zeta}^l \\ & \xi_{n+1}^l = \xi_{n+1}^{l-1} + \boldsymbol{a}^l \boldsymbol{\eta}^l - \boldsymbol{b}^l \boldsymbol{\zeta}^l \\ & \xi_i^l \geq 0 \\ & \eta_i^l \geq 0 \\ & \zeta_i^l \geq 0. \end{array}$$

Relaxing the two equalities to inequalities, we obtain an equivalent formulation, which

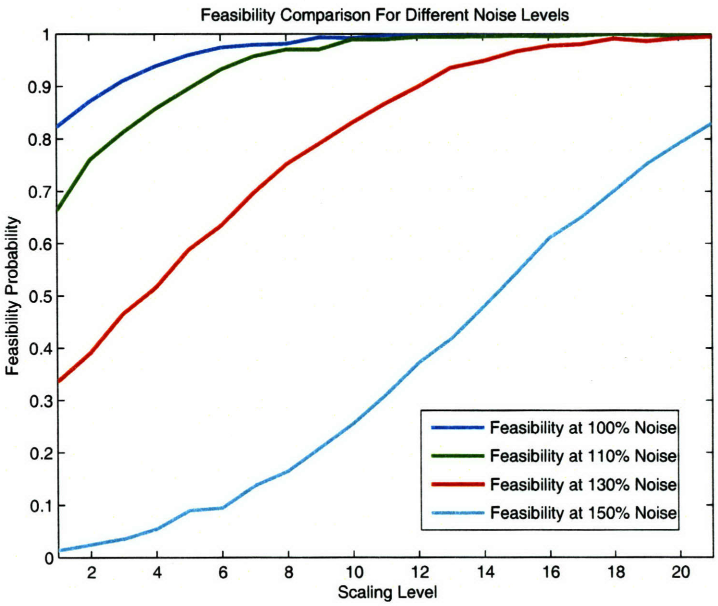

Figure 4-15. In this figure we show the comparison of the feasibility graphs for different amplifications of the testing noise.

is the nominal problem we consider:

$$\begin{array}{rl} \max: & w \\ \text{s.t.}: & w - \sum_{i=1}^{n+1} R_i^{L+1}\xi_i^L \leq 0 \\ & \boldsymbol{\xi}^l - \boldsymbol{\xi}^{l-1} + \boldsymbol{\eta}^l - \boldsymbol{\zeta}^l \leq 0 \\ & \xi_{n+1}^l - \xi_{n+1}^{l-1} - \boldsymbol{a}^l\boldsymbol{\eta}^l + \boldsymbol{b}^l\boldsymbol{\zeta}^l \leq 0 \\ & \xi_i^l \geq 0 \\ & \eta_i^l \geq 0 \\ & \zeta_i^l \geq 0. \end{array} \tag{4.6.9}$$

Note that now the optimization variables are not the concrete decisions of how much of asset i to buy or sell at period l. Instead, the decision variables specify the policy. That is, on realization $\boldsymbol{r} = (r_i^l)$ of the uncertainty, the policy at time l, for $1 \leq l \leq L$ is given by:

$$\begin{array}{rcl} x_i^l & = & (r_i^0 \cdots r_i^{l-1})\xi_i^l \\ y_i^l & = & (r_i^0 \cdots r_i^{l-1})\eta_i^l \\ z_i^l & = & (r_i^0 \cdots r_i^{l-1})\zeta_i^l. \end{array}$$

While this formulation, then, is adaptable, we nevertheless refer to it as the "static" solution in our computational experiment.

In this reformulation, the uncertainty in the returns affects the vectors $\{a^l\}$ and $\{b^l\}$. Thus this is a problem with uncertain matrix coefficients. The model of uncertainty, and thus the structure of the uncertainty sets used in the robust formulation of [13], assumes that the returns are correlated across assets, but not correlated from one stage to the next. Considering the nominal formulation 4.6.9 above, we see that this effectively enforces a rectangularity condition on the uncertainty sets. That is, the independence assumption amounts to constraint-wise uncertainty, since the uncertain returns for stage l appear in the single constraint

$$\xi^l_{n+1} - \xi^{l-1}_{n+1} - a^l \eta^l + b^l \zeta^l \leq 0,$$

and therefore different constraints are affected by different levels of uncertainty. It has been pointed in various places (e.g., [12], [22]) and indeed it is a straightforward exercise in duality, that in the face of rectangular (in this sense) uncertainty sets, adaptability cannot improve the robust solution, in the context of robust optimization, that is, in the context of a worst-case approach. Therefore, unless we change the notion of feasibility in [13] from a worst-case approach, to something else (like an expected penalty minimization approach) then it seems that adaptability such as the adaptability they use, or other more flexible models, cannot improve the solutions there. Thus, since they assume that the returns are independent from stage to stage, the conservatism of the robust solution is not because of the lack of adaptability with respect to the stochastic optimization formulation, but rather because robust optimization prepares for the worst-case scenario, while stochastic optimization tries to take care of the "average" case. Not surprisingly, on average cases (not to be confused with "on average") the stochastic optimization formulation (when computable to begin with) should outperform the robust. Because of the receding horizon approach, and because the correlations among returns in a single stage are captured by the reformulation of the LP (i.e., from (4.6.8) to (4.6.9)), the receding horizon robust approach compares very favorably to the stochastic programming approach, even on average cases. In large part, however, we believe that this is due to the imposed independence of the returns from one stage to the next. Removing this assumption, one expects to see considerable advantages to adaptability. It is precisely this scenario that we consider here.

A Stylized Example

Before we consider a generalization of the statistical model adopted in [13] to generate data randomly, we consider a particular (rather stylized) example that illustrates the potential benefits of introducing adaptability beyond what is contained in the formulation of [13], and also the potential benefit of higher order models for adaptability beyond the affine case. We consider first a 4 stage model with 2 risky assets, and a risk-free asset (cash). Suppose that the assets are negatively correlated with each other, and furthermore the returns of a single asset are negatively correlated over time. This limited universe may accurately model certain collections of goods. But to illustrate the point, we consider an extreme situation, where one good does extremely well, and the other very very poorly. Since there are two goods, we have two data points:

$$\begin{aligned} \boldsymbol{r}^{(1)} &= \begin{bmatrix} 2 & 0.01 & 2 & 0.01 \\ 0.01 & 2 & 0.01 & 2 \\ 1 & 1 & 1 & 1 \end{bmatrix} \\ \boldsymbol{r}^{(2)} &= \begin{bmatrix} 0.01 & 2 & 0.01 & 2 \\ 2 & 0.01 & 2 & 0.01 \\ 1 & 1 & 1 & 1 \end{bmatrix}. \end{aligned}$$

In this case, the static solution (the solution without further adaptability) puts everything into the risk-free asset. Meanwhile, the affine as well as the quadratic are able to exploit the pattern, and do dramatically better. The results are given in Table 4.2.

Let us next consider an extension of this example, to the case of three assets plus the risk-free asset, over 6 periods. When we have the three points:

$$\begin{aligned} \boldsymbol{r}^{(1)} &= \begin{bmatrix} 2 & 0.01 & 0.01 & 2 & 0.01 & 0.01 \\ 0.01 & 2 & 0.01 & 0.01 & 2 & 0.01 \\ 0.01 & 0.01 & 2 & 0.01 & 0.01 & 2 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \\ \boldsymbol{r}^{(2)} &= \begin{bmatrix} 0.01 & 2 & 0.01 & 0.01 & 2 & 0.01 \\ 0.01 & 0.01 & 2 & 0.01 & 0.01 & 2 \\ 2 & 0.01 & 0.01 & 2 & 0.01 & 0.01 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \\ \boldsymbol{r}^{(3)} &= \begin{bmatrix} 0.01 & 0.01 & 2 & 0.01 & 0.01 & 2 \\ 2 & 0.01 & 0.01 & 2 & 0.01 & 0.01 \\ 0.01 & 2 & 0.01 & 0.01 & 2 & 0.01 \\ 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \end{aligned}$$

then again the affine and quadratic formulations are equal in performance, and together outperform the static formulation. When we add the remaining three points to give all possible permutations of the rows, the performance of the affine and quadratic decreases, but again manages to track the profitable asset. The results of this simple example are contained in Table 4.2.

	Static	Affine	Quadratic
2pt Example:	3	24	24
3pt Example:	4.0	128.0	128.0
6pt Example:	4.0	64.32	64.32

Table 4.2. For the two point and three point example, the affine formulation is sufficient to track the asset with the strong growth. This is no longer true for the six point example. Here we see that quadratic adaptability outperforms the static and the affine adaptability formulations.

Statistical Model

The key difference between the results here, and what has been done before, is that we assume there is a dependence between the stage-to-stage returns. Our stochastic model for the returns is:

$$\begin{aligned} \ln r_i^l &= \Omega_i^T[\kappa e + \sigma v^l], \quad l = 0, \ldots, L, \; i = 1, \ldots, n; \\ \ln r_{n+1}^l &= \kappa, \quad l = 0, \ldots, L, \end{aligned}$$

where $\{v^0, \ldots, v^L\}$ are each k-dimensional Gaussian random vectors, and together are jointly Gaussian, but *not independent*. In particular, this means that the vectors $\{\boldsymbol{a}^l\}$ and $\{\boldsymbol{b}^l\}$, will be dependent across different values of l, i.e., different stages. Also, $e = (1, \ldots, 1) \in \mathbb{R}^k$, and $\Omega_i \in \mathbb{R}_+^k$, $\kappa, \sigma \in \mathbb{R}_+$ are given constants. We choose these parameters, as motivated by the work in [13], in order to have stock returns that exceed the safe return on cash, but whose variance is large enough so that the actual return has a sufficiently large probability of being less than the return to cash. For the full details and motivation, we refer the reader to [13]. For the case of three assets, in addition to cash, we choose the parameters so that we have returns as given below in Table 4.3.

	Asset 1	Asset 2	Asset 3	Cash
Avg Return:	1.0929	1.1135	1.140	1.02
Std. Dev:	0.0582	0.0865	0.1269	0.0000

Table 4.3. This table gives the values for the stock returns. Cash is a safe asset, and thus guarantees return 7.25%. The three stocks have average return below the return of cash, but they have a corresponding increase in volatility. Note that the higher the average return, the higher the standard deviation on the return.

In addition, we consider experiments where now we no longer have a risk-free asset. This is designed to test the behavior of the adaptability models, when it is not possible to put all the holdings in cash, thus avoiding all risk. The statistic of these returns are given in Table 4.4.

We have an additional parameter, α, governing the correlation between stages. The

	Asset 1	Asset 2	Asset 3	Cash
Avg Return:	1.0895	1.1068	1.260	1.1456
Std. Dev:	0.0558	0.0760	0.1086	0.1266

Table 4.4. This table gives the values for the stock returns when cash is no longer a safe asset. Again, the variances of the returns are chosen so that the higher the average return, the higher the corresponding variance.

correlation matrix is designed to look like:

$$\begin{pmatrix} I & \alpha I & \alpha^2 I & \alpha^3 I & \alpha^4 I \\ \alpha I & I & \alpha I & \alpha^2 I & \alpha^3 I \\ \alpha^2 I & \alpha I & I & \alpha I & \alpha^2 I \\ \alpha^3 I & \alpha^2 I & \alpha I & I & \alpha I \\ \alpha^4 I & \alpha^3 I & \alpha^2 I & \alpha I & I \end{pmatrix}$$

Thus the correlation is controlled by a single parameter, α, and the correlation between stages l and l' dies off like $\alpha^{|l-l'|}$.

Numerical Experiments

The numerical experiments test the robust approach against the adaptable approach, where we use affine and quadratic adaptability. The robust, affine, and quadratic models are all solved using the same set of samples. This allows us to fairly compare all three approaches. The sampling and receding horizon procedure we implement is as follows:

1. Let L be the number of stages.
2. Generate N samples of the L-stage uncertainty.
3. Solve the resulting LPs to obtain the robust, affine adaptable, and quadratically adaptable solutions for all L stages, feasible to the generated samples. Fix the solution policies computed (recall that even what we call the static robust solution here, in fact has some adaptability).
4. Generate 5,000 new sample paths for the L stages. Implement the computed solutions, and compute the returns.

Numerical Results

The first experiment is run as follows.

1. We have a model with 2 trading periods, and 4 assets, with a fourth asset representing risk-free cash. The returns are as given above in Table 4.3.

2. The initial portfolio at time zero contains one unit (or dollar) uniformly distributed across the four assets in the portfolio (that is, the three stocks and cash).

3. We consider 9 different levels of correlation between the stages: $\alpha = 0.8$, to $\alpha = -0.8$, in increments of 0.2.

4. For each fixed α, we generate $N = 300$ samples, i.e., 300 independent and identically distributed data points, for the returns. These are our training data. We use these to obtain a solution for the static (robust) solution. We also generate the policies for:

   (a) Affine adaptability: Here the investing decisions at time t may depend in an affine manner on all returns realized before time t.

   (b) Quadratic adaptability: Here the decisions may depend on the past returns, in a quadratic fashion.

   (c) Integer Affine adaptability: Here, we let F be the map $F : r \mapsto (r, \hat{r})$, where $\hat{r}_i$ is equal to 1 if the i^{th} return exceeded the return of the risk-free asset, cash, and is equal to zero otherwise.

   (d) Integer Quadratic adaptability: Here F is given by the composition of two maps: first, let F be the map $F : r \mapsto (r, \hat{r})$, where $\hat{r}_i$ is equal to 1 if the i^{th} return exceeded the return of the risk-free asset, cash, and is equal to zero otherwise. Then, take F_2 to be the map returning the quadratic values of the new vector. Then F is given as the composition of these two maps: $F = F_2 \circ F_1$.

5. With the policies generated, we next generate 5,000 new data points, and compute the average performance of each policy over these 5,000 points.

6. The above process is repeated 50 times for each value of the correlation coefficient α, and then the results are averaged. This is designed to reduce the variability that could arise from a particularly good or bad initial training data set.

