Models and Computational Strategies for Multistage Stochastic Programming under Endogenous and Exogenous Uncertainties

Submitted in partial fulfillment of the requirements for
the degree of
Doctor of Philosophy
in
Chemical Engineering

Robert M. Apap

B.Sc., Chemical Engineering, University of Connecticut

Carnegie Mellon University
Pittsburgh, PA 15213

July 2017

Acknowledgments

In what I assumed was a clerical error, I was accepted to the Department of Chemical Engineering at Carnegie Mellon University in 2012. I had applied to the program at almost the last possible minute at the urging of a good friend. She was in a Ph.D. program at Dartmouth College, and I was essentially wasting away my undergraduate education by working as a controls engineer. While rethinking my earlier opposition to academia, my friend (neglecting to mention how much she hated graduate school) further urged me to accept the offer from CMU.

To her credit, this has turned out to be a good decision. I was lucky enough to end up in the research group of Ignacio Grossmann, who is not only a brilliant researcher but a great human being (a seriously underrated quality in academia). I am extremely grateful for the opportunity to have had him as a mentor for these last few years. I also have to acknowledge his patience, which has somehow tolerated my habit of taking a short eternity to perfect everything that we've worked on together.

One aspect of this thesis that may seem a bit unusual, and I believe deserves some further explanation, is that I have an English professor on my committee. There is an interesting story behind this choice.

While selecting members of the committee before my thesis proposal, I was having trouble finding a Carnegie Mellon professor outside the Department of Chemical Engineering. This was one of the requirements laid out in the graduate-student handbook. Professors outside the university could serve as "external members," but could not fulfill this requirement, even if they were the expert of all experts in the respective research area. I had a more-than-competent expert outside the university (Laureano Escudero) and didn't see any purpose in stacking the committee with yet another CMU mathematician or engineer for the sake of fulfilling a somewhat silly rule. After all, a thesis would seem to be *more* properly vetted if it were approved by an expert from an outside institution. Restricting this committee member to be outside the Department of Chemical Engineering but still affiliated with Carnegie Mellon seemed to be completely arbitrary and extremely short-sighted.

I discussed these concerns with Ignacio. He agreed that this was a special case given the subject of my thesis, and he encouraged me to contact the department administrator (Cindy). Ignacio did not have high hopes, however. He predicted that Cindy would forward the request to the director of graduate education (let's call him Paul), and that Paul would say no.

Cindy replied (more or less): "I think that you should check with Paul on this one." Despite the fact that I had specifically quoted the grad-student handbook in my email and asked if the requirement could be waived, Paul informed me that the handbook was quite clear about this. He further let me know that this rule was fundamental (he didn't tell me why) and that there would have to be a precedent to waive the rule (he was aware of no such precedent). What I gathered from his message was that I had to follow the rule because it was written in the handbook. He was the author of the handbook, so this was non-negotiable. I took another look at the requirements and vowed to put an art professor on my thesis committee.

The more I thought about this, the more I realized that there was actually quite a lot of value in selecting a committee member from a tangentially-related research area. Maybe not the art department, but there was the beginning of an idea here. Engineers tend to live in their own world and often fail miserably at explaining their ideas to non-experts. (There are some who even fail miserably at explaining their ideas to other experts.) After looking around for a little while, I found Necia Werner, a CMU English professor who specializes in engineering communications. I pitched the idea of "stochastic programming for non-experts" and she was intrigued by the concept. I am grateful to Necia for agreeing to the unorthodox idea of joining a chemical engineer's thesis committee, since her decision motivated me to prepare the publication that appears in Chapter 7 of this thesis. As it turns out, the silly handbook rule inspired some rather unique interdisciplinary work.

I would like to further thank my other thesis committee members, Nikolaos Sahinidis, Chrysanthos Gounaris, Bora Tarhan, and Laureano Escudero, for providing invaluable insights to my work.

Additionally, as a special thank you for making all of this possible, I gratefully acknowledge financial support from the John E. Swearingen Graduate Fellowship and the Center for Advanced Process Decision-making at Carnegie Mellon University. I also acknowledge partial support from the ExxonMobil Upstream Research Company.

The template for this thesis was prepared by Bruno Calfa, and I offer my thanks to him as well.

Finally, I cannot forget the countless other students in the department who have had an unbelievable influence on me (and my work) over the course of the last few years. To these students, my friends, and family: I will thank you in person, since I have already rambled for almost two pages and this document is due in 20 minutes.

Abstract

This dissertation addresses the modeling and solution of mixed-integer linear multistage stochastic programming problems involving both endogenous and exogenous uncertain parameters. We propose a composite scenario tree that captures both types of uncertainty, and we exploit its unique structure to derive new theoretical properties that can drastically reduce the number of non-anticipativity constraints (NACs). Since the reduced model is often still intractable, we discuss two special solution approaches. The first is a sequential scenario decomposition heuristic in which we sequentially solve endogenous MILP subproblems to determine the binary investment decisions, fix these decisions to satisfy the first-period and exogenous NACs, and then solve the resulting model to obtain a feasible solution. The second approach is Lagrangean decomposition. We present numerical results for a process network planning problem and an oilfield development planning problem. The results clearly demonstrate the efficiency of the special solution methods over solving the reduced model directly. To further generalize this work, we also propose a graph-theory algorithm for non-anticipativity constraint reduction in problems with arbitrary scenario sets. Finally, in a break from the rest of the thesis, we present the basics of stochastic programming for non-expert users.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

In the optimization of process systems, there is often some level of uncertainty in one or more of the input parameters. A major challenge for the decision-maker, then, is to determine how to best account for this uncertainty. Rather than optimizing for expected values, which can often lead to suboptimal or even infeasible solutions, these problems can usually be effectively approached with mathematical programming techniques such as stochastic programming (Birge and Louveaux, 2011), robust optimization (Ben-Tal et al., 2009), or chance-constrained optimization (Li et al., 2008).

In stochastic programming, the topic of this thesis, a decision-maker must implement a set of decisions at the beginning of the planning horizon without knowing exactly what the true values of some of the input parameters will be. After the uncertainty in those parameters is resolved, the decision-maker can take corrective action based on this new information. Since this approach does not fix all of the decisions at the beginning of the planning horizon, it tends to be an appropriate choice for long-term planning projects that may span several decades (Grossmann et al., 2016).

In robust optimization, on the other hand, the general goal is to guarantee feasibility over a specified uncertainty set. This is typically more appropriate for short-term scheduling problems where feasibility is a major concern and where there is little scope for corrective action (Grossmann et al., 2016). Chance-constrained optimization also has a similar emphasis on constraint feasibility; specifically, some of the constraints must be satisfied with at least a given level of probability for all possible outcomes of the uncertain parameters present in those respective constraints (Calfa et al., 2015). As our intended applications are long-term planning problems in which corrective action is essential and probabilistic constraints are not required, we focus on a stochastic programming framework to effectively hedge against parameter uncertainties. Extensions of robust optimization and chance-constrained optimization that allow for some corrective action will not be considered here; however, a discussion of these approaches can be found in Ben-Tal et al. (2004) (as well as Lappas and Gounaris, 2016; Zhang et al., 2016) and Liu et al. (2016), respectively.

A second major concern for the decision-maker is the *type* of uncertainty. In general, there are two types: *exogenous*, where the true parameter values are revealed independently of decisions, and *endogenous*, where the parameter realizations are influenced by the decisions (Jonsbråten, 1998). In the context of process systems engineering, exogenous uncertainties often correspond to market uncertainties, such as crude-oil prices. The corresponding realizations occur automatically in each period of the planning horizon, independently of any decisions. For example, in an oilfield planning problem, we may rely on a forecast to predict the price of oil in the upcoming year. At the time the forecast is prepared, the true price is unknown. Once next year arrives, however, we will realize the true price of oil, regardless of the decisions that have been made.

For endogenous uncertainties, we may be dealing with at least two distinct types which we will refer to as Type 1 and Type 2 (Goel and Grossmann, 2006). In the case of Type 1 endogenous uncertainties, decisions influence the parameter realizations by altering the underlying probability distributions for the uncertain parameters. A simple example of this may be an oil company's decision to flood the market in order to force a competitor out of business. Here the uncertainty is no longer strictly exogenous, as the decision will make lower oil-price realizations more probable. This type has been considered in relatively few stochastic-programming publications; namely, as far as we are aware: Ahmed (2000); Viswanath et al. (2004); Flach (2010); Peeta et al. (2010); Tong et al. (2012) and Escudero et al. (2013a) (which consider both exogenous and Type 1 endogenous uncertainties); Laumanns et al. (2014); Hellemo (2016); and Escudero et al. (2016a) (which also considers both exogenous and Type 1 endogenous uncertainties).

In the case of Type 2 endogenous uncertainties, decisions influence the parameter realizations by affecting the time at which we observe these realizations. This refers specifically to technical parameters, such as oilfield size, for which the true values cannot be determined until a particular investment decision is made (Goel and Grossmann, 2006). For instance, seismic studies may provide a good indication of the size of an oilfield, but we will not know the exact recoverable oil volume until we drill the field and begin producing from it (Goel and Grossmann, 2004). Note that Type 1 and Type 2 endogenous uncertainties are not mutually exclusive; for example, the choice of drilling technology may make higher oil recoveries more likely (Type 1), but the true recovery will only be revealed if we decide to develop that field (Type 2). This case is referred to as Type 3 endogenous uncertainty in Hellemo (2016).

It is worth noting that Powell (2011) classifies problems with either Type 1 or Type 2 endogenous uncertainty as "state-dependent information processes" and recommends the use of approximate dynamic programming (ADP) to solve them. In fact, dynamic programming methods have been successfully applied to optimization problems involving exogenous uncertainties (e.g., Powell, 2011), Type 1 endogenous uncertainties (e.g., Webster et al., 2012), and Type 2 endogenous uncertainties (e.g., Choi et al., 2004). Such methods are outside the scope of this thesis; however, we refer the reader to these selected references for further details.

For the endogenous uncertainties considered here, we will focus exclusively on Type 2, where decisions affect the timing of realizations. This is sometimes referred to as 'exogenous uncertainty with endogenous observation,' in view of the fact that the technical uncertainty itself is exogenous (as we cannot alter it), but the time at which this uncertainty is resolved is endogenous (since it depends on our investment decisions) (Colvin and Maravelias, 2011; Mercier and Van Hentenryck, 2011). For the remainder of this thesis, we will drop the "Type 2" prefix and simply refer to these uncertainties as *endogenous*.

The literature on stochastic programming (SP) has focused primarily on problems with exogenous uncertainties. Reviews of this area are given in Birge (1997), Schultz (2003), and Sahinidis (2004). Endogenous uncertainty is a newer area and has received far less attention in the literature, with the first publication introduced by Jonsbråten et al. (1998) less than 20 years ago.[1]

In the area of process systems engineering, Goel and Grossmann (2004) and Goel et al. (2006) addressed a gas-field development problem in which the size and initial deliverability of reserves are uncertain, and these endogenous uncertainties are resolved immediately after the drilling decisions are made. Tarhan and Grossmann (2008) explored the synthesis of process networks with endogenous uncertainty in the process yields and relaxed the common assumption of immediate resolution of the uncertainty. Instead, the authors modeled the gradual resolution of uncertainty over time, which is more in line with reality in some applications. Tarhan et al. (2009) applied this approach to the oil/gas-field development problem and considered nonlinearities in the reservoir model. Boland et al. (2008) studied the open pit mine production scheduling problem with endogenous uncertainty in the geological properties of the mined materials. The authors proposed a lazy-constraints approach for handling the large number of non-anticipativity constraints, whereby these constraints are only added to the problem as needed.

Colvin and Maravelias (2010) (an extension of Colvin and Maravelias, 2008, 2009 considered endogenous uncertainty in the scheduling of pharmaceutical clinical trials, and proposed a branch-and-cut method for this problem, as well as several theoretical reduction properties. Although many of these reduction properties are specific to the pharmaceutical scheduling problem, one applies to the general case considered here and will be discussed later in this thesis. Solak et al. (2010) studied R&D project portfolio management under endogenous uncertainty, where the investment requirement for each project resolves gradually as a function of the progress of the respective project. The authors solved the resulting model with the sample average approximation method. The sample problems in this method were solved through the use of Lagrangean relaxation and a heuristic. In a related study, Colvin and Maravelias (2011) explored endogenous uncertainty in R&D activities in an R&D pipeline management problem, and also explored risk management strategies in this context.

Gupta and Grossmann (2011) discussed process networks with endogenous uncertainty in process yields, and proposed a general theoretical property that can considerably reduce the dimensionality of the model when there are uncertain parameters defined with three or more possible realizations. Gupta and Grossmann (2014a) developed a scenario grouping Lagrangean decomposition algorithm for solving large-scale problems of this class (which is similar in concept to the scenario clustering approach of Escudero et al., 2013b, for two-stage exogenous problems). Gupta and Grossmann (2014b) also made advances in the modeling of the oilfield development planning problem under endogenous uncertainty. More recently, Christian and Cremaschi (2015) proposed two heuristic solution methods for the R&D pipeline management problem: a shrinking-horizon, multiple two-stage stochastic programming decomposition algorithm, and a knapsack decomposition algorithm. The authors extended the knapsack decomposition algorithm in Christian and Cremaschi (2017). Additionally, Boland et al. (2016) and Hooshmand Khaligh and MirHassani

[1] Jonsbråten et al. (1998) is the first work to address the specific case considered here, where decisions must be made in order to gain more accurate process information. Pflug (1990) is the first work (of which we are aware) to consider the case of a decision-dependent stochastic process.