Note that we are interested in the performance of the adaptability schemes not on the training data, but on the 5,000 new data points that are generated, and tested against the adaptability scheme computed by means of the training data. Table 4.5 reports the results of this computation.

We repeat the above experiment with the second distribution for the returns, as given above, where now the fourth asset is not risk-free, but rather the most volatile of all. In this case, we observe a different performance. We report the results in Table 4.6.

There are some common themes in the data presented in Table 4.5 and Table 4.6. In both cases, with and without the risk-free asset, the quadratic adaptability seems to have the best returns over the four models of adaptability we use, even when all

	Affine	Quadratic	Affine Int	Quadratic Int
$\alpha = 0.8$	0.1002	0.1233	0.1092	0.1009
$\alpha = 0.6$	0.1296	0.1365	0.1258	0.1280
$\alpha = 0.4$	0.0796	0.0749	0.0738	0.0797
$\alpha = 0.2$	0.0874	0.0920	0.0873	0.0875
$\alpha = 0.0$	0.0884	0.0849	0.0862	0.0878
$\alpha = -0.2$	0.0350	0.0311	0.0319	0.0343
$\alpha = -0.4$	0.0233	0.0213	0.0232	0.0224
$\alpha = -0.6$	-0.0074	-0.0054	-0.0087	-0.0076
$\alpha = -0.8$	-0.0093	-0.0053	-0.0119	-0.0112

Table 4.5. This table gives the difference in returns from the static policy, for the 2-stage portfolio problem for different levels of correlation, ranging from 0.8 to -0.8. This is the case where there is a risk-free low return asset (cash). We have the difference from the static robust returns for the affine, the quadratic, the integer affine, and the integer quadratic adaptability policies.

	Affine	Quadratic	Affine Int	Quadratic Int
$\alpha = 0.8$	-0.0839	-0.0662	-0.0734	-0.0714
$\alpha = 0.6$	-0.0593	-0.0440	-0.0496	-0.0570
$\alpha = 0.4$	-0.0483	-0.0310	-0.0419	-0.0436
$\alpha = 0.2$	-0.0504	-0.0407	-0.0372	-0.0488
$\alpha = 0.0$	-0.0238	-0.0147	-0.0263	-0.0237
$\alpha = -0.2$	-0.0057	0.0099	-0.0060	0.0015
$\alpha = -0.4$	-0.0057	0.0018	-0.0097	-0.0020
$\alpha = -0.6$	-0.0086	0.0055	-0.0100	-0.0017
$\alpha = -0.8$	-0.0052	0.0065	-0.0079	0.0004

Table 4.6. This table gives the difference in returns from the static policy, for the 2-stage portfolio problem for different levels of correlation, ranging from 0.8 to -0.8. In this case, the fourth asset is not risk free, but rather is the riskiest asset in the portfolio. As in Table 4.5, we have the difference in returns (with respect to the static robust policy) for the affine, the quadratic, the integer affine, and the integer quadratic adaptability policies.

four schemes are outperformed by the static robust model. In the presence of the low-return risk-free asset, in particular for positive correlations over time, the adaptability we introduce seems to improve the returns, over the static robust scheme. For high correlations, the static robust model places most of the portfolio in the risk-free asset. As a result, it is overly conservative, and as is revealed in the computations we present. For negative correlations, the advantages of adaptability seem to vanish, and in fact the static robust formulation slightly outperforms the other models.

Finally, we report the performance of the same experiment detailed above, but with 4 stages, instead of just two. The results are contained in Table 4.7. The advantages of increased adaptability in the case of high correlation across stages is more pronounced in the 4-stage case. However, the apparent disadvantage in the negative correlation regime is also more apparent. We believe that this illustrates the potential of overfitting when the number of samples is not sufficiently large.

	Affine	Quadratic	Affine Int	Quadratic Int
$\alpha = 0.75$	0.2520	0.2438	0.2509	0.2514
$\alpha = 0.25$	-0.0056	0.0011	-0.0126	-0.0069
$\alpha = -0.25$	-0.0672	-0.0902	-0.0816	-0.0715
$\alpha = -0.75$	-0.0314	-0.0475	-0.0406	-0.0307

Table 4.7. This table gives the difference in returns from the static policy, for the 4-stage portfolio problem for different levels of positive correlation. We have the returns for the static robust, the affine, the quadratic, the integer affine, and the integer quadratic adaptability policies.

■ 4.7 Conclusions

In this chapter we considered sampling approaches to designing structured adaptability for multi-stage problems. Sampling the uncertainty rather than solving a robust problem, replaces the worst-case with respect to a continuous uncertainty set, by a worst-case with respect to a finite set consisting of the samples. That is, we replace Ω by Ω_N. The benefit is that we circumvent the difficulty of solving the inner problem of robust optimization, as this is typically intractable. Solving the inner problem over the finite set Ω_N is done simply by enumerating the points of Ω_N. The price we pay is that the solution is no longer deterministically feasible, as in the Robust Optimization case. Rather, the robustness of the solution is defined by a reliability parameter δ, and a feasibility parameter ε. The meaning of (ε, δ), is that with probability at least $(1 - \delta)$, the solution obtained is feasible with probability at least $(1 - \varepsilon)$.

The second element of this chapter is the fact that we impose structure on the adaptability. The multi-stage adaptability is defined by nonlinear feature functions, F_i. The sample complexity required to guarantee a particular (ε, δ) pair, is affine in the sum of the dimension of the image of the mappings F_i. In particular, for well-behaved functions F_i, such as polynomial maps, the sample complexity is polynomial in the number of stages of the problem.

In addition, we consider computing bounds on sample complexity from two different perspectives: the convex optimization perspective, introduced by Calafiore and Campi, and then the approach motivated by results in statistical learning theory. The former approach is quite elegant, and when applicable, it provides very strong results. However, it is an analysis that ultimately counts the dimension of the parameter space defining the adaptability. We have considerable control over the adaptability functions we choose. In particular, this control extends well beyond merely the dimensionality of the space. The learning theory approach allows us to capture additional regularity aspects of the adaptability functions. We believe that this can prove to be important, especially as more interesting feature functions F, or even families of feature functions, $\mathcal{F}$, are employed.

The section containing our computational results, further points to some directions worth further exploration. The section on feasibility maximization, illustrates through the 3-stage network design problem, that adaptability manages to exploit structure in the problem in a way that the static robust approach does not. Thus, at least for the case of the network design problem, the adaptable solution seems to have better feasibility properties than the static robust solution, even though the upper bounds on sample complexity do not predict this. We believe this avenue deserves considerably more attention.

CHAPTER 5

Air Traffic Control: A Robust Adaptable Approach

This chapter considers the problem of air traffic control. We consider the global problem of scheduling ground delays, air delays, as well as dynamic route selection, subject to take off and landing capacity constraints on airports, as well as sector capacity constraints for sectors over the National Air Space (NAS). There are primarily two conceptual contributions: First, we consider uncertainty in the dynamically changing weather conditions, with the goal of building a schedule that is *robust* to this uncertainty. Second, we model the problem as one of sequential decision-making, thus placing the robust scheduling problem on a dynamic footing. While the weather impacted capacity uncertainty is naturally a high-dimensional object, we demonstrate that it can be well-approximated by a very low dimensional representation. We then exploit this low-dimensionality, adding what we term *finite adaptability* to the formulation, capturing important aspects of the uncertainty in a dynamic and adaptable framework.

We test the framework developed here on two single multi-sector, multi-airplane, single-airport problems, and thereby demonstrate the potential advantages of a finite adaptability approach.

■ 5.1 Introduction

The Airline industry in the United States makes up a huge part of the economy. Its revenue is in the hundred of billions of dollars, annually, and it transports close to 2 billion passengers. The impact on the international economy cannot be underestimated. There are many disparate challenges facing the Air Transport Industry. Many of these have been very successfully addressed by sophisticated optimization and other operations research tools developed by the community. For an excellent survey of applications of techniques of Operations Research in the Air Transport Industry see the review paper [9]. Indeed, many problems, such as aircraft and crew schedule planning (including fleet assignment, crew scheduling, maintenance routing, etc) and airline revenue management to name but two important areas, successfully implement sophisticated tools to manage uncertainty and large scale scheduling problems in some optimal sense (again, see [9]).

In the area of Air Traffic Management at the level of the global Air Traffic Control problem (as opposed to, say, gate scheduling, or slot assignment for takeoff and landing) the state of the art, i.e., what is currently implemented, lacks any appreciable degree of automation or assistance from optimization problems. Air Traffic Control of flights over the National Air Space (NAS) is accomplished by a hierarchical control system. At the top of the hierarchy is a single Air Traffic Control System Command Center (ATCSCC) that disseminates information and flow directives to 22 Air Route Traffic Control Centers (ARTCC), and each one of these control centers is further divided into approximately 20 sectors. In each such sector, there are Air Traffic Controllers, that maintain responsibility for each individual plane in their respective sector, and perform hand-offs with controllers from adjacent sectors when a plane crosses over from one sector to another. The total number of air traffic controllers over the NAS are around 20,000. The air traffic controllers are responsible for ensuring that planes maintain appropriate distance from each other, and that the capacity of their sector is not exceeded. The capacity of a sector is impacted by the weather conditions, and can be reduced significantly, even to zero, in the presence of severe weather conditions.

The air traffic controllers, responsible for up to about 25-30 flights each at any given time, and they instruct each flight to make small, local changes in altitude, speed, and direction. If a particular sector faces congestion, either due to too much incoming traffic, or adverse weather conditions that decrease the capacity of the sector, the air traffic controllers report this information up the hierarchy to the Air Route Traffic Control Centers, who can then send it further up the hierarchy to the Air Traffic Control System Command Center. It is thus higher up in the air traffic control hierarchy that more global decisions, such as assigning ground holds, or air holds, or redirection of flights, is decided.

This hierarchical system, while centralized, does not use a global optimization framework to synthesize the updated weather information with the location and trajectories of planes in the air and on the ground waiting to take off. Sector capacities are maintained by what amounts to essentially local modifications of the clear-weather schedule, scaling down and ramping up air traffic in a local manner in response to changes in sector capacities.

The focus of this paper is to address exactly this: the problem of Air Traffic Management (ATM), specifically the issue of managing delays due to dynamic weather conditions. The motivation is twofold, based first on the increasing costs of delays to the industry, and second, based on the recent advances in optimization under uncertainty.

The level of delays, as well as the cost of these delays to the airline industry, has been growing at a steady rate in the last decades. In the last half of the past decade, the average delay of US flights increased by almost 20%. The number of cancelled flights increased by over 65%. Meanwhile, the number of flights and passengers per year are forecasted to increase at a steady rate of around 4-5% annually ([9]). Such increases in traffic may cause even more severe increases in delay. A study by the European Organization for the Safety of Air Navigation (EUROCONTROL) Experimental Centre predicts a 26% increase in delay for a corresponding 5% increase in air traffic.

Given that the cost of delays to the airlines, airports, and consumers are measured in the billions, annually, severe increases in delays could well impede further growth of the airline industry.

In this paper we propose a mathematical framework for the dynamic scheduling problem of Air Traffic Control, impacted by uncertain weather. In large-scale scheduling problems, obtaining solutions with good performance requires capturing the global effects of local scheduling changes. Air Traffic Control is no exception, since delays propagate in many ways, and furthermore avoiding bottleneck phenomena such as gridlocked airports requires a more global view, that does not localize the problem by considering only a small subset of the aircraft to be scheduled. In this sense, our work is quite different in focus and scope from work such as [97] and [95].

To the best of our knowledge, this is the first optimization-based proposal that is designed to be able to handle such a global view, incorporating a significant fraction of the 30,000 daily flights over the NAS, that takes a dynamic viewpoint. Indeed, we believe that the central contribution of our proposal lies in two new features of the proposal: we focus on both robustness of the schedule to uncertainties in the weather forecast, and also adaptability. The ATM scheduling problem is a dynamic one: decisions are made sequentially over time, while weather information is updated, and thus the uncertainty is partially realized over time.

Therefore we apply the tools of robustness and adaptability developed in the first chapters of this thesis. In particular, we focus on finite adaptability as a tool to introduce adaptability in a manner that controls the number of variables, and can also accommodate the natural discrete aspect of the scheduling problem.

■ 5.2 Summary and Literature Review

There are many facets of the Air Traffic Management problem, ranging from local aspects, such as gate assignment, crew scheduling, contention for takeoff or landing slots that may often take a decentralized and game theoretic nature, as airlines compete for scarce resources (e.g., [130],[129], and also [77], [137]). The point of departure for the work of this chapter, is the formulation of Bertsimas and Stock in [29]. There, the authors formulate a linear integer programming approach to scheduling ground holding, and air holding so as to minimize delay costs. While they do not consider the fact that capacities change stochastically, nevertheless the basic model they develop is our point of departure, and in our computational experiments proves particularly capable of handling large-scale instances of the problem. We discuss the specifics of [29] and the relation to our work here, in Section 5.4.

In response to reduced capacities over certain sectors of the NAS, air traffic controllers can redirect flights, delay them in the air, or delay them on the ground. Ground holding costs are considerably less than the cost of holding a plane in the air, as this typically results in significantly increased fuel costs, as well as other costs, measured in passenger time and crew wages. There has been considerable effort devoted to the ground holding problem. The ground holding problem is further subclassified into the single airport ground holding problem, and the multi-airport ground holding problem. There have been several approaches for solving the ground holding problem. For example, [101] considers a finite perturbation analysis approach to the this problem. In [141] and [140], the authors consider an optimization-based scheme for the ground holding problem in a single airport, and in a network of airports. The work in [29] which essentially serves as our starting point, is more general, in that it considers routing, path selection and air-holding, as well as intermediate sector capacity constraints.