(2016b) explored non-anticipativity constraint reduction for multistage stochastic programs with arbitrary scenario sets, which will be of particular interest later in this thesis. Other publications on stochastic programming under endogenous uncertainty which we will not discuss here, but may be of interest to the reader, include: multistage stochastic network interdiction (Held and Woodruff, 2005); the decision-rule approach to multistage stochastic programming (Vayanos et al., 2011); the optimal design of integrated chemical-production sites (Terrazas-Moreno et al., 2012); computational strategies for nonconvex, multistage mixed-integer nonlinear programs (Tarhan et al., 2013); and the dynamic single-vehicle routing problem with uncertain demands (Hooshmand Khaligh and MirHassani, 2016a).

Although many problems contain both endogenous and exogenous uncertainties (e.g., uncertain field sizes *and* uncertain oil prices), optimization under both types has been largely unexplored in the literature. To the best of our knowledge, Goel and Grossmann (2006) has been the only previous work to comprehensively explore multistage stochastic programming (MSSP) problems of this class.[2] The authors introduced a hybrid mixed-integer linear disjunctive programming model for these problems and proposed two efficient theoretical properties for eliminating redundant constraints; however, their numerical studies considered only endogenous uncertainties in capacity expansion and sizing problems. Dupačová (2006) briefly discussed optimization under both types of uncertainty but did not provide a specific multistage formulation, new solution strategies, or numerical results. More recently, Bruni et al. (2015) proposed a stochastic programming approach for the operating theater scheduling problem, in which there is exogenous uncertainty in the arrival of emergency patients and endogenous uncertainty in the duration of surgery. The authors offered only brief details on the modeling of the endogenous uncertainty and employed a heuristic approach to solve the problem. As our focus is on a general framework for multistage stochastic programming, Goel and Grossmann (2006) will serve as the foundation for this thesis.

The primary goals of this work are to: (1) efficiently model multistage stochastic programming problems that involve both endogenous and exogenous parameters; (2) develop effective solution methods for these problems; and (3) apply the proposed methods to challenging applications. Given the complexity of these problems and the fact that only little work has been reported on them, we begin in the next section with a detailed review of the relevant background regarding multistage stochastic programming under exogenous uncertainty, as well as multistage stochastic programming under endogenous uncertainty. In Chapter 2, we then introduce the definitions and notation necessary to model these types of uncertainties and propose a composite scenario tree that captures all possible realizations of both endogenous and exogenous parameters. Next, in Chapter 3, we present the multistage stochastic programming models for purely exogenous uncertainty, purely endogenous uncertainty, and both endogenous and exogenous uncertainties. After this point, we focus our attention on the latter case, and in Chapter 4, we discuss reduction properties that can significantly reduce the dimensionality of these problems. In Chapter 5, we introduce a sequential scenario decomposition heuristic and briefly review Lagrangean decomposition, and then apply these algorithms to solve a process network example and an oilfield development planning problem.

[2] It is worth noting that there is also a significant lack of literature on multistage stochastic programs with both exogenous and Type 1 endogenous uncertainties. For a discussion of modeling and solution considerations for this class of problems, see Escudero et al. (2013a).

We propose a graph-theory algorithm for non-anticipativity constraint reduction in problems with arbitrary scenario sets in Chapter 6 in order to further generalize this work. In Chapter 7, in a break from the rest of the thesis, we present the basics of stochastic programming for non-expert users. Finally, in Chapter 8, we critique the work presented in this thesis, summarize the primary contributions, and propose possible directions for future research.

1.1 Background

1.1.1 Stochastic Programming under Exogenous Uncertainty

A common approach for optimization under exogenous uncertainty is two-stage stochastic programming (Birge and Louveaux, 2011). In this approach, first-stage decisions are made 'here and now' at the beginning of the first time period, without knowing exactly how the uncertainty will unfold. The decision-maker then waits for the outcome. At some point following these decisions, the uncertainty is resolved and the true values of the exogenous-uncertain parameters become known. Second-stage, or recourse ('wait-and-see'), decisions are then taken by the decision-maker as corrective action. For example, in a problem spanning multiple time periods, the decision-maker's first-stage decisions may enforce an investment plan that is fixed for the entire horizon. Subsequent recourse decisions allow operating conditions to be specified in response to this plan, based on the realizations observed for the exogenous-uncertain parameters (see, for instance, Liu and Sahinidis, 1996).

In practice, however, it is often necessary for the decision-maker to have the additional freedom to make new here-and-now decisions at the beginning of each time period. This leads to a multistage stochastic programming formulation; decisions, realizations, and recourse actions occur sequentially, allowing for a more accurate description of the decision-making process for long-term planning projects. This is illustrated in Figure 1.1 for a three-stage problem with one exogenous-uncertain parameter, ξ_t. We use y_t^s and x_t^s to denote the vectors of here-and-now decisions and recourse decisions, respectively, in each time period t and scenario s. Note that $t = 0$ corresponds to the beginning of the first time period (stage 1), $t = 1$ corresponds to the end of the first time period/beginning of the second time period (stage 2), and $t = 2$ corresponds to the end of the second time period (stage 3). As will be discussed, multistage stochastic programming also provides a more suitable framework for endogenous uncertainties, as these realizations can occur at any point in the time horizon.

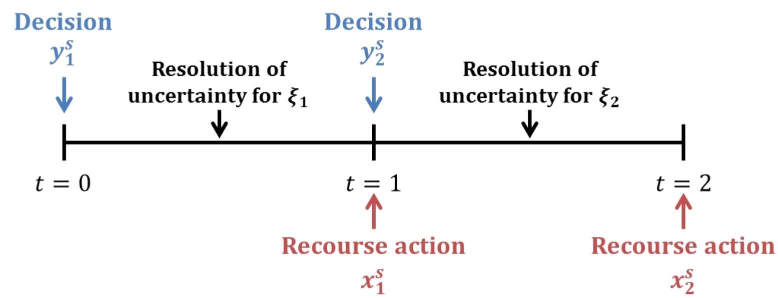


Figure 1.1: Sequence of events in multistage stochastic programming under exogenous uncertainty.

One fundamental assumption in stochastic programming, which we have already made here, is that the time horizon is represented by a set of discrete time periods. A second very common assumption is that the possible realizations (possible values) for each uncertain parameter are available from a discretized probability distribution. With these two assumptions in place, the stochastic process can be represented by a scenario tree, like that shown in Figure 1.2a. Note that this is the scenario-tree representation of Figure 1.1, where the exogenous parameter ξ_t has two possible realizations (*low* (L) or *high* (H)) in each time period. Each node in the tree represents a different possible state of the system in time period t. Arcs indicate a possible transition from a state in time period t to a new state in time period $t+1$, with a given probability of this transition occurring. For example, the system shown in Figure 1.2a can transition from its initial state in time period 1 to either of two different states in time period 2 depending upon the realized value of ξ_1. A complete path from the root node to a leaf node represents a scenario, which corresponds to one possible combination of realizations for the uncertain parameters (e.g., $(\hat{\xi}_1^L, \hat{\xi}_2^L)$). Note that since the uncertainty is purely exogenous in this case, and exogenous realizations occur automatically in each time period, the structure of the scenario tree is known in advance.

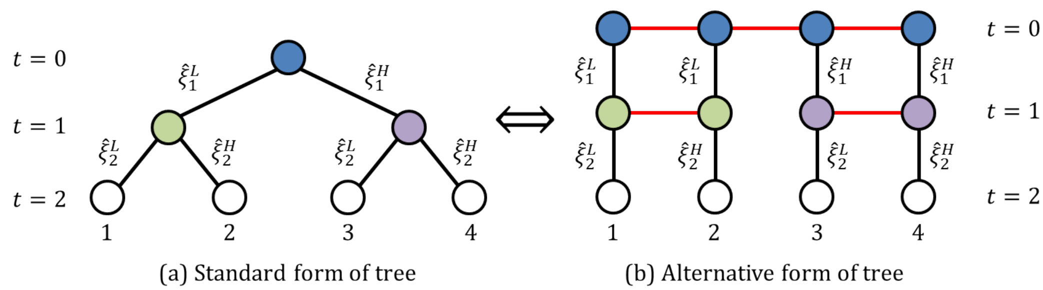


Figure 1.2: An exogenous scenario tree and its alternative representation.

One complicating aspect of the standard form of the scenario tree (Figure 1.2a) is that the corresponding stochastic programming problem contains variables that are shared among two or more scenarios. For instance, in Figure 1.2a, all four scenarios share the variables of the root node (shown in blue), scenarios 1 and 2 share the variables of the green node at $t = 1$, and scenarios 3 and 4 share the variables of the purple node at $t = 1$. This prevents the direct application of scenario-decomposition approaches like Lagrangean decomposition which can be effective for solving large stochastic programs.

Ruszczyński (1997) proposed an alternative form of the scenario tree in which shared nodes are split such that each scenario is given its own unique set of nodes. This is shown in Figure 1.2b. The alternative form is more amenable to scenario decomposition, as variables are no longer shared and each scenario represents a different instance of the same deterministic problem with different realizations for the uncertain parameters. Notice, however, that in moving from the standard form of the tree to the alternative form, we have created several copies of the same states. For example, the root node in Figure 1.2a has been split into four separate nodes in Figure 1.2b. These four nodes all have identical information at that point in time. Accordingly, scenarios 1–4 are said to be *indistinguishable* at the beginning of the first time period. It follows that because these scenarios are indistinguishable at that time, we must treat them all in the same way, and we must make the same here-and-now decisions in all four scenarios at the beginning of the first time period. This equality between the states is enforced by the red horizontal lines connecting the nodes in Figure 1.2b. These red lines represent what are known as *non-anticipativity constraints* (Rockafellar and Wets, 1991; Ruszczyński, 1997). Without these constraints, it is clear that the tree would decompose into independent scenarios in which we would be anticipating *one* particular outcome for each of the uncertain parameters. Since we do not have this level of information, these constraints are required. Using the notation from Figure 1.1, we express the non-anticipativity constraints (NACs) as $y_1^1 = y_1^2$, $y_1^2 = y_1^3$, and $y_1^3 = y_1^4$. Similarly, as can be seen from the green nodes at $t = 1$ in Figure 1.2b, we must make the same recourse decisions at the end of the first time period and the same here-and-now decisions at the beginning of the second time period in scenarios 1 and 2. Thus, the corresponding NACs are $x_1^1 = x_1^2$, and $y_2^1 = y_2^2$. A similar argument can be made regarding the purple nodes at $t = 1$ and the corresponding decisions in scenarios 3 and 4. Notice that by the end of the time horizon, all scenarios differ in the realizations of exogenous parameter ξ_t, and the leaf nodes refer to independent states. Accordingly, the scenarios are said to be *distinguishable* at that time, and non-anticipativity no longer applies (as noted by the absence of any red lines connecting the scenarios). In other words, at the end of the second time period, we are free to make independent recourse decisions in each of the four scenarios.

It is important to note that the alternative form of the scenario tree corresponds directly to the non-anticipativity formulation of the *deterministic equivalent* for stochastic programming problems (Birge and Louveaux, 2011). In this formulation, as the preceding discussion suggests, each scenario represents a different instance of the deterministic problem with different realizations for the uncertain parameters, and non-anticipativity constraints ensure that we make the same decisions in indistinguishable scenarios in each time period. This is the modeling approach that will be used in this thesis. We will rely heavily on the concept that two scenarios are indistinguishable in time period t if they are identical in the realizations of all uncertain parameters that have been resolved up until that time;[3] and as soon as the scenarios differ in the realization of any uncertain parameter, they are distinguishable for the remainder of the time horizon. As we will describe in the next section, the alternative form of the scenario tree is also very useful in modeling endogenous uncertainties.

1.1.2 Stochastic Programming under Endogenous Uncertainty

In the case of stochastic programming under endogenous uncertainty, a multistage framework is generally the logical starting point. This can be seen when considering a problem such as the capacity expansion of process networks (Goel and Grossmann, 2006), where small installations are made in early time periods to determine the true yields of new process units. Capacity expansions can then be made at a later point in time to capitalize on that knowledge. This type of decision making is not possible with only two stages. Furthermore, in the two-stage case, if investments are not made at the beginning of the first time period (as this may not be optimal), the uncertainty in the endogenous parameters cannot be resolved during the time horizon.

The decision-making process in these types of multistage stochastic programming problems

[3] The phrase "indistinguishable in time period t" will be used as a shorthand way of stating: "indistinguishable at the end of time period t, after all realizations in that period have occurred."

proceeds in a manner similar to that of the exogenous case (Figure 1.1). The primary difference here is that the timing of realizations depends on the decisions. Hence, uncertainty is not resolved automatically in each time period, and the uncertainty in some parameters may not be resolved at all. This is illustrated in Figure 1.3 for a three-stage problem with two endogenous-uncertain parameters, θ_1 and θ_2, where set $\overline{\mathcal{I}}_t^s$ indicates the parameters that are realized in each time period t of scenario s. It is important to note that rather than being associated with a particular time period, endogenous parameters represent intrinsic properties of a given *source*, such as the size of an *oilfield* or the yield of a *process unit* (Goel and Grossmann, 2006). Accordingly, in the case of Figure 1.3, we state that θ_1 is an endogenous parameter associated with a given "Source 1," and θ_2 is an endogenous parameter associated with a given "Source 2."