Recently, the authors in [97] and [95] have taken a Robust Markov Decision-Process approach to modeling the Air Traffic Management problem. This is an appealing approach, as it allows the modeling of the multi-stage nature of the problem, as well as the inherent uncertainty in the weather. However, the MDP approach makes some "rectangularity" assumptions, which effectively means that future stage decisions need have no adaptability on past realizations of the uncertainty. Thus that formulation is different in spirit from what we propose here. Furthermore, one of the central motivations of our approach, is the desire to obtain an algorithm applicable to a very large-scale problem. It is not clear that the MDP approach is appropriate for very large scale formulations, although the results presented in the referenced papers indeed seem to handle the smaller instances well.

The general formulation of air traffic control via on-line flight scheduling using optimization, dates back to [99]. Since then, there has been considerable work done, see for instance, [9], [3], [8], and [30], and references therein.

The new elements proposed here are robustness, and adaptability. That is, we explicitly model uncertainty in the weather forecasts, and then the subsequent adaptability to that uncertainty.

■ 5.3 Weather and Uncertainty

The current system for Air Traffic Control over the United States, is supported by several weather prediction tools for the major terminals, the minor terminals, and for the en route sectors. For a taxonomy of these, we refer the reader to the study produced by Lincoln Laboratory on their Corridor Integrated Weather System ([111]).

■ 5.4 The Model

The framework for our model begins with the work in [29]. This marked the first attempt to formulate in an optimization framework, the scheduling problem, including ground holding, air delay, and also route selection. While we incorporate both robustness and adaptability in our model, we start from fundamentally the same model. That is, the restriction of our optimization model to no adaptability and without uncertainty, gives back the model of [29]. Our model, then, is as follows:

■ 5.4.1 The Formulation: The Nominal Problem

In this section, we give the formulation of the scheduling problem without adaptability or robustness. We address those extensions in Sections 5.4.3 and 5.4.4, respectively.

We consider a set of time periods: $\mathcal{T} = \{1, \ldots, T\}$, flights: $\mathcal{F} = \{1, \ldots, F\}$, airports: $\mathcal{K} = \{1, \ldots, K\}$, continued flights: $\mathcal{C} = \{(f', f) \ : \ f' \text{ is continued by } f,\ f, f' \in \mathcal{F}\}$. Putting data of the problem are the following:

$$\begin{aligned}
\mathcal{P}_f &= \text{the collection of paths available for flight } f \in \mathcal{F}. \\
N_f(p) &= \text{number of sectors traversed by flight } f \text{ taking path } p \in \mathcal{P}_f. \\
P(f,i,p) &= \text{the } i^{th} \text{ sector of path } p \text{ of flight } f. \\
P_f &= \{P(f,i,p) : 1 \leq i \leq N_f(p)\} \\
D_k(t) &= \text{the departure capacity of airport } k \text{ at time } t. \\
A_k(t) &= \text{the arrival capacity of airport } k \text{ at time } t. \\
S_j(t) &= \text{the capacity of sector } j \text{ of the NAS at time } t. \\
d_f &= \text{the scheduled departure time of flight } f \in \mathcal{F}. \\
r_f &= \text{the scheduled arrival time of flight } f \in \mathcal{F}. \\
s_f &= \text{the time required to prepare aircraft completing flight } f \in \mathcal{F}. \\
c_f^g &= \text{ground holding cost for flight } f \in \mathcal{F}. \\
c_f^a &= \text{air holding cost for flight } f \in \mathcal{F}. \\
l_{fj} &= \text{minimum time for flight } f \text{ to traverse sector } j. \\
T_f^j &= \text{time window when flight } f \text{ can be in sector } j.
\end{aligned}$$

The decision variables specify if a particular flight $f \in \mathcal{F}$ has arrived at the i^{th} sector along path p by time t. If it has, then the decision variable is set to 1: $w_{ft}^{ip} = 1$. Otherwise we set $w_{ft}^{ip} = 0$. We can impose continuity of the flight paths, as well as constraints enforcing the planes in fact land, so as to give proper physical meaning to these variables. We can also use these variables to express the amount of ground-holding, and air-holding (air delay) time, thus allowing us to express the cost:

1. Ground Cost: We can express the total time a particular flight $f \in \mathcal{F}$ is ground-held as:
$$g_f \overset{\triangle}{=} \sum_{t \in T_f^k, p \in \mathcal{P}_f, k = P(f,1,p)} t(w_{ft}^{1p} - w_{f,t-1}^{1p}) - d_f.$$

2. Air Holding Cost: The air-holding cost of a flight $f in \mathcal{F}$ can be expressed as:
$$a_f \overset{\triangle}{=} \sum_{t \in T_f^k, p \in \mathcal{P}_f, k = P(f,N_f(p),p)} t(w_{ft}^{kp} - w_{f,t-1}^{kp}) - r_f - g_f.$$

3. Therefore the total cost of a particular schedule is then:
$$\sum_{f \in \mathcal{F}} \{c_f^g g_f + c_f^a a_f\}.$$

Next we have the constraints:

1. Capacity constraints for the departures at airport k at time t, capacity constraints for the arrivals, and capacity constraints for each sector of the NAS, at each time interval t:
$$\begin{aligned} \textstyle\sum_{f:P(f,1,p)=k} (w_{ft}^{kp} - w_{f,t-1}^{kp}) &\leq D_k(t) \quad \forall k \in \mathcal{K}, t \in \mathcal{T}, p \in \mathcal{P}_f \\ \textstyle\sum_{f:P(f,N_f(p),p)=k} (w_{ft}^{kp} - w_{f,t-1}^{kp}) &\leq A_k(t) \quad \forall k \in \mathcal{K}, t \in \mathcal{T}, p \in \mathcal{P}_f \\ \textstyle\sum_{f:P(f,i,p)=j', P(f,i+1,p)=j', i<N_f(p)} (w_{ft}^{kp} - w_{f,t-1}^{kp}) &\leq S_k(t) \quad \forall k \in \mathcal{K}, t \in \mathcal{T}, p \in \mathcal{P}_f. \end{aligned}$$

2. Connectivity between sectors: $\forall f \in \mathcal{F}, t \in T_f^j, p \in \mathcal{P}_f, j = P(f,i,p), j' = P(f,i+1,p), i < N_f(p)$:
$$w_{f,t+l_{fj}}^{j'p} - w_{ft}^{jp} \leq 0.$$

3. Continuity between airports for continued flights: $\forall (f', f) \in \mathcal{C}, t \in T_f^k, p \in \mathcal{P}_f, k = P(f,1,p) = P(f', N_f(p), p)$:
$$w_{ft}^{kp} - w_{f',t-s_{f'}}^{kp} \leq 0.$$

4. Connectivity in time: $\forall f \in \mathcal{F}, p \in \mathcal{P}_f, j \in P_f, t \in T_f^j$:
$$w_{ft}^{jp} - w_{f,t-1}^{jp} \geq 0.$$

5. Finally, we have the integrality constraints:

$$w_{ft}^{jp} \in \{0, 1\}.$$

The nominal optimization problem is defined by minimizing the cost function above, subject to the above constraints.

Size of the Nominal Formulation

The size of the nominal formulation (that is, as given above, without robustness or adaptability) depends on the number of time periods, sectors, and flights considered. Recall that $\mathcal{F}$ is the set of all flights considered, $\mathcal{C}$ the flights that are continued, $\mathcal{K}$ the airports, and $\mathcal{T}$ the set of time periods. Let $\mathcal{N}$ denote the set of sectors in the air space considered. Then, the number of variables in the nominal problem can be upper bounded by

$$|\mathcal{F}| \times \left(\max_{f \in \mathcal{F}, j \in P_f} |T_f^j| \right) \times \left(\max_{f \in F, p \in \mathcal{P}_f} N_f(p) \right).$$

The number of constraints can be upper bounded by

$$2|\mathcal{K}||\mathcal{T}| + |\mathcal{J}||\mathcal{N}| + 2|\mathcal{F}| \times \left(\max_{f \in \mathcal{F}, j \in P_f} |T_f^j| \right) \times \left(\max_{f \in F, p \in \mathcal{P}_f} N_f(p) \right) + |\mathcal{C}| \times \left(\max_{f \in \mathcal{F}, j \in P_f} |T_f^j| \right).$$

To give a feeling for the size of the formulation, consider that if we have 15,000 flights, and each flight is allowed up to 2 hours delay on the ground, and one hour delay in the air, and our time periods are 5 minute intervals, then since the number of sectors traversed is typically around 20, the upper bound for the variables becomes:

$$15,000 \times 36 \times 20 = 10,800,800,$$

while the number of constraints is bounded by

$$2 \times 15,000 \times 36 = 21,600,000.$$[1]

As we discuss in further detail below, several approximation techniques aid in the reduction of the size of the variables. Nevertheless, a problem of this size pushes the limits of computational and memory resources available today. Stochastic optimization methods, including the sampling results presented in Chapter 4 face the immediate challenge that expanding the number of constraints by a factor of, say, 300, as would be required by a sampling approach to the uncertainty with only 300 samples (let alone a sample size proportional to the number of variables, as in all the sample complexity bounds given) would put the problem well beyond the capacity of current

[1] This is the largest term in the upper bound for the number of constraints.

computing.

Instead, we use a robust approach, with finite adaptability, with a very small number of partitions. As explained further below, this allows us to provide feasibility guarantees while controlling the proliferation of variables, and at the same time giving a formulation that does not destroy the (empirically) strong integrality properties of the nominal formulation.

5.4.2 Sectors, Schedules, and Weather

The primitives for our dynamic optimization problem are the scheduled flights, meaning the published time table of the scheduled departure and arrival times, and departure and arrival coordinates, as well as the actual structure of the national air space, including the topology of the sectors, and the allowable paths from any given location to another. In addition to this, the weather forecast for the next 2 hours, as well as the current weather conditions, must be specified. In fact, the input to the optimization must be the sector capacities, which must be determined as a function of the current weather conditions in a given sector. While the FAA has established guidelines for determining the safe number of planes over a particular sector, these are far from being able to determine the capacity of each sector. In practice, it seems that the individual air traffic controllers locally determine capacity, based on experience, as opposed to clearly quantifiable guidelines ([63], [111]). The capacity of each sector at each time period was determined in an operational manner, by tracking flights over different weather scenarios over each given sector. Since weather impacted sector capacities are the scarce resource, obtaining accurate estimates that can be used by the optimization to produce trustworthy scheduling recommendations, is of high importance. The computational experiments we present in Section 5.5 on the benefit of finite adaptability, do not rely on capacity obtained from the real weather data.

5.4.3 The Robust Problem

The primary source of uncertainty which we address, is due to the weather, and its impact on the sector capacities at a given time. Therefore we have uncertainty affecting the right hand side vector of the nominal optimization problem constructed above, and the constraints affected by the uncertainty are the ones reflecting the capacity constraints. The uncertainty in the capacity is captured in the robust formulation by replacing the nominal capacity vector (across both time and sectors) by an uncertainty set. This uncertainty set reflects the uncertainty in the weather forecast. The nominal capacity vector, $(D(t), A(t), S(t))$, defines a trajectory over time in the capacity space, and we can think of the full trajectory as a point in the time-capacity space, that is, a point in $\mathbb{R}^{|\mathcal{T}|\times(|\mathcal{K}|+|\mathcal{K}|+|\mathcal{S}|)}$. The uncertainty set is a subset of this high-dimensional space: $\Omega \subseteq \mathbb{R}^{|\mathcal{T}|\times(|\mathcal{K}|+|\mathcal{K}|+|\mathcal{S}|)}$. Considering the uncertainty set as a subset of this space,

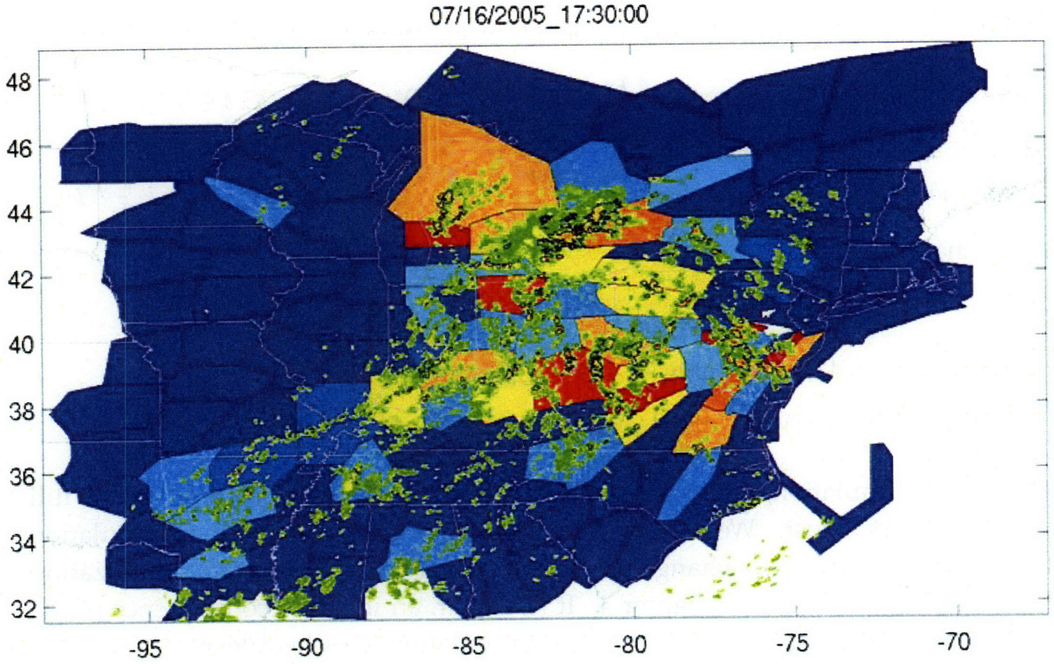

Figure 5-1. This figure shows the color-coded sector blockage, with the weather generating the blockage superimposed over the sector map. This illustrates the fact that sector capacity can offer only an at times coarse quantization of the true impact of the weather.