Consider the case where we make an investment[4] in both Source 1 and Source 2 at the beginning of the first time period. Also, assume that the uncertainty is resolved immediately after we implement this decision. As indicated by the sequence of events in Figure 1.3, we will realize the values of θ_1 and θ_2 in the first time period, and no realizations will occur in the second time period (i.e., $\overline{\mathcal{I}}_1^s = \{1, 2\}$, and $\overline{\mathcal{I}}_2^s = \emptyset$). Notice that unlike the exogenous case, we do not know which parameters will be realized until we know which decisions we will make. This information is not known in advance and must be determined by solving the corresponding stochastic programming problem. We use dotted lines in Figure 1.3 to indicate that the timing of the realizations is conditional.

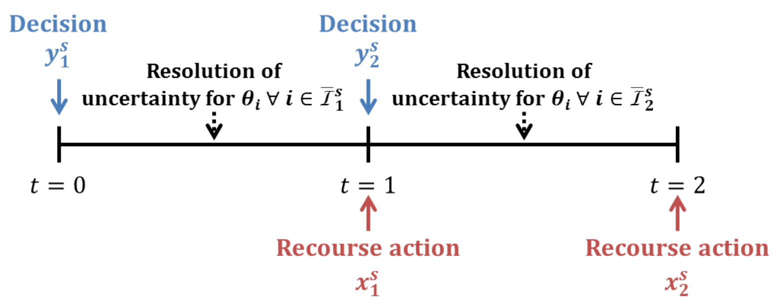


Figure 1.3: Sequence of events in multistage stochastic programming under endogenous uncertainty.

As this discussion suggests, the scenario-tree representation of these stochastic processes is also not as straightforward as the exogenous case. This is for the simple reason that there are many possible outcomes for the decisions, and accordingly, there will be many possible outcomes for the structure of the scenario tree. This is illustrated in Figure 1.4 with just a few of the many possible scenario-tree representations of Figure 1.3, where the endogenous parameters θ_1 and θ_2 each have two possible realizations (*low* (L) or *high* (H)). (Note that above each scenario in the alternative form of the tree, we indicate the possible realizations defined for that particular scenario.) We again assume that the uncertainty in a parameter is resolved immediately after an investment is made in its respective source. In the first case, Figure 1.4a, an investment is made in Source 1 at the beginning of the first time period. As a result, the value of θ_1 is immediately realized in all scenarios. Notice that non-anticipativity constraints still apply for the beginning of the first time

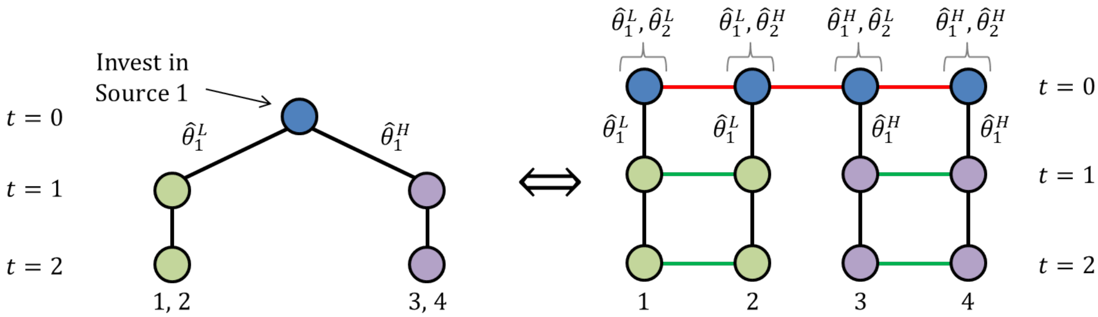
[4] An 'investment in a source' broadly refers to any here-and-now decision that allows us to realize the values of the endogenous parameters associated with that source.

period in the alternative form of the tree, just as they do in the exogenous case (i.e., we have the same red lines at $t = 0$ as we do in Figure 1.2b). This is because at the beginning of the time horizon (prior to the implementation of the decisions for the first time period), we have yet to make any decisions, and no realizations have occurred. Thus, all scenarios must be indistinguishable at that time, regardless of the type of uncertainty being considered.

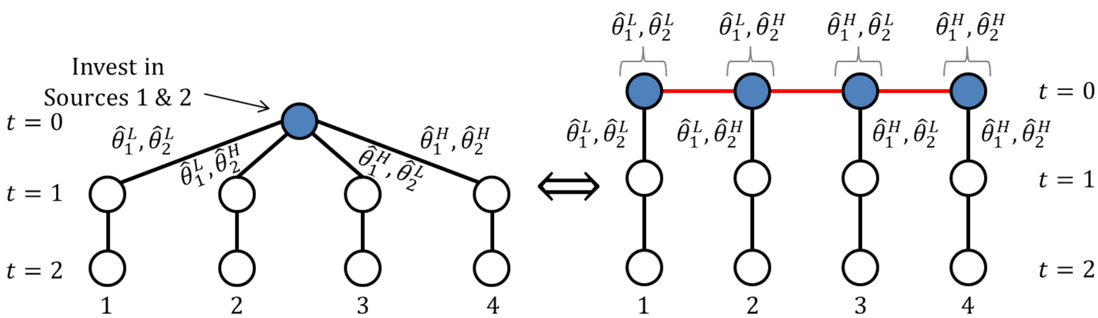
Continuing with the discussion of Figure 1.4a, we note that no other investments are made after the first-stage decisions, so the value of θ_2 is never realized. Non-anticipativity constraints (shown in green) therefore restrict our decision-making such that, for the remainder of the time horizon, we must make all of the same decisions in scenarios 1 and 2, as well as all of the same decisions in scenarios 3 and 4. In the second case, Figure 1.4b, an investment is made in both Source 1 and Source 2 at the beginning of the first time period (this is the case that was previously described in relation to Figure 1.3). The values of θ_1 and θ_2 are immediately realized in all scenarios, and the four scenarios are distinguishable for the remainder of the time horizon. In other words, by the end of the first time period, we are free to make independent decisions in all scenarios. In Figure 1.4c and Figure 1.4d, an asymmetric scenario tree results from making an investment in only two of the four scenarios. By simply swapping the order of investments, the alternative tree in Figure 1.4d looks very different from that of Figure 1.4c, in the sense that non-anticipativity constraints no longer apply solely between adjacent scenarios. We again emphasize that many other outcomes for the tree are possible, even with only four scenarios.

Due to the conditional structure of the endogenous scenario tree, it is clearly impractical to model all possible outcomes with the standard form of the tree. To deal with this issue, we adopt the alternative form and create a superstructure in which non-anticipativity constraints are applied conditionally (as inspired by Gupta and Grossmann, 2014a). This is shown in Figure 1.5, where the dotted green lines represent these conditional NACs. Notice that the superstructure form of the tree accounts for all possible outcomes, and any of the alternative trees shown in Figure 1.4 can easily be recovered from Figure 1.5.