allows us to capture the correlations in the weather across both time, and sectors. As we further discuss in Section 5.4.4, modeling and exploiting these correlations is important for obtaining a robust formulation that is not too conservative. The constraints affected by the uncertainty are:

$$\begin{array}{rll} \sum_{f:P(f,1,p)=k}(w^{kp}_{ft}-w^{kp}_{f,t-1}) & \leq D_k(t) & \forall k \in \mathcal{K}, t \in \mathcal{T}, p \in \mathcal{P}_f \\ \sum_{f:P(f,N_f(p),p)=k}(w^{kp}_{ft}-w^{kp}_{f,t-1}) & \leq A_k(t) & \forall k \in \mathcal{K}, t \in \mathcal{T}, p \in \mathcal{P}_f \\ \sum_{f:P(f,i,p)=j',P(f,i+1,p)=j',i<N_f(p)}(w^{kp}_{ft}-w^{kp}_{f,t-1}) & \leq S_k(t) & \forall k \in \mathcal{K}, t \in \mathcal{T}, p \in \mathcal{P}_f. \end{array}$$

In a single-stage model, the robust version of these constraints now becomes:

$$\left\{ \begin{array}{rll} \sum_{f:P(f,1,p)=k}(w^{kp}_{ft}-w^{kp}_{f,t-1}) & \leq D_k(t) & \forall k \in \mathcal{K}, t \in \mathcal{T}, p \in \mathcal{P}_f \\ \sum_{f:P(f,N_f(p),p)=k}(w^{kp}_{ft}-w^{kp}_{f,t-1}) & \leq A_k(t) & \forall k \in \mathcal{K}, t \in \mathcal{T}, p \in \mathcal{P}_f \\ \sum_{f:P(f,i,p)=j',P(f,i+1,p)=j',i<N_f(p)}(w^{kp}_{ft}-w^{kp}_{f,t-1}) & \leq S_k(t) & \\ \forall k \in \mathcal{K}, t \in \mathcal{T}, p \in \mathcal{P}_f & & \end{array} \right\}$$

$$\forall (D(t), A(t), S(t)) \in \Omega.$$

Because of the worst-case nature of the robust formulation, this reduces to:

$$\begin{aligned}
\textstyle\sum_{f:P(f,1,p)=k}(w_{ft}^{kp} - w_{f,t-1}^{kp}) &\le D_k^*(t) \quad \forall k \in \mathcal{K}, t \in \mathcal{T}, p \in \mathcal{P}_f \\
\textstyle\sum_{f:P(f,N_f(p),p)=k}(w_{ft}^{kp} - w_{f,t-1}^{kp}) &\le A_k^*(t) \quad \forall k \in \mathcal{K}, t \in \mathcal{T}, p \in \mathcal{P}_f \\
\textstyle\sum_{f:P(f,i,p)=j',P(f,i+1,p)=j',i<N_f(p)}(w_{ft}^{kp} - w_{f,t-1}^{kp}) &\le S_k^*(t) \quad \forall k \in \mathcal{K}, t \in \mathcal{T}, p \in \mathcal{P}_f,
\end{aligned}$$

where we have:

$$\begin{aligned}
D_k^*(t) &\triangleq \min\{D_k(t) \,:\, (D(t), A(t), S(t)) \in \Omega\} \\
A_k^*(t) &\triangleq \min\{A_k(t) \,:\, (D(t), A(t), S(t)) \in \Omega\} \\
S_k^*(t) &\triangleq \min\{S_k(t) \,:\, (D(t), A(t), S(t)) \in \Omega\}.
\end{aligned}$$

In the robust formulation, therefore, the correlation information captured by the full uncertainty set Ω is lost. We remark that this is not an artifact of the formulation that could be alleviated by a change of variables similar to the portfolio optimization problem of [13] discussed in Chapter 4. It is, rather, an intrinsic consequence of the physical problem. The solution to any robust formulation is a schedule over time, i.e., a trajectory for all the aircraft, that does not violate any of the sector capacity constraints for any sector, at any point in time, for any realization of $(D(t), A(t), S(t)) \in \Omega$. Any single trajectory that satisfies this, will be feasible to the robust constraints given above.

The nature of the weather uncertainty that actually impacts sector capacities is important for our robust formulation, and our subsequent development of an adaptable model. We discuss this in the next section.

■ 5.4.4 Adaptability

We assume that current weather conditions, i.e., the instantaneous capacities of every sector, are known deterministically. The weather forecast's accuracy decreases, however, as we move out in time. For our purposes, the important question is to understand the nature of the uncertainty in the weather prediction. Our belief that finite adaptability, as developed in Chapter 3 is an appropriate adaptability model for the Air Traffic Control problem, is based on the claim that while the capacity uncertainty is high dimensional, in the sense that there are over 500 sectors whose capacity is affected by the uncertainty, in fact there are only a few parameters of uncertainty, that control the real impact of the weather on air traffic control. Local weather variations are independent of each other, and indeed variations in very localized weather do make up a full high dimensional uncertainty set. However, these local weather variations are not the weather phenomena that severely impact air traffic. Thus we focus on large storm fronts, that have the capability of disrupting the flow of the air traffic over the NAS. These weather fronts typically are accurately predicted hours in advance.

The uncertainty in such strong storm fronts comes in the actual time of arrival, and also in small perturbations in location, direction, and intensity. That is to say, there is indeed a small parameter set that successfully captures the main uncertainty in the weather forecasts of storm fronts big enough to impact air traffic.

Consider now the vector of capacity constraints. We have a constraint for every sector of the NAS, as well as one representing takeoff and landing for each of the airports. The uncertainty set represents the variation in the capacity vector, and therefore it is a subset of $\mathbb{R}^M$, for M equal to the dimension of the capacity vector, and thus $M \approx 500$. However, if the main uncertainty in the weather is captured by the few parameters mentioned above (arrival time, intensity, direction and location) this effectively means that the variation in the vector of sector capacity constraints is very tightly correlated. In a sense, the uncertainty set is effectively very low dimensional, and not a full M-dimensional set. However, this correlation is across constraints. In the context of the discussion of uncertainty sets in Chapter 3, the uncertainty set is far from being rectangular. Therefore the pure static robust approach is unable to take advantage of this correlation.

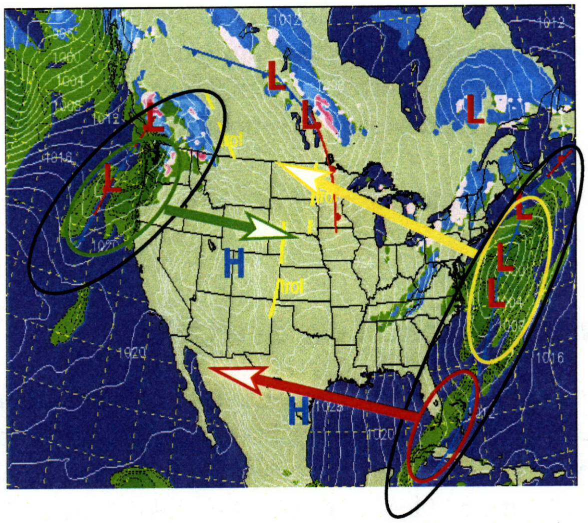

Figure 5-2. This figure gives a pictorial representation of three weather fronts over the NAS. The arrows represent the forecasted direction of travel of the weather fronts. While this forecast may not be exact, the central claim on which we base our modeling choice for adaptability, is that the main factors of inaccuracy in the prediction will be in the speed of the storm fronts, and perhaps of their exact direction. Therefore the uncertainty in the capacities of the underlying sectors, will be very tightly correlated. If the storm front arrives 30 minutes ahead of the forecasted schedule, the capacity of a very large number of the sectors will vary in a tightly correlated manner.

Let us now recall our results from Chapter 3. There we saw that when the dimension of the uncertainty set is small, finite adaptability is theoretically tractable. Furthermore, we saw in our simulations, that for low-dimensional uncertainty sets, finite adaptability can cover a significant percentage of the gap between static and fully adaptable formulations, with only a small number of regions in the partition.

It is further important to stress that the air traffic control problem is inherently an integer optimization problem, since the decisions involve indivisible quantities (planes). Furthermore, while the aggregate traffic is quite heavy, with over 30,000 commercial flights per day, there are many routes where the number of planes at any given time are very low. Therefore, there does not seem to be a reason to believe that a continuous approximation of the planes would lead to anything that is in fact easily implementable.

Finite adaptability, however, is specifically designed to handle multistage problems with discrete variables.

We have outlined, therefore, three aspects of the problem that lead us to implement finite adaptability. First, the problem has $\{0,1\}$ variables, and therefore continuous adaptability schemes are simply not applicable. Second, the effective uncertainty set is low-dimensional, and therefore from our results in Chapter 3, the finite adaptability problem is theoretically tractable, and furthermore we expect it to be able to improve the static solution with only a few partitions. Finally, the starting point is a very large scale nominal problem. The large scale nature essentially limits the class of approaches one can take. A sampling approach, for instance, seems impractical, since even the most optimistic numbers for the sample complexity seem to drive the number of constraints into the *billions*.

It is precisely because of these three aspects of the problem that we implement finite adaptability.

■ 5.5 Computational Results: Two Scenarios

In this section, we give two examples that illustrate the applicability of our formulation above, and the advantages of finite adaptability. Both examples represent two-stage problems, with multiple planes traversing several sectors to arrive at a single airport. First, we consider the utopic set-up. Here, the decision-maker knows the realization of the weather in advance of making any decisions. This is not implementable, because it requires advance knowledge of the weather. It represents an upper bound on the best performance any adaptability scheme could achieve. After this we consider four schemes which are implementable. First, we consider a nominal approach, i.e., one without any adaptability, but rather where the first stage solution is constructed assuming the weather follows a (single) deterministic trajectory. Then we consider the robust approach, where a single solution with no adaptability is designed to be robust to the possible uncertainty in the weather forecast. Finally, we consider two levels of finite adaptability: 2-adaptability and 4-adaptability.

5.5.1 Example 1: Ground Holding

In this example we illustrate how finite adaptability can significantly decrease the impact of a storm on a single-airport landing problem. Here, we consider a major airport such as JFK international airport in New York, that accepts heavy traffic from airports to the West, like Chicago, San Francisco, and Los Angeles, and also from airports to the South, such as Dulles, National, and Atlanta. We consider the situation where the weather forecast predicts that the capacity of the approach to JFK will be severely impacted by an approaching storm. Because of timing uncertainty, it is not known exactly when the storm will affect the western, and when the southern approaches. The general picture of this phenomenon is illustrated in Figure 5-3. There are some number of planes scheduled to land in JFK around the time the bad weather is forecast to impact the capacities of the final approach to JFK. The central scheduler must decide whether to keep JFK-bound planes on the ground at their airport of origin, or send them further along their route.

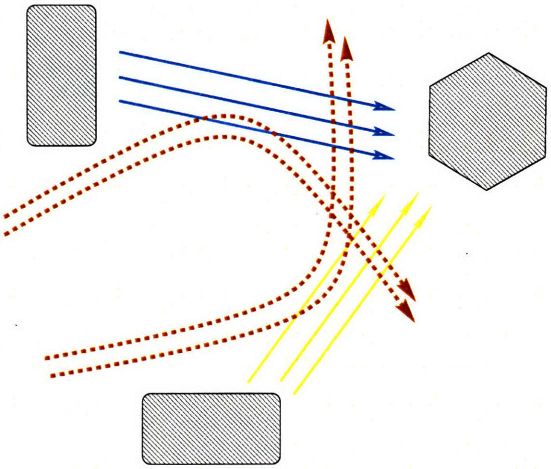

Figure 5-3. This figure gives the scenario we consider in the first Air Traffic Control example. We have planes arriving at a single hub such as JFK in NYC. Planes arrive to JFK from the West, and also from the South. The uncertainty in the weather is expressed in the dashed lines. This is meant to illustrate that while the forecast predicts the presence of a disruptive storm, there is uncertainty in the exact time the impact will be felt in the air corridors. In this example, the uncertainty produces increased ground holding in the robust model, and increased air holding in the adaptable model.

We make the following simplifications, so that we can present a clear experiment that illustrates the benefits of adaptability. We consider a three hour interval. We assume that the storm is scheduled to impact the capacity of the sectors in the approach to JFK, in the second hour of our horizon, and that regardless of when it moves over a particular sector, the impact does not last more than 30 minutes. We assume that there are 50 planes approaching from the West, and 50 planes approaching from the South. If these planes are not held on the ground, or in the air, they arrive at JFK in two hours.

Each plane may be held either on the ground, or in the air, or both, for a total delay not exceeding 60 minutes. Therefore all 50 planes from the West, and 50 planes from the South, will have landed by the end of the three hour window under consideration. The simplified picture is presented in Figure 5-4. Here, the rectangular nodes represent the airports, and the self-link ground holding. The intermediate circular nodes represent a location one hour from JFK, in a geographical region whose capacity is unaffected by the storm. The self-link here represents air holding. The final hexagonal node represents the destination airport, JFK. The links from the two circular nodes to the final hexagonal node are the capacitated links.

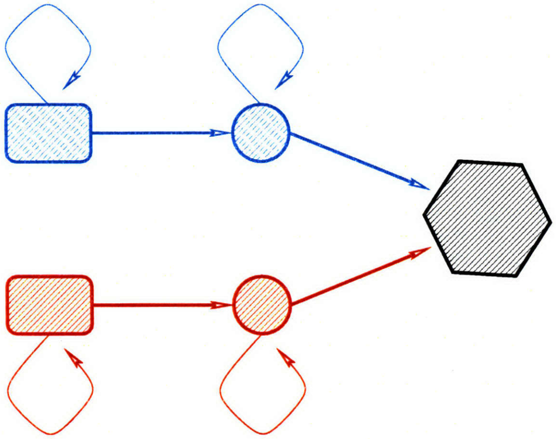

Figure 5-4. This figure gives the simplified version for the scenario we consider in the first Air Traffic Control example. We model the planes originating from the West, as scheduled to depart at the same time from a single airport, and similarly for the planes arriving to JFK from the South. The hexagonal node represents JFK airport. The two circular nodes represent the intermediate sectors where the planes may air hold, if the storm has reduced capacity to the point that they cannot proceed to the final hexagonal node. The two rectangular nodes represent the initial airports, and the self-loops represent ground holding.

We discretize time into 10-minute intervals. We assume that the impact of the storm lasts 30 minutes. The uncertainty is in the timing of the storm, and the order in which it will affect the capacity of the southward and westward approaches. There is essentially a single continuous parameter here, controls the timing of the storm, and whether the most severe capacity impact hits the approach from the south before, after, or at the same time as it hits the approach from the west. Because we are discretizing time into 10 minute intervals, there are four possible realizations of the weather-impacted capacities in the second hour of our horizon. These four scenarios are as follows. We give the capacity in terms of the number of planes per 10-minute interval:

$$\begin{array}{cl}
(1) & \begin{bmatrix} \text{West:} & 15 & 15 & 15 & \underline{5} & \underline{5} & \underline{5} \\ \text{South:} & \underline{5} & \underline{5} & \underline{5} & 15 & 15 & 15 \end{bmatrix} \\
(2) & \begin{bmatrix} \text{West:} & 15 & 15 & \underline{5} & \underline{5} & \underline{5} & 15 \\ \text{South:} & 15 & \underline{5} & \underline{5} & \underline{5} & 15 & 15 \end{bmatrix} \\
(3) & \begin{bmatrix} \text{West:} & 15 & \underline{5} & \underline{5} & \underline{5} & 15 & 15 \\ \text{South:} & 15 & 15 & \underline{5} & \underline{5} & \underline{5} & 15 \end{bmatrix} \\
(4) & \begin{bmatrix} \text{West:} & \underline{5} & \underline{5} & \underline{5} & 15 & 15 & 15 \\ \text{South:} & 15 & 15 & 15 & \underline{5} & \underline{5} & \underline{5} \end{bmatrix}
\end{array}$$

We compare several different adaptability schemes. In the utopic set-up, the decision-maker knows even before the first stage decision is implemented, which of the four potential weather scenarios will be realized. Because of the advance knowledge requirement, this is not implementable. Rather, it provides a bound on the best possible performance.

Next, we consider a nominal, no-robustness scheme. Here, the decision-maker simply assumes that the realization will be one of the four predicted scenarios (we choose the first).[2] The first stage decision is implemented. Then, the true realization is revealed, and the decision-maker must act accordingly in the second stage. If the realization turns out to be the first uncertainty realization, then the decision-maker gets lucky, and the final cost matches that of the utopic solution. If this is not the case, however, the final cost may be considerably worse: in fact it is, as we see below.

Next, we consider robust adaptability formulations. First, we consider the static scheme. Here, the decision-maker chooses a single solution that is feasible for any allowable realization of the weather. Next we consider the 2-adaptable solution, and the 4-adaptable solution. Recall from Chapter 3, that for k-adaptability, the decision-maker selects a decision to implement in the first stage, and k decisions to implement in the second stage. Thus, for the 2-adaptable solution, the decision-maker computes two potential schedules for the second stage, so that at least one of them will be feasible, regardless of the realization of the weather. This is in contrast to the static formulation, where the decision-maker must choose a single plan that is feasible for all possible weather realizations.

The cost is computed from the total amount of ground holding and the total amount of air holding. Each 10-minute interval that a single flight is delayed on the ground, contributes 10 units to the cost. Each 10-minute interval of air-delay contributes 20 units.

Note that the structure of the static and utopic solutions is qualitatively different

[2] Since the decision-maker assume that the weather evolution is perfectly known, there is no advantage to using any adaptability in computing the first-stage decision.

	Delay Cost	Ground Holding	Air Holding
Utopic:	2,050	205	0
Static:	3,900	390	0
2-Adaptable:	3,300	170	80
4-Adaptable:	2,900	130	80

Table 5.1. In this table we give the results for the cost of total delay, as well as the total ground-holding time, and air-holding time, for the utopic, robust, 2-adaptable, and 4-adaptable schemes, for the Air Traffic Control example described in Section 5.5.1. The ground- and air-holding time is given as the number of 10 minute segments incurred by each flight (so if a single flight is delayed by 40 minutes, it contributes 4 to this count).

than the adaptable solutions. In the adaptable solutions, the adaptability allows the decision-maker to take a "more optimistic" first-stage solution, that is, to send more planes in the air. This ultimately results in a larger air-holding cost, but considerably decreased ground-holding. It is this gain that leads to a reduced overall cost.

Next, we give the delay cost for the nominal problem, where the decision-maker ignores robustness, assumes that the first potential weather realization above is in fact the true realization, and implements the corresponding optimal first-stage solution. In Table 5.2, we give the cost depending on what the actual realization turns out to be. Note that if the second weather scenario above is the true realization, then the corresponding cost of the nominal solution is 2,950, which exceeds the 4-adaptable solution cost. If the third scenario is the true realization, then the cost of the nominal solution is 3,950, which is more than the static robust formulation, and considerably higher than the cost of the 2-adaptable and 4-adaptable solutions. Finally, if the true realization happens to be the fourth scenario, then the cost of the nominal solution is 4,750, which is much higher than the cost of the robust, 2-adaptable, or 4-adaptable solutions.

	Realization 1	Realization 2	Realization 3	Realization 4
Nominal Cost:	2,050	2,950	3,950	4,750

Table 5.2. In this table we give the results for the cost of total delay for each scenario, when the first-stage solution is chosen without robustness considerations, assuming that the first realization is in fact the true realization.

■ 5.5.2 Example 2: Route Selection

The example in this section is similar to the example in Section 5.5.1, but here we illustrate the benefit of adaptability in making good flight selection decisions in the early stages.

In this set-up, flights from a single origin must be assigned not only ground holding and air holding, but also a particular route to their final destination. The general setup is depicted in Figure 5-5. There, we consider the situation where flights arrive at a single airport from a single direction. Any particular flight in general may have several different opportunities to select a path to its destination. The general phenomenon we wish to capture is when adaptability allows less pessimistic routes to be selected, even in the presence of potentially bad weather. The basic set-up is that there is a conservative route, that completely avoids the storm front. However, if the storm front is not particularly intense, typically planes are able to find a way through the storm, that is not apparent from earlier forecasts (see, e.g., the schedules and actual trajectories as shown in detail in the various studies contained in [111]).

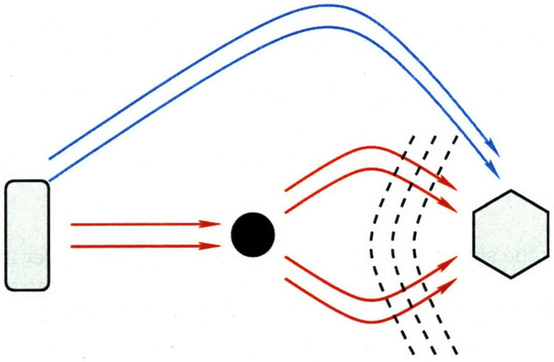

Figure 5-5. This figure gives the scenario we consider in the second Air Traffic Control example. It models the situation where planes arrive from a common direction, towards a single destination airport, e.g., JFK. The weather uncertainty in this case affects the capacity of some of the routes towards JFK. The approaching planes have the option of flying towards the storm in the hope that they will find some way around or through the storm, or to divert their path early on, making absolutely certain to avoid the footprint of the storm front. This latter alternative is represented by the longer paths that arch well above the center of the figure.

As for our first example, we model this set-up with a simplified model. Consider Figure 5-6. We assume that all the planes originate at the same airport node. This is the square node at the left of the figure. At this point, a plane has three choices: to ground hold, to continue straight to the nearest circle-node, or or to follow the path to the highest circle-node. This path represents the conservative path that seeks to entirely avoid the storm. Note now, that if the decision-maker selects to go forward, following the shorter path, then there is another point where a directional decision must be made, either to follow the top branch, or the bottom branch. We model a storm that has some partial affect on both of these branches. There is some uncertainty, however, in the timing of when the storm affects which branch.

Similarly to the previous example, we consider a three hour horizon, and assume that the weather impacts the capacity of the two inner branches at some point during the second hour. Again we use a single parameter uncertainty set, essentially the same as the one used in the previous example. Again we consider storms whose impact in a particular sector can be at most 30 minutes. In this case, then, we obtain the same four weather realization scenarios as before.

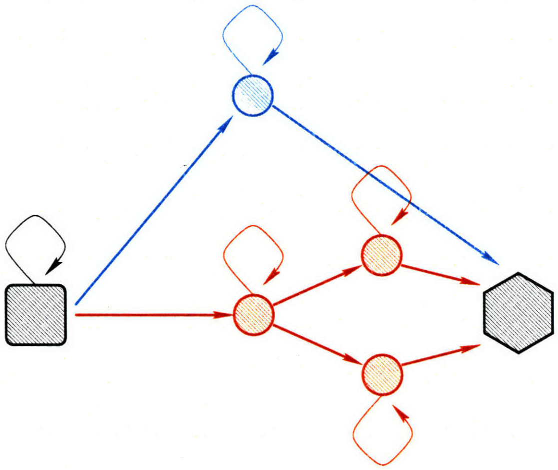

Figure 5-6. This figure gives the simplified version of the scenario we consider in the second Air Traffic Control example. We model the planes arriving from the West as though they arrive from a single airport, in this figure represented by the square node at the left of the figure. The path through the circular node at the top of the figure represents the conservative path that entirely circumvents the storm. The hexagonal node again represents the common destination airport, JFK.

Now, in addition to reporting the ground- and air-holding, we are interested in the number of planes that are sent along the conservative route that entirely avoids the weather front, and the number that take the shorter, or more optimistic path through the storm. We consider 35 planes. As in the previous example, ground holding costs 10 units per plane, per 10 minute interval, and air holding 20 units. In addition to this, we incur a cost of 10 units for each plane traveling on the first edge of the conservative path, and 10 units for the second edge of that path.

In this example, finite adaptability not only improves the static solution significantly, but in fact it is able to attain the performance of the utopic solution. In the utopic, and 2-, and 4-adaptable solutions, no planes are directed to the conservative path that seeks to fly well out of the footprint of the storm. The static robust solution needs to send 5 out of the 35 flights along this path, and still requires twice the ground holding used by the adaptable solutions.

■ 5.6 Conclusions

This chapter considered the problem of scheduling ground delay, air delay, and routing for the global air traffic management problem. The two new ingredients we intro-

	Delay Cost	Ground Holding	Air Holding	C-Route
Utopic:	150	15	0	0
Static:	400	30	0	5
2-Adaptable:	150	15	0	0
4-Adaptable:	150	15	0	0

Table 5.3. In this table we give the results for the cost of total delay, as well as the total ground-holding time, air-holding time, and number of flights directed along the most conservative path, for the utopic, robust, 2-adaptable, and 4-adaptable schemes, for the Air Traffic Control example described in Section 5.5.2. As in Table 5.1, the ground- and air-holding time is given as the number of 10 minute segments incurred by each flight (so if a single flight is delayed by 40 minutes, it contributes 4 to this count).

duce are the uncertainty in the weather, and and the sequential aspect of the weather forecast. Our proposal builds in robustness to the scheduling problem. By using finite adaptability, we are able to build in adaptability of future scheduling decisions to updates in the weather forecast. Our computational examples, while relatively small-scale compared to the full-size problem we wish to solve, nevertheless illustrate that robustness is needed in the fact of weather uncertainty, and importantly, that adaptability can greatly mitigate the conservative aspect of adding robustness.

We used the ideas of finite adaptability developed in Chapter 3. The main motivation for this was two-fold: first, because of the integral nature of the scheduling decisions, continuous adaptability schemes such as affine adaptability, do not seem to be applicable. Second, because of the large-scale nature of the nominal problem, there is no room to increase the number of variables or constraints by any appreciable level. Finite adaptability is appropriate on both counts. It is able to accommodate the discrete nature of the variable, and also the number of variables and constraints can be controlled. This is particularly true because of the low-dimensional nature of the uncertainty set; indeed, this is what makes finite adaptability particularly attractive for this application.

CHAPTER 6

Conclusions and Future Work

Adaptability of future decisions on past realizations of the uncertainty, is the central theme of this dissertation. The central ingredients in a multistage optimization problem with parameter uncertainty, is the model for the uncertainty, and the model for the adaptability. As we have shown in this thesis, the interaction of these two ingredients is of utmost importance, for obtaining effective and efficient algorithms.

In the first part of this thesis (Chapter 3), we considered a robust noise model. Motivated by the need to accommodate discrete variables, we developed a piecewise constant approach which we called Finite Adaptability. The main features of our proposal are that it can, as designed, accommodate discrete variables, and that it presents a hierarchy of adaptability. Under certain circumstances, such as when the dimension of the uncertainty is small, or the dimension of the problem is small, or the number of constraints affected by uncertainty is small, we prove tractability results. Even in the general case when all three of these do not hold, we are able to derive necessary conditions that can guide us in choosing good partitions, based on geometric considerations of duality.

The second part of this thesis takes a different approach to uncertainty, assuming that it is stochastic. Here, the assumed stochastic nature of the uncertainty not only specifies the problem for us, but also serves as the judge of "how good" our final solution is. Here we show that by controlling the structure of the adaptability, we are able to control the number of samples required for multi-stage optimization. In particular, we are able to obtain polynomial bounds on the sample complexity, for arbitrary number of stages.

One of the fundamental themes of this thesis has been an attempt to strike a balance between performance and tractability. Robust Optimization has recently received a surge of attention, precisely because of its appealing tractability properties in a fairly wide class of convex optimization problems. Apparently, this tractability does not seem to immediately extend to the multi-stage setting. Nevertheless, we believe that tractability considerations should continue to be a strong guide to direct future research.

A second important theme of this thesis has been the idea that successfully balancing tractability and performance, requires managing the two main elements, uncertainty and adaptability, within the framework of the optimization problem at hand. It is our belief that this strongly argues for a unified treatment of statistics and optimization. Indeed, the adaptability of future stage decisions on past uncertainty gets to the core of optimization, estimation, and learning.