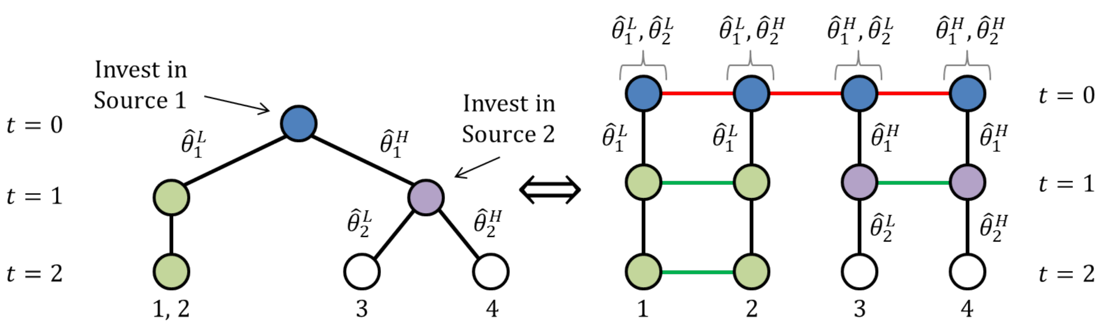
Because we are now dealing with conditional NACs, the modeling approach is significantly different from the simple equality constraints for exogenous uncertainty. In the exogenous case, if two scenarios differ in the realization of uncertain parameter ξ_t in time period t_X^*, the scenarios will be distinguishable by the end of that time period (since realizations occur automatically). Therefore, we apply non-anticipativity constraints between these scenarios in all time periods up to, but not including, the end of t_X^*. In the endogenous case, however, it is not this simple. Scenarios that differ in the possible realization of an uncertain parameter θ_i will remain indistinguishable until the uncertainty in that parameter is resolved; up until that point, t_N^*, the scenarios are identical. Because we do not know the value of t_N^* for these scenarios, we must conditionally apply NACs for all decisions in all time periods (excluding the decisions made at the beginning of the first time period and in other initial time periods, as well). The indistinguishability is determined at each point in time as part of the stochastic programming problem, and the NACs are enforced if the scenarios are indistinguishable and ignored if they are not. As opposed to a fixed scenario tree in the exogenous case, the optimal structure of the endogenous scenario tree is determined by solving this stochastic program. The modeling of NACs will be discussed in greater detail later in this



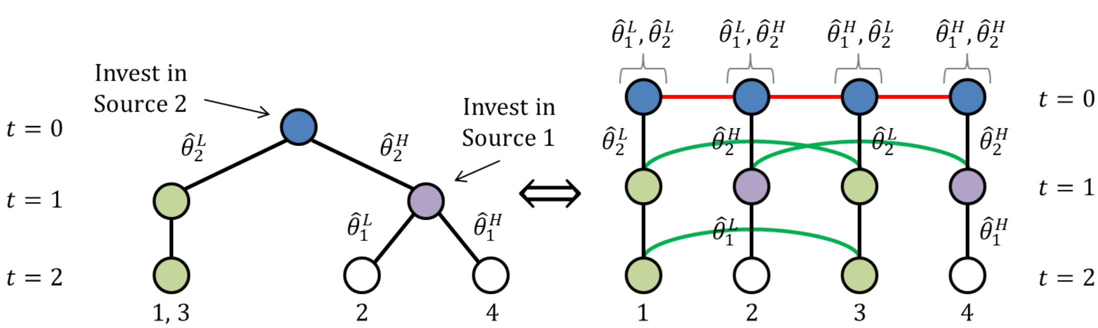
(a) θ_1 is realized during the first time period in all scenarios.



(b) θ_1 and θ_2 are both realized during the first time period in all scenarios.



(c) θ_1 is realized during the first time period in all scenarios. θ_2 is realized during the second time period in scenarios 3 and 4 only.



(d) θ_2 is realized during the first time period in all scenarios. θ_1 is realized during the second time period in scenarios 2 and 4 only.

Figure 1.4: Four possible structures for a scenario tree with two endogenous parameters.

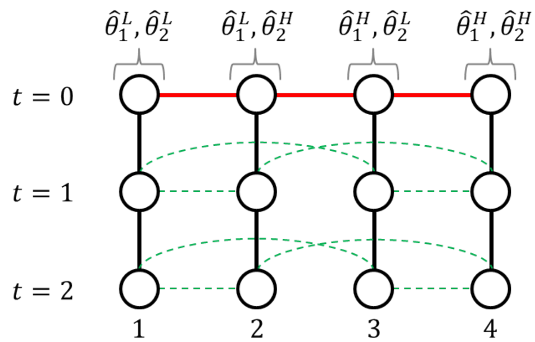


Figure 1.5: A superstructure representation for endogenous scenario trees.

thesis.

Chapter 2

Definitions and Notation

2.1 Mathematical Description of Exogenous Uncertainty

Let the time horizon be divided into a set of discrete time periods $\mathcal{T} := \{t : t = 1, 2, \ldots, T\}$, and let set $\mathcal{J} := \{j : j = 1, 2, \ldots, J\}$ define the index of each exogenous-uncertain parameter. We define $\xi_{j,t}$ as exogenous parameter $j \in \mathcal{J}$ in time period $t \in \mathcal{T}$. The exogenous parameter has a number of possible realizations given by the ordered set $\Xi_{j,t} := \{\hat{\xi}^r_{j,t} : r = 1, 2, \ldots, R_{j,t}\}$, where r refers to the index of one particular realization, and for convenience, we set $\hat{\xi}^1_{j,t} < \hat{\xi}^2_{j,t} < \cdots < \hat{\xi}^{R_{j,t}}_{j,t}$. As an example of how we use this notation, if $r = 2$ is the index of the actual realization for parameter j in time period t, we will have $\xi_{j,t} = \hat{\xi}^2_{j,t}$. The total number of possible realizations for this parameter is given by $|\Xi_{j,t}| = R_{j,t}$. Note that because the uncertainty in parameter j is exogenous, it is resolved automatically in each time period t, regardless of the decisions that have been made. In instances where there is only one exogenous parameter, we will frequently drop the j subscript to simplify the notation.