This thesis represents ultimately our initial efforts to address these issues. In the remainder of this chapter, we outline what we believe to be some important open questions and directions, towards continuing this work, along the two main themes outlined in the paragraphs above.

■ 6.1 Robust Optimization

The primitives of the robust optimization noise model, are the uncertainty sets themselves, as these define the uncertainty of the problem. The question of where these sets arise, or how they should be chosen, is hardly touched upon in this thesis. While there has been some work on constructing uncertainty sets from data (see [41]), this important question is still largely unexplored. This is particularly true in the case of multistage problems and adaptability. Throughout this thesis, we have not addressed the possibility that the uncertainty sets may themselves depend on the action taken. This is an important setup, as it seems quite plausible that different implemented actions could very well expose the system to different sets of uncertainty.

This further points to the concept of some structured uncertainty. In this thesis, and more generally in the robust optimization literature, the focus has been on parameter uncertainty that is essentially full dimensional (or perhaps restricted to lie in some subspace). The dynamic nature of multistage optimization makes the consideration of more complicated structural forms of uncertainty an important (and unexplored) area of research.

■ 6.2 Dealing With Data

The work in Chapter 4 incorporates the data directly into the optimization. That is, there is no separate estimation phase, where one uses some procedure to extract an estimate of some parameters from a data sample, independently from the optimization, and then in turn uses these parameter estimates to solve an optimization. The high-level point is an obvious one, namely, there is no separation between estimation and optimization, meaning that given a data sample, any estimate of parameters should not be done in a vacuum, but rather the estimation procedure should depend on the optimization problem. We have avoided this altogether, by simply using the data directly in the optimization. We pay a price, however, in that the proposed method does not scale: the effort required to solve a problem is related in an albeit polynomial, but nonetheless nonfavorable way on the size of the data sample. A procedure that, say, retains only an unbiased estimate of the first two moments of the data, and then uses those estimates to solve an optimization problem, does not suffer from this dependence on the size of the data sample, since the optimization problem solved depends only on the two moment estimates.

Designing the estimation procedures, as well as the collected statistics themselves, within the context of the optimization problem to be solved, seems like an important question. This is particularly important in view of the cost of incorporating all the data directly into the optimization. While the sample complexity bounds are indeed polynomial, they can in practice increase the size of a problem unacceptably. Indeed, even 1,000 samples takes an easily solvable linear optimization with, say, 500 inequalities, and produces a very large scale problem with half a million constraints.

■ 6.3 The Feature Functions and Regularization

In Chapter 4, we considered adaptability defined by maps to potentially high dimensional feature spaces. The possibility of using the covering number and fat-shattering approach of Section 4.5, in conjunction with regularization, may free us from using the dimensionality of the feature mappings as a proxy for the complexity, instead permitting the use of other notions of regularity. While potentially appealing to use maps to much higher dimensional spaces (potentially even to infinite dimensional spaces, as is common in the Reproducing Kernel Hilbert Space (RKHS) literature in classification (e.g., [64], [115])) this seems to lead to a formulation where both the number of constraints, and variables are dependent on the number of samples. If there are efficient column and constraint generating approaches to such problems, the door would be open to pursuing a richer class of mappings, without having to explicitly specify the finite dimensional image of the feature mapping F. In the absence of such procedures, however, it seems that choosing good low-dimensional functions, F, is of utmost importance, since the dimensionality controls the number of variables, and thus indirectly, the sample complexity and therefore the number of constraints as well.

■ 6.4 Structural Risk Minimization

The sample complexity bounds of Chapter 4 quantify how the reliability and feasibility guarantees degrade when, for a fixed number, N, of samples, we increase the level, or complexity of the adaptability. Recall from that chapter, that the level of adaptability is essentially encoded into the dimension of the image of the feature functions F.

As the empirical results in that chapter suggest, however, the sample complexity bounds do not tell the full story, as far as feasibility goes. Indeed, the results there indicate that while higher levels of adaptability may have worse sample-complexity based performance guarantees, nevertheless the solutions produced may have better feasibility properties than solutions generated by lower levels of adaptability. The idea is that adaptability may allow the optimization to exploit structure in the uncertainty data. We illustrate this further by means of a very simple network design problem. Consider the tandem 3-node network shown in Figure 6-1.

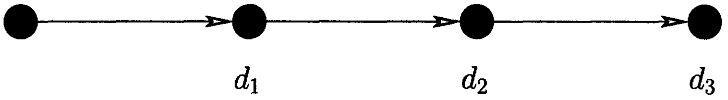

Figure 6-1. This figure shows a tandem network with three nodes. We consider the two-stage network design problem, with demand uncertainty at the second, third, and fourth nodes.

As in Section 4.6.1 in Chapter 4, we assume there is uncertainty in the demand. In this case, let us assume a simple model for the uncertainty: suppose that the samples of the three demand vectors, (d_1, d_2, d_3) are drawn uniformly and at random from the simplex:

$$\left\{(d_1, d_2, d_3) \;:\; 0 \leq d_i \leq 1, \sum_i d_i = 1\right\}.$$

If the number, N, of samples is large, the optimal static robust policy will be to distribute the capacity to the three edges as in Figure 6-2.

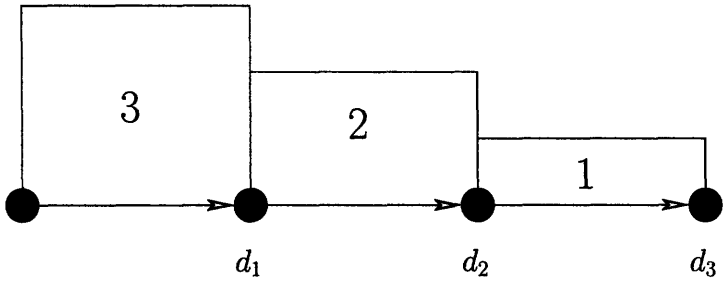

Figure 6-2. This figure shows the static solution to the tandem network with three nodes.

Given the same budget for a first-stage solution, namely, 6 capacity units, the optimal allocation, given the structure of the uncertainty set, is evenly: $(c_1, c_2, c_3) = (2, 2, 2)$. Indeed, in this case, the solution will be deterministically feasible to noise from the same source, that is amplified 100%. Note that the static solution will be feasible to this amplified noise with probability no more than 50%.

The principle of Structural Risk Minimization (see, e.g., [138], [121], and references therein, for SRM in the context of statistical learning) seeks to balance the benefits of increased adaptability in future stages, against the possibility of over-fitting, since the number of sample points available is taken to be fixed. The bounds for the sample complexity control the risk of over-fitting. What we are missing from this picture, yet is apparent empirically from our examples, is an expression for the benefits of adaptability for feasibility of the first stage decision. A natural candidate for performance on the training data is the empirical feasibility of the solution. Certainly, empirical feasibility increases as adaptability increases. The sample complexity also increases, however. Therefore given a fixed number, N, of samples, as the level of adaptability increases, the empirical error decreases, but the guarantees that this empirical error is a close approximation of the true level of feasibility, deteriorate. In a nutshell, the principle of structural risk minimization require the balance of the improved performance on the training error, i.e., the improved empirical feasibility in our case, with the deteriorating bounds on the closeness of the empirical feasibility to the true feasibility. The first term improves with increasing adaptability, while the second deteriorates.

The difficulty in this case, is that it is intractable in general to compute the empirical feasibility in the context of optimization. That is, given uncertainty realizations $\omega^{(1)}, \dots, \omega^{(N)}$, if there is no solution feasible to all these realizations simultaneously, it is in general NP-hard to compute a solution that is feasible to the maximum number of these. Nevertheless, it seems like an interesting and worthwhile endeavor to try to obtain natural and meaningful bounds on these quantities. This would serve as an *a priori* guide as to what level of adaptability might be appropriate for different applications.

CHAPTER A

Simulation and Computation Details

In this appendix we give the details of the computations and simulations performed in Chapters 3, 4, and 5.

All the computations were performed using Matlab [128], AMPL and CPLEX [2], and also the package CPLEXINT [131], which provides a very convenient interface between Matlab and CPLEX. All computations in all chapters were run on a PC Pentium IV with a 2.4 Ghz processor, and 1 Gb RAM.

CHAPTER B

Scale Sensitive Complexity

In this appendix, we give some additional background on learning with real-valued functions (as opposed to $\{0,1\}$-valued functions). This allows us to complete the proof of Proposition 4.8 from Chapter 4.

B.1 Function Complexity Background

In the background section of Chapter 2, we introduced the complexity of a class of sets, $\mathcal{C}$, as given by the growth function. Recall that the growth function measures the maximum cardinality of different ways a collection of m vectors can be classified by the binary classifier rules specified by the sets $C \in \mathcal{C}$.

If we view the sets $C \in \mathcal{C}$ as indicator functions,

$$\mathbb{I}_C(x) = \begin{cases} 1, & \text{if } \omega \in C \\ 0, & \text{otherwise} \end{cases}$$

then the growth function computes the maximum cardinality of the image of m points, under the collection of indicator functions corresponding to the sets $C \in \mathcal{C}$.

This measure of complexity of the set of indicator functions is also useful for understanding the complexity of a set of continuous functions. For continuous functions, however, it no longer makes sense to consider the cardinality of the image, since the image will typically be continuous. Instead, we replace the cardinality with the notion of a covering number. There has been extensive work done on covering numbers in Functional Analysis, and learning theory. We refer the reader to [57], [136], [74], and references therein, and provide only a brief discussion here.

Given a subset S of Euclidean space with metric d, the γ-covering number of S with respect to d, denoted by $\mathcal{N}(\gamma, S, d)$, is the smallest number of points, $s_1, \ldots, s_k$, such that any point in S is at most a distance γ from some s_j. For indicator functions, we considered the cardinality of the image of a collection of points, $\omega^{(1)}, \ldots, \omega^{(m)}$, under all possible indicator functions in our set. Now consider a class $\mathcal{F}$ of continuous functions. Here we consider all possible mappings of the points $\omega^{(1)}, \ldots, \omega^{(m)}$ under all functions $f \in \mathcal{F}$, and instead of the cardinality of this set (which is typically infinite) we ask for the covering number. In symbols, we consider:

$$F|_{(\omega^{(1)},\dots,\omega^{(m)})} \triangleq \{(f(\omega^{(1)}),\dots,f(\omega^{(m)})) \,:\, f \in \mathcal{F}\} \subseteq \mathbb{R}^m,$$

and then the covering number of the set $F|_{(\omega^{(1)},\dots,\omega^{(m)})}$. Now we can define:

Definition B.1

The covering number of a function class $\mathcal{F}$, for a given m and γ, is the maximum size of the covering number, $\mathcal{N}(\gamma, F|_{(\omega^{(1)},\dots,\omega^{(m)})}, d)$, where we take the maximum over all $(\omega^{(1)},\dots,\omega^{(m)}) \in \Omega^m$, and where d denotes the a metric, typically the Euclidean or ∞-metric.

Note that this is exactly parallel to the development of the growth function. The main difference, however, is the additional parameter, γ. Unlike the growth function, the covering number depends on this parameter γ, and therefore is called a scale-sensitive dimension.

The growth function controls the probability of error in empirical error minimization algorithms for classification. For classification by indicator functions, a point is either correctly or incorrectly classified. When we use real-valued functions, however, we classify according to some threshold, say, if the real-valued label attached to a point is above or below $1/2$, the corresponding point is then labeled $+1$ or 0, respectively. But then the notion of "how close" a correctly classified point is to misclassification, becomes meaningful. This closeness is called the margin. Scale-sensitive measures control the probability of error for algorithms using real-valued functions, that use a margin definition for the error. Thus, for a γ-margin criterion for the error, a point is considered misclassified if its real-valued label is within γ of the threshold point, $1/2$.

Again this allows us to compute an empirical error of a particular real-valued function classifier, and then to use uniform convergence results to obtain bounds on how far the true error is from the empirical error. The difference is in the definition of the empirical error, as now it explicitly involves the margin.

For more on learning with real-valued classifiers, we refer the reader to Part II of [5].

■ B.2 A Sample Complexity Result

In this section prove Proposition 4.8. The proposition considers the robust sampled learning problem, where feasibility no longer requires feasibility to the sampled points in Ω_N, but instead requires the more stringent condition of feasibility for every realization of the uncertainty within a distance γ of every sampled point in Ω_N. Recall the statement of the proposition.

Proposition B.1
Consider a two-stage linear optimization problem, and structured adaptability function $\boldsymbol{y}(\omega)$:

$$\begin{aligned} \min: \quad & \boldsymbol{c}^{\top}\boldsymbol{x} \\ \text{s.t.}: \quad & \boldsymbol{A}(\omega)\boldsymbol{x} + \boldsymbol{B}(\omega)\boldsymbol{y}(\omega) \leq \boldsymbol{b}. \end{aligned}$$

Let Ω_N denote N random samples of ω, and let $\mathcal{XY}_N(\gamma)$ denote the set defined above, of pairs of solutions feasible for the robustified sampled. Then with probability at least $(1-\delta)$,

$$\sup_{\{(\boldsymbol{x},\boldsymbol{y}(\cdot))\in\mathcal{XY}_N(\gamma)\}} \mathbb{P}(\boldsymbol{A}(\omega)\boldsymbol{x} + \boldsymbol{B}(\omega)\boldsymbol{y}(\omega) - \boldsymbol{b} > 0) \leq \varepsilon,$$

as long as the number of samples $N = N(\varepsilon, \delta, \gamma)$ satisfies

$$N(\varepsilon, \delta, \gamma) \geq \frac{4}{\varepsilon}\left(V_\gamma \ln\frac{12}{\varepsilon} + \ln\frac{2}{\delta}\right),$$

where V_γ denotes the fat shattering dimension of the collection of sets

$$\hat{C}_{(\boldsymbol{x},\boldsymbol{y}(\cdot))} = \{\omega \in \Omega : \boldsymbol{A}(\omega)\boldsymbol{x} + \boldsymbol{B}(\omega)\boldsymbol{y}(\omega) - \boldsymbol{b} \leq 0\}.$$

PROOF. The main difference between this result, and Proposition 4.5, is that the feasible set has been reduced because of the stricter feasibility requirement, and the VC dimension in the sample complexity, has been replaced by the fat-shattering dimension of the family of sets, $C_{(\boldsymbol{x},\boldsymbol{y}(\cdot))}$.[1] The fat-shattering dimension is traditionally defined for sets of functions. The extension to families of sets is natural in our context. Indeed, given a family of sets, $\mathcal{C}$, define the family of $[-1,1]$-valued functions as follows:

$$\mathcal{F}_{\mathcal{C}} \triangleq \{f_C(\cdot) = [\min(d(\cdot, C), 1) - \min(d(\cdot, C^c), 1)] \,:\, C \in \mathcal{C}\}.$$

Then the fat-shattering dimension of $\mathcal{F}_{\mathcal{C}}$ measures exactly the ability of the class of sets in $\mathcal{C}$ to separate not single points as is the definition for the VC-dimension, but γ-balls, as illustrated in Figure 4-6 in Chapter 4. This is because by construction, the functions in the family have Lipschitz constant equal to 1, and hence the margin in the input, i.e., with respect to ω, is equal to the margin in the output space, $[-1,1]$. From this point, and from the fact that we assume that the set

$$\mathcal{XY}_N(\gamma) \triangleq \left\{ (\boldsymbol{x},\boldsymbol{y}(\cdot)) : \begin{array}{r} \boldsymbol{A}(\omega)\boldsymbol{x} + \boldsymbol{B}(\omega)\boldsymbol{y}(\omega) - \boldsymbol{b} \leq 0, \quad \forall \omega \in B_\gamma(\omega^{(1)}) \\ \vdots \\ \boldsymbol{A}(\omega)\boldsymbol{x} + \boldsymbol{B}(\omega)\boldsymbol{y}(\omega) - \boldsymbol{b} \leq 0, \quad \forall \omega \in B_\gamma(\omega^{(N)}) \end{array} \right\},$$

[1] The notational extensions for multiple stage adaptability are tedious, but straightforward.

is non-empty, the proof follows from standard learning theory results. For the details, we refer the reader to [4], Theorem 8.3.1. □

References

[1] N. Alon, S. Ben-David, N. Cesa-Bianchi, and D. Haussler. Scale sensitive dimensions, uniform convergence, and learnability. *Journal of the ACM*, 44(4):615–631, 1997. *(Cited on page 130.)*

[2] AMPL and CPLEX. See `http://www.ilog.com`. *(Cited on page 183.)*

[3] G. Andreatta and L. Brunetta. Multiairport ground holding problem: A computational evaluation of exact algorithms. *Operations Research*, 46(1):57–64, 1998. *(Cited on page 160.)*

[4] D. Anthony and N. Biggs. *Computational Learning Theory*. Cambridge Univeristy Press, 1992. *(Cited on pages 54, 56, 127 and 188.)*

[5] M. Anthony and P. Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 1999. *(Cited on pages 54, 130, 132 and 186.)*

[6] A. Atamtürk and M. Zhang. Two-stage robust network flow and design under demand uncertainty. Technical Report BCOL.04.03, IEOR, University of California–Berkeley, December 2004. *(Cited on pages 61 and 62.)*

[7] T. Başar and P. Bernhard. *H^∞-Optimal Control and Related Minimax Design Problems: A Dynamic Game Approach*. Birkhäuser, Boston, MA, 1995. *(Cited on page 35.)*

[8] M. Ball, R. Hoffman, A. Odoni, and R. Rifkin. A stochastic integer program with dual network structure and its application to the ground holding problem. *Operations Research*, 51(1):167–171, 2003. *(Cited on page 160.)*

[9] C. Barnhart, P. Belobaba, and A. Odoni. Applications of operations research in the air transport industry. *Transportation Science*, 37(4):368–391, 2003. *(Cited on pages 157, 158, 159 and 160.)*

[10] A. Barvinok. *A Course in Convexity*. American Mathematical Society, 2002. *(Cited on page 133.)*

[11] R. Bellman and S. Dreyfus. *Applied Dynamic Programming*. Princeton Univesity Press, 1962. *(Cited on page 17.)*

[12] A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski. Adjustable robust solutions of uncertain linear programs. *Math. Programming*, 99:351–376, 2003. *(Cited on pages 26, 29, 43, 44, 59, 61, 62, 65, 86, 95, 99, 101, 102, 108, 112, 113 and 149.)*

[13] A. Ben-Tal, T. Margalit, and A. Nemirovski. Robust modeling of multi-stage portfolio problems. *High Performance Optimization*, pages 303–328, 2000. *(Cited on pages 21, 110, 145, 146, 147, 149, 151 and 166.)*

[14] A. Ben-Tal and A. Nemirovski. Robust convex optimization. *Math. Oper. Res.*, 23:769–805, 1998. *(Cited on pages 18, 35, 41 and 60.)*

[15] A. Ben-Tal and A. Nemirovski. Robust solutions of linear programming problems contaminated with uncertain data. *Math. Programming*, 88:411–421, 2000. *(Cited on pages 18, 34, 35, 42, 60, 61, 68 and 113.)*

[16] A. Ben-Tal and A. Nemirovski. *Lectures on Modern Convex Optimization*. MPS-SIAM, 2001. *(Cited on pages 33 and 42.)*

[17] A. Ben-Tal, A. Nemirovski, and C. Roos. Robust solutions of uncertain quadratic and conic-quadratic problems. *SIAM Journal on Optimization*, 13(2):535–560, 2002. *(Cited on pages 18, 35, 39, 40, 44, 102, 108 and 113.)*

[18] D. Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, Mass., 1995. *(Cited on page 33.)*

[19] D. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 2005. *(Cited on page 17.)*

[20] D. Bertsekas and J. Tsitsiklis. *Neuro Dynamic Programming*. Athena Scientific, 1996. *(Cited on page 17.)*

[21] D. Bertsekas, with A. Nedic, and A. Ozdaglar. *Convex Analysis and Optimization*. Athena Scientific, 2003. *(Cited on pages 33 and 37.)*

[22] D. Bertsimas and C. Caramanis. Finite adaptability in linear optimization. Technical Report available from: `http://web.mit.edu/cmcaram/www/`, M.I.T., September 2005. *(Cited on pages 108, 111, 112, 139 and 149.)*

[23] D. Bertsimas, C. Caramanis, and W. Moser. Air traffic flow management: An adaptable robust approach. Technical Report In preparation, M.I.T. and Lincoln Laboratories, 2006. *(Cited on pages 63, 91 and 92.)*

[24] D. Bertsimas, D. Pachamanova, and M. Sim. Robust linear optimization under general norms. *Operations Research Letters*, 2004. *(Cited on page 38.)*

[25] D. Bertsimas and M. Sim. Robust discrete optimization and network flows. *Mathematical Programming Series B*, 98:49–71, 2003. *(Cited on pages 18, 35, 41, 60 and 63.)*

[26] D. Bertsimas and M. Sim. The price of robustness. *Operations Research*, 52(1):35–53, 2004. *(Cited on pages 18, 35, 38, 42, 60 and 113.)*

[27] D. Bertsimas and M. Sim. Robust discrete optimization under ellipsoidal uncertainty sets. Technical Report available from: `http://web.mit.edu/dbertsim/www`, Massachusetts Institute of Technology, 2004. *(Cited on page 35.)*

[28] D. Bertsimas and M. Sim. Tractable approximations to robust conic optimization problems. *Mathematical Programming*, 2005. *(Cited on pages 18, 35, 41 and 42.)*

[29] D. Bertsimas and S. Stock. The air traffic flow management problem with enroute capacities. *Operations Research*, 46(3):406–422, 1998. *(Cited on pages 31, 159, 160 and 161.)*

[30] D. Bertsimas and S. Stock. The traffic flow management rerouting problem in air traffic control: A dynamic network flow approach. *Transportation Science*, 34(3):239–255, 2000. *(Cited on pages 31, 91, 92 and 160.)*

[31] D. Bertsimas and J. Tsitsiklis. *Introduction to Linear Optimization*. Athena Scientific, 1997. *(Cited on page 33.)*

[32] J. Birge. The value of the stochastic solution in stochastic linear programs with fixed recourse. *Math. Programming*, 24:314–325, 1982. *(Cited on page 34.)*

[33] J. Birge. Decomposition and partitioning methods for multistage stochastic linear programs. *Operations Research*, 33(5):989–1007, 1985. *(Cited on page 49.)*

[34] J. Birge and J. Dulá. Bounding separable recourse functions with limited distribution information. *Annals of Operations Research*, 30:277–298, 1991. *(Cited on page 47.)*

[35] J. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer-Verlag, 1997. *(Cited on pages 18, 45, 47, 59 and 104.)*

[36] J. Birge and J.-B. Wets. Computing bounds for stochastic programming problems by means of a generalized moment problem. *Mathematics of Operations Research*, 12:149–162, 1987. *(Cited on page 47.)*

[37] A. Blumer, A. Ehrenfeucht, D. Haussler, and M. Warmuth. Learnability and the vapnik-chervonenkis dimension. *Journal of the ACM*, 36(4):929–965, 1989. *(Cited on page 56.)*

[38] S. Boyd, L. E. Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. Society for Industrial and Applied Mathematics, 1994. *(Cited on page 40.)*

[39] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. *(Cited on pages 33 and 133.)*

[40] L. Brickman. On the field of values of a matrix. *Proceedings of the AMS*, pages 61–66, 1961. *(Cited on page 40.)*

[41] D. Brown. *Risk and Robust Optimization*. PhD dissertation, Massachusetts Institute of Technology, Department of EECS, June 2006. *(Cited on pages 19 and 178.)*

[42] C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998. *(Cited on page 128.)*

[43] G. Calafiore and M. Campi. Decision making in an uncertain environment: the scenario-based optimization approach. *Mathematical Programming*, 102(Series A):25–46, 2005. *(Cited on pages 19, 27, 29, 33, 51, 57, 99, 106, 108, 109, 121, 122, 123 and 125.)*

[44] A. Charnes and W. Cooper. Chance-constrained programming. *Management Science*, 6(1):73–79, 1959. *(Cited on page 48.)*

[45] A. Charnes, W. Cooper, and G. Symonds. Cost horizons and certainty equivalents: an approach to stochastic programming of heating oil. *Management Science*, 4:235–263, 1958. *(Cited on page 48.)*

[46] X. Chen, M. Sim, and P. Sun. A robust optimization perspective of stochastic programming. *Submitted*, 2005. *(Cited on pages 18, 43 and 62.)*

[47] X. Chen, M. Sim, P. Sun, and J. Zhang. Stochastic programming: Convex approximation and modified linear decision rule. *Submitted*, 2005. *(Cited on pages 29 and 43.)*

[48] G. Dantzig. Linear programming under uncertainty. *Management Science*, 1(3-4):197–206, 1955. *(Cited on page 45.)*

[49] G. Dantzig and A. Madansky. On the solution of two-stage linear programs under uncertainty. In *Proceedings of the Fourth Berkeley Symposium on Statistics and Probability*, volume 1, pages 165–176. University of California Press, 1961. *(Cited on pages 60 and 65.)*

[50] D. de Farias and B. V. Roy. The linear programming approach to approximate dynamic programming. *Operations Research,* 51(6):850–865, 2003. *(Cited on page 57.)*

[51] D. de Farias and B. Van-Roy. On constraint sampling in the linear programming approach to approximate dynamic programming. *Mathematics of Operations Research,* 29(3):462–478, 2004. *(Cited on pages 19, 29, 33, 106, 125 and 127.)*

[52] B. Dean, M. Goemans, and J. Vondrák. Approximating the stochastic knapsack problem: The benefit of adaptivity. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science, Rome, Italy,* pages 208–217, 2004. *(Cited on pages 107 and 135.)*

[53] A. Dembo and O. Zeitouni. *Large Deviations Techniques and Applications.* Springer-Verlag, 1998. *(Cited on pages 48 and 51.)*

[54] S. Dokov and D. Morton. Second-order lower bounds on the expectation of a convex function. *Mathematics of Operations Research,* 30(3):662–677, 2005. *(Cited on page 47.)*

[55] J. Doyle, B. Francis, and A. Tannenbaum. *Feedback Control Theory.* Macmillan Publishing Co., 1990. *(Cited on page 17.)*

[56] R. Dudley. Central limit theorems for empirical measures. *Annals of Probability,* 6(6):899–929, 1978. *(Cited on page 127.)*

[57] R. Dudley. *Uniform Central Limit Theorems.* Cambridge University Press, 1999. *(Cited on pages 54, 130 and 185.)*

[58] G. Dullerud and F. Paganini. *A Course in Robust Control Theory: A Convex Approach.* Springer-Verlag, New York, NY, 1999. *(Cited on page 35.)*

[59] H. Edmundson. Bounds on the expectation of a convex function of a random variable. Technical Report Paper 982, The Rand Corporation, 1956. *(Cited on page 47.)*

[60] E. Erdogan, D. Goldfarb, and G. Iyengar. Robust portfolio management. Technical Report CORC TR-2004-11, IEOR, Columbia University, November 2004. *(Cited on pages 21, 62 and 145.)*

[61] E. Erdogan and G. Iyengar. Ambiguous chance constrained problems and robust optimization. Technical Report CORC TR-2004-10, IEOR, Columbia University, September 2004. *(Cited on page 126.)*

[62] E. Erdogan and G. Iyengar. On two-stage convex chance constrained problems. Technical Report CORC TR-2005-06, IEOR, Columbia University, November 2005. *(Cited on page 106.)*

[63] J. Evans and B. Crowe. Personal communication, 2006. *(Cited on page 164.)*

[64] T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. *Advances in Computational Mathematics*, 13(1):1–50, 2000. *(Cited on pages 125 and 179.)*