Each scenario in the model corresponds to *one* possible combination of realizations for the uncertain parameters. We assume that these parameters are independent (see Appendix A.1) and that the complete set of scenarios corresponds to *all* possible combinations of their realizations. Accordingly, in the case where the uncertainty is purely exogenous, the complete set of scenarios $\mathcal{R}_X$ is represented by a Cartesian product over the sets of realizations for the exogenous parameters:

$$\mathcal{R}_X := \times_{t\in\mathcal{T}} \left(\times_{j\in\mathcal{J}} \Xi_{j,t}\right) = \left\{ \left(\hat{\xi}^1_{1,1}, \ldots, \hat{\xi}^1_{J,T}\right), \ldots, \left(\hat{\xi}^{R_{1,1}}_{1,1}, \ldots, \hat{\xi}^{R_{J,T}}_{J,T}\right) \right\} \tag{2.1}$$

where we use the subscript X to indicate eXogenous. We enforce a lexicographical ordering on the Cartesian product (and all other Cartesian products in this thesis) based on the index of each realization.[1] Set $\mathcal{R}_X$ corresponds to a scenario tree constructed from all possible combinations of realizations of the exogenous parameters; e.g., Figure 1.2. Note that in this figure there is only one exogenous parameter, so we have dropped the j subscript to simplify the notation, and we have $\mathcal{R}_X = \Xi_1 \times \Xi_2 = \{\hat{\xi}^L_1, \hat{\xi}^H_1\} \times \{\hat{\xi}^L_2, \hat{\xi}^H_2\} = \{(\hat{\xi}^L_1, \hat{\xi}^L_2), (\hat{\xi}^L_1, \hat{\xi}^H_2), (\hat{\xi}^H_1, \hat{\xi}^L_2), (\hat{\xi}^H_1, \hat{\xi}^H_2)\}$. The L and

[1] For example, tuple $(\hat{\xi}^1_{1,1}, \hat{\xi}^1_{1,2})$ would be placed before $(\hat{\xi}^1_{1,1}, \hat{\xi}^2_{1,2})$ based on a comparison of the realization indices (i.e., superscripts) for each respective element: $1 = 1$ for the first element of the tuples, so we proceed to the second element and see that $1 \leq 2$. Tuple $(\hat{\xi}^1_{1,1}, \hat{\xi}^2_{1,2})$ would be placed before $(\hat{\xi}^2_{1,1}, \hat{\xi}^1_{1,2})$ since a comparison of the realization indices for the first element gives $1 < 2$ (and we do not consider the other elements in such a case).

H superscripts here refer to the index of the *low* and *high* realizations for the uncertain parameter ($r = 1$ and $r = 2$), respectively. The cardinality of $\mathcal{R}_X$, or the number of exogenous scenarios, is simply equal to the product of the cardinality of all of the sets in the Cartesian product in Equation (2.1). In other words, the number of exogenous scenarios is equal to the product of the number of realizations for each exogenous parameter,

$$S_X := |\mathcal{R}_X| = \prod_{t\in\mathcal{T}} \prod_{j\in\mathcal{J}} R_{j,t} \tag{2.2}$$

This allows us to index the exogenous scenarios by defining the ordered set of indices $\mathcal{S}_X := \{s : s = 1, 2, \ldots, S_X\}$. Applying this analysis to Figure 1.2, we have $\mathcal{J} = \{1\}$, $\mathcal{T} = \{1, 2\}$, and $R_1 = R_2 = 2$, which gives $S_X = R_1 \cdot R_2 = 2 \cdot 2 = 4$. Accordingly, $\mathcal{S}_X = \{1, 2, 3, 4\}$. Note that if all exogenous parameters have the same number of realizations in all time periods (i.e., $|\Xi_{j,t}| = R \ \forall\ j \in \mathcal{J},\ t \in \mathcal{T}$), as they do in Figure 1.2, Equation (2.2) can be simplified to $S_X = R^{J\cdot T}$.

Since it will be necessary to know the realization of the exogenous parameter $\xi_{j,t}$ in each scenario $s \in \mathcal{S}_X$, we introduce the scenario index to this parameter to define $\xi^s_{j,t}$. Notice, however, that in order to assign a realization value to $\xi^s_{j,t}$ for each scenario, we must first establish a link between the scenario *index* (i.e., $s \in \mathcal{S}_X$) and the *actual scenario* that it represents (i.e., the corresponding tuple in $\mathcal{R}_X$). To do so, we first restate Equation (2.1) with the new notation: $\mathcal{R}_X = \{(\xi^s_{1,1}, \ldots, \xi^s_{J,T}) : s \in \mathcal{S}_X\}$. We then equate the right-hand side of this expression with the right-hand side of Equation (2.1) to give the value of $\xi^s_{j,t}$ for all $j \in \mathcal{J}$, $t \in \mathcal{T}$, and $s \in \mathcal{S}_X$. For instance, for scenario $s = 1$, we are considering the first tuple in $\mathcal{R}_X$. Thus, we have $(\xi^1_{1,1}, \ldots, \xi^1_{J,T}) = (\hat{\xi}^1_{1,1}, \ldots, \hat{\xi}^1_{J,T})$, which implies $\xi^1_{1,1} = \hat{\xi}^1_{1,1}, \ldots, \xi^1_{J,T} = \hat{\xi}^1_{J,T}$. Similarly, for scenario $s = S_X$, we are considering the final tuple in $\mathcal{R}_X$. Now we have $(\xi^{S_X}_{1,1}, \ldots, \xi^{S_X}_{J,T}) = (\hat{\xi}^{R_{1,1}}_{1,1}, \ldots, \hat{\xi}^{R_{J,T}}_{J,T})$, which implies $\xi^{S_X}_{1,1} = \hat{\xi}^{R_{1,1}}_{1,1}, \ldots, \xi^{S_X}_{J,T} = \hat{\xi}^{R_{J,T}}_{J,T}$. The same reasoning applies for all other scenarios in $\mathcal{S}_X$.

2.2 Mathematical Description of Endogenous Uncertainty

Let set $\mathcal{I} := \{i : i = 1, 2, \ldots, I\}$ represent the sources of endogenous uncertainty, and let set $\mathcal{H}_i := \{h : h = 1, 2, \ldots, H_i\}$ define the index of each endogenous-uncertain parameter associated with source $i \in \mathcal{I}$. We define $\theta_{i,h}$ as endogenous parameter $h \in \mathcal{H}_i$ for source $i \in \mathcal{I}$. Recall that we must consider the *source* of uncertainty for each endogenous parameter because the realization for that parameter will only occur once a certain decision has been made for that source. For instance, if source $i = 1$ is an oilfield that has not yet been drilled, the values of the associated endogenous parameters (e.g., oilfield size and initial deliverability) will only be resolved once the oilfield has been drilled. Parameter $\theta_{i,h}$ has a number of possible realizations given by the ordered set $\Theta_{i,h} := \{\hat{\theta}^m_{i,h} : m = 1, 2, \ldots, M_{i,h}\}$, where m refers to the index of one particular realization, and for convenience, we set $\hat{\theta}^1_{i,h} < \hat{\theta}^2_{i,h} < \cdots < \hat{\theta}^{M_{i,h}}_{i,h}$. Thus, if $m = 2$ is the index of the actual realization for endogenous parameter h of source i, we will have $\theta_{i,h} = \hat{\theta}^2_{i,h}$. The total number of possible realizations for this parameter is given by $|\Theta_{i,h}| = M_{i,h}$. We emphasize that, unlike exogenous uncertainty, the resolution of uncertainty in $\theta_{i,h}$ depends on the timing of decisions related to source i and is not an automatic occurrence in each time period. When there is only one endogenous parameter associated with each source of uncertainty, we will often drop the h subscript to simplify the notation.