[65] C. Floudas and V. Visweswaran. *Quadratic Optimization*, pages 217–270. Handbook of Global Optimizaion. Kluwer Academic Publishers, 1994. *(Cited on page 75.)*

[66] S. Floyd and M. Warmuth. Sample compression, learnability, and the Vapnik-Chervonenkis dimension. *Machine Learning*, 21(3):269–304, 1995. *(Cited on page 53.)*

[67] G. Franklin, J. Powell, and A. Emami-Naeini. *Feedback Control of Dynamic Systems*. Addison-Wesley, 1986. *(Cited on page 17.)*

[68] K. Fukuda. CDD. See `http://www.cs.mcgill.ca/~/fukuda/soft/cddman/`. *(Cited on page 85.)*

[69] M. Garey and D. Johnson. *Computers and Intractability*. W.H. Freeman, 1979. *(Cited on pages 26 and 80.)*

[70] L. E. Ghaoui and H. Lebret. Robust solutions to least-squares problems with uncertain data. *SIAM J. Matrix Analysis and Applications*, 18(4):1035–1064, 1997. *(Cited on page 35.)*

[71] L. E. Ghaoui, F. Oustry, and H. Lebret. Robust solutions to uncertain semidefinite programs. *Siam J. Optimization*, 9(1), 1998. *(Cited on pages 18, 35 and 60.)*

[72] M. Goemans and J. Vondrák. Stochastic covering and adaptivity. In *Proceedings of LATIN 2006*, pages 532–543, 2006. *(Cited on pages 107 and 135.)*

[73] D. Goldfarb and G. Iyengar. Robust portfolio selection problems. Technical Report CORC TR-2002-03, IEOR, Columbia University, December 2001. *(Cited on pages 21 and 145.)*

[74] Y. Guo, P. Bartlett, Shawe-Taylor, and R. Williamson. Covering numbers for support vector machines. *IEEE Transactions on Information Theory*, 48(1):239–250, 2002. *(Cited on page 185.)*

[75] R. Hamming. Error detecting and error correcting codes. *Bell Syst. Tech. J.*, 1950. *(Cited on page 35.)*

[76] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. *(Cited on page 51.)*

[77] K. Hoffman and M. Padberg. Solving airline crew scheduling problems by branch-and-cut. *Management Science*, 39(6):657–682, 1993. *(Cited on page 159.)*

[78] X. Huan and S. Mannor. Tradeoff performance and robustness in linear programming and markov decision processes. Manuscript in preparation, 2006. *(Cited on pages 25 and 125.)*

[79] G. Infanger. *Planning under Uncertainty: Solving Large-Scale Stochastic Linear Programs*. Boyd and Fraser, 1994. *(Cited on page 45.)*

[80] G. Iyengar. Robust dynamic programming. *Math. of Operations Research*, 30(2):257–280, 2005. *(Cited on page 25.)*

[81] P. Kall and S. Wallace. *Stochastic Programming*. John Wiley & Sons, 1994. *(Cited on pages 18, 45 and 104.)*

[82] J. Kallberg, R. White, and W. Ziemba. Short term financial planning under uncertainty. *Management Science*, 28:670–682, 1982. *(Cited on page 34.)*

[83] R. Kannan, L. Lovász, and M. Simonovits. Isoperimetric problems for convex bodies and a localization lemma. *Discrete Computational Geometry*, 13:541–559, 1995. *(Cited on page 19.)*

[84] J. Kemperman. The general moment problem: a geometric approach. *Annals of Mathematical Statistics*, 39:93–122, 1968. *(Cited on page 47.)*

[85] B. Korte and J. Vygen. *Combinatorial Optimization*. Springer-Verlag, 2002. *(Cited on page 80.)*

[86] A. Madansky. Bounds on the expectation of a convex function of a multivariate random variable. *Annals of Mathematical Statistics*, 30:743–746, 1959. *(Cited on page 47.)*

[87] A. Madansky. Inequalities for stochastic linear programming problems. *Management Science*, 6:197–204, 1960. *(Cited on page 47.)*

[88] W. Mak, D. Morton, and R. Wood. Monte Carlo bounding techniques for determining solution quality in stochastic programs. *Operations Research Letters*, 24:47–56, 1999. *(Cited on page 104.)*

[89] J. Matousek. *Lectures on Discrete Geometry*. Springer Verlag, Berlin, 2002. *(Cited on page 19.)*

[90] L. Miller and H. Wagner. Chance-constrained programming with joint constraints. *Operations Research*, 13:930–945, 1965. *(Cited on page 48.)*

[91] D. Morton and R. Wood. Restricted-recourse bounds for stochastic linear programming. *Operations Research*, 47:943–956, 1999. *(Cited on page 47.)*

[92] A. Nemirovski and A. Shapiro. Scenario approximations of chance constraints. *Optimization Online*, 2004. *(Cited on pages 19, 57, 106 and 133.)*

[93] A. Nemirovski and A. Shapiro. Convex approximations of chance constrained programs. *Optimization Online*, 2005. *(Cited on pages 48, 49 and 57.)*

[94] Netlib repository. Available at: `http://www.netlib.org`. *(Cited on page 34.)*

[95] A. Nilim and L. E. Ghaoui. Air traffic control under stochastic environments. In *American Control Conference*, 2004. *(Cited on pages 92, 159 and 160.)*

[96] A. Nilim and L. E. Ghaoui. Robust markov decision processes with uncertain transition matrices. Technical Report M04-26, Berkeley EECS, January 2004. *(Cited on pages 25 and 31.)*

[97] A. Nilim, L. E. Ghaoui, M. Hansen, and V. Duong. Trajectory-based air traffic management under weather uncertainty. In *Fourth USA/EUROPE ATM R & D Seminar*, 2001. *(Cited on pages 31, 159 and 160.)*

[98] E. Novak and H. Woźniakowski. Intractability results for integration and discrepancy. *Journal of Complexity*, 17(2):388–441, 2001. *(Cited on page 47.)*

[99] A. Odoni. The flow management problem in air traffic control. *Flow Control of Congested Networks*, 1987. *(Cited on pages 92 and 160.)*

[100] A. Packard. Gain scheduling via linear fractional transformations. *Systems and Control Letters*, 22(2):79–92, 1994. *(Cited on page 17.)*

[101] C. Panayiotou and C. Cassandras. A sample path approach for solving the ground-holding policy problem in air traffic control. *IEEE Transactions on Control Systems Technology*, 9(3):510–523, 2001. *(Cited on page 160.)*

[102] P. Parrilo. Semidefinite programming relaxations for semialgebraic problems. *Mathematical Programming Ser. B*, 96(2):293–320, 2002. *(Cited on page 40.)*

[103] D. Pollard. *Empirical Processes: Theory and Applications*, volume 2 of *NSF-CBMS regional conference serios in probability and statistics*. Institute of Mathematical Statistics and American Statistical Association, 1990. *(Cited on page 54.)*

[104] A. Prékopa. On probabilistic constrained programming. *Mathematical Programming Study,* 28:113–138, 1970. *(Cited on pages 48 and 105.)*

[105] A. Prékopa. Contributions to the theory of stochastic programmin. *Mathematical Programming,* 4:202–221, 1973. *(Cited on page 48.)*

[106] A. Prékopa. On logarithmic concave measures and functions. *Acta Sci. Math.,* 34:335–343, 1973. *(Cited on page 48.)*

[107] A. Prékopa. Programming under probabilistic constraint and maximizing a probability under constraints. Technical Report RRR 35-93, Rutgers University, December 1993. *(Cited on pages 48, 59 and 105.)*

[108] A. Prékopa. *Stochastic Programming.* Kluwer, 1995. *(Cited on pages 18, 45, 48, 59, 60, 61, 65 and 104.)*

[109] M. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* Wiley-Interscience, 1994. *(Cited on page 17.)*

[110] C. Revelle, E. Joeres, and W. Kirby. The linear decision rule in reservoir management and design. i. development of the stochastic model. *Water Resources Res,* 5(4):767–777, 1969. *(Cited on page 29.)*

[111] M. Robinson, J. Evans, B. Crowe, D. Klingle-Wilson, and S. Allan. Corridor integrated weather system operational benefits 2002-2003: Initial estimates of convective weather delay reduction. Technical Report Project Report ATC-313, Lincoln Laboratories, April 2005. *(Cited on pages 92, 161, 164 and 173.)*

[112] R. Rockafellar. *Convex Analysis.* Princeton University Press, 1970. *(Cited on page 37.)*

[113] R. T. Rockafellar. Optimization under uncertainty. Lecture Notes. *(Cited on page 59.)*

[114] S. Ross. *Introduction to Stochastic Dynamic Programming.* Academic Press, 1983. *(Cited on page 17.)*

[115] B. Scholkopf and A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond.* MIT Press, 2001. *(Cited on pages 54, 125 and 179.)*

[116] A. Schrijver. *Theory of Linear and Integer Programming.* Jon Wiley & Sons, New York, NY, 1986. *(Cited on page 33.)*

[117] A. Shapiro. Monte Carlo sampling methods. In A. Ruczyński and A. Shapiro, editors, *Stochastic Programming,* volume 10. Management Science, North Holland, 2003. *(Cited on page 104.)*

[118] A. Shapiro. On complexity of multistage stochastic programs. *Optimization Online,* 2005. *(Cited on page 59.)*

[119] A. Shapiro. On complexity of multistage stochastic programs. Technical Report available from: `http://www.optimization-online.org`, Georgia Tech, MONTH 2005. *(Cited on pages 104, 110 and 125.)*

[120] A. Shapiro and A. Nemirovski. On complexity of stochastic programming problems. Technical Report available from: `http://www.optimization-online.org`, Georgia Tech, 2005. *(Cited on pages 59, 60 and 110.)*

[121] J. Shawe-Taylor, P. Bartlett, R. Williamson, and M. Anthony. Structural risk minimization over data-dependent hierarchies. *IEEE Transactions on Information Theory,* 44(5):1926–1940, 1998. *(Cited on page 180.)*

[122] H. Sherali and A. Alameddine. A new reformulation-linearization technique for bilinear programming problems. *Journal of Global Optimization,* 2:379–410, 1992. *(Cited on page 75.)*

[123] H. Sherali and C. Tuncbilek. A reformulation-convexification approach for solving nonconvex quadratic programming problems. *Journal of Global Optimization,* 7:1–31, 1995. *(Cited on page 75.)*

[124] R. V. Slyke and R. Wets. *l*-Shaped linear programs with applications to optimal control and stochastic linear programs. *SIAM J. Appl. Math.,* 17:638–663, 1969. *(Cited on page 49.)*

[125] A. Soyster. Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research,* 21:1154–1157, 1973. *(Cited on pages 18, 35 and 38.)*

[126] C. Swamy and D. Shmoys. Sampling-based approximation algorithms for multistage stochastic optimization. In *Proc. 46th FOCS,* pages 357–366, 2005. *(Cited on page 104.)*

[127] A. Takeda, S. Taguchi, and R. Tütüncü. Adjustable robust optimization models for nonlinear multi-period optimization. *Submitted,* 2004. *(Cited on pages 61 and 62.)*

[128] The Mathworks. Matlab. See `http://www.mathworks.com`. *(Cited on page 183.)*

[129] C. Tomlin, J. Lygeros, and S. Sastry. A game theoretic approach to controller design for hybrid systems. *Proceedings of the IEEE*, 88(7):949–970, 2000. *(Cited on page 159.)*

[130] C. Tomlin, G. Pappas, and S. Sastry. Conflict resolution for air traffic management: a study in multiagent hybrid systems. *IEEE Transactions on Automatic Control*, 43(4):509–521, 1998. *(Cited on page 159.)*

[131] F. Torrisi and M. Baotic. CPLEXINT. Available at: `http://control.ee.ethz.ch/ hybrid/cplexint.php`. *(Cited on page 183.)*

[132] J. Traub and A. Werschulz. *Complexity and Information*. Cambridge University Press, 1998. *(Cited on pages 19 and 47.)*

[133] L. Valiant. A theory of the learnable. *Comm. ACM*, 27(11):1134–1142, 1984. *(Cited on page 55.)*

[134] S. van de Geer. *Empirical Processes in M-Estimation*. Cambridge University Press, 2000. *(Cited on page 54.)*

[135] A. van der Vaart. *Asymptotic Statistics*. Cambridge University Press, 2000. *(Cited on page 54.)*

[136] A. van der Vaart and J. Wellner. *Weak Convergence and Empirical Processes*. Springer-Verlag, 1996. *(Cited on pages 54, 130 and 185.)*

[137] P. Vance, C. Barnhart, E. Johnson, and G. Nemhauser. Airline crew scheduling: A new formulation and decomposition algorithm. *Operations Research*, 45(2):188–200, 1997. *(Cited on page 159.)*

[138] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer, 1999. *(Cited on page 180.)*

[139] M. Vidyasagar. Randomized algorithms for robust controller synthesis using statistical learning theory. *Automatica*, 37:1515–1528, 2001. *(Cited on page 50.)*

[140] P. Vranas, D. Bertsimas, and A. Odoni. Dynamic ground-holding policies for a network of airports. *Transportation Science*, 1994. *(Cited on page 160.)*

[141] P. Vranas, D. Bertsimas, and A. Odoni. The multi-airport ground-holding problem in air traffic control. *Operations Research*, pages 249–261, 1994. *(Cited on page 160.)*

[142] G. Watson. Robust solutions to a general class of approximation problems. *SIAM Journal on Scientific Computing*, 25(4):1448–1460, 2004. *(Cited on page 35.)*

[143] Y. Zhang. A general robust-optimization formulation for nonlinear programming. Technical Report TR04-13, Rice University CAAM, June 2005. *(Cited on page 35.)*

[144] K. Zhou, J. Doyle, and K. Glover. *Robust and Optimal Control.* Prentice-Hall, Upper Saddle River, NJ, 1996. *(Cited on page 35.)*