In the case where the uncertainty is purely endogenous and the uncertain parameters are independent, the complete set of scenarios $\mathcal{R}_N$ is represented by a Cartesian product over the sets of realizations for the endogenous parameters:

$$\mathcal{R}_N := \times_{i\in\mathcal{I}} \left(\times_{h\in\mathcal{H}_i} \Theta_{i,h}\right) = \left\{ \left(\hat{\theta}^1_{1,1}, \ldots, \hat{\theta}^1_{I,H_I}\right), \ldots, \left(\hat{\theta}^{M_{1,1}}_{1,1}, \ldots, \hat{\theta}^{M_{I,H_I}}_{I,H_I}\right) \right\} \tag{2.3}$$

where we use the subscript N to indicate eNdogenous. Set $\mathcal{R}_N$ corresponds to a scenario tree constructed from all possible combinations of realizations of the endogenous parameters; e.g., Figure 1.5. Note that in this figure there is only one endogenous parameter associated with each of the two sources, so we have dropped the h subscript for simplicity in the notation (as we did for the j subscript in Figure 1.2), and we have $\mathcal{R}_N = \Theta_1 \times \Theta_2 = \{\hat{\theta}^L_1, \hat{\theta}^H_1\} \times \{\hat{\theta}^L_2, \hat{\theta}^H_2\} = \{(\hat{\theta}^L_1, \hat{\theta}^L_2), (\hat{\theta}^L_1, \hat{\theta}^H_2), (\hat{\theta}^H_1, \hat{\theta}^L_2), (\hat{\theta}^H_1, \hat{\theta}^H_2)\}$. The cardinality of $\mathcal{R}_N$, or the number of endogenous scenarios, is simply equal to the product of the number of realizations for each endogenous parameter,

$$S_N := |\mathcal{R}_N| = \prod_{i\in\mathcal{I}} \prod_{h\in\mathcal{H}_i} M_{i,h} \tag{2.4}$$

This allows us to index the endogenous scenarios by defining the ordered set of indices $\mathcal{S}_N := \{s : s = 1, 2, \ldots, S_N\}$. In the context of Figure 1.5, we have $\mathcal{I} = \{1, 2\}$, $\mathcal{H}_1 = \mathcal{H}_2 = \{1\}$, and $M_1 = M_2 = 2$. Thus, $S_N = M_1 \cdot M_2 = 2 \cdot 2 = 4$, and $\mathcal{S}_N = \{1, 2, 3, 4\}$. If all endogenous parameters have the same number of realizations (i.e., $|\Theta_{i,h}| = M \;\; \forall\, i \in \mathcal{I},\ h \in \mathcal{H}_i$), as is the case in Figure 1.5, Equation (2.4) can be simplified to $S_N = M^{\sum_{i\in\mathcal{I}} H_i}$.

As in the exogenous case, we also assign the index s to the endogenous parameter $\theta_{i,h}$ to indicate the parameter's realization in each scenario; i.e., $\theta^s_{i,h}$. Using this notation, we restate Equation (2.3) as $\mathcal{R}_N = \{(\theta^s_{1,1}, \ldots, \theta^s_{I,H_I}) : s \in \mathcal{S}_N\}$, and equate the right-hand side of this expression with the right-hand side of Equation (2.3) to give the value of $\theta^s_{i,h}$ for all $i \in \mathcal{I}$, $h \in \mathcal{H}_i$, and $s \in \mathcal{S}_N$.

2.3 Mathematical Description of Endogenous and Exogenous Uncertainties

We now consider the case where we have both endogenous and exogenous uncertain parameters. Because these parameters are entirely independent of one another, we must ensure that we can observe *any* possible combination of realizations for the exogenous parameters, regardless of the outcome for the endogenous parameters (and vice versa). Accordingly, we generate the complete set of scenarios $\mathcal{R}$ by the Cartesian product of all possible combinations of realizations of the endogenous parameters and all possible combinations of realizations of the exogenous parameters, $\mathcal{R}_N \times \mathcal{R}_X$:

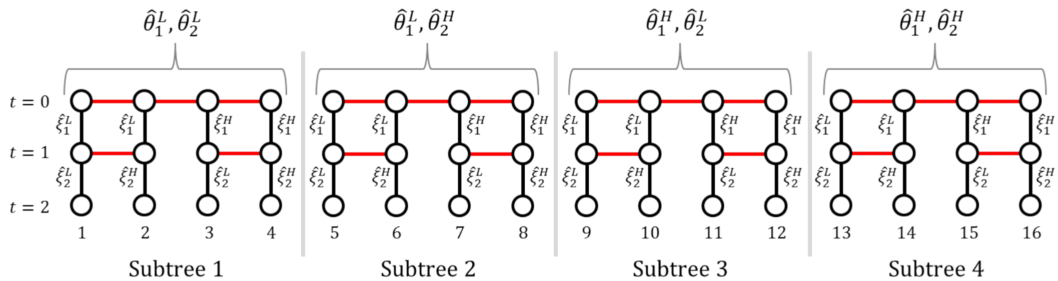
$$
\begin{aligned}
\mathcal{R} :=& \mathcal{R}_N \times \mathcal{R}_X \\
=& \Big\{ \Big(\hat{\theta}^1_{1,1}, \ldots, \hat{\theta}^1_{I,H_I}, \hat{\xi}^1_{1,1}, \ldots, \hat{\xi}^1_{J,T}\Big), \ldots, \Big(\hat{\theta}^1_{1,1}, \ldots, \hat{\theta}^1_{I,H_I}, \hat{\xi}^{R_{1,1}}_{1,1}, \ldots, \hat{\xi}^{R_{J,T}}_{J,T}\Big), \ldots, \\
& \Big(\hat{\theta}^{M_{1,1}}_{1,1}, \ldots, \hat{\theta}^{M_{I,H_I}}_{I,H_I}, \hat{\xi}^1_{1,1}, \ldots, \hat{\xi}^1_{J,T}\Big), \ldots, \Big(\hat{\theta}^{M_{1,1}}_{1,1}, \ldots, \hat{\theta}^{M_{I,H_I}}_{I,H_I}, \hat{\xi}^{R_{1,1}}_{1,1}, \ldots, \hat{\xi}^{R_{J,T}}_{J,T}\Big) \Big\}
\end{aligned}
\tag{2.5}
$$

Set $\mathcal{R}$ corresponds to a "composite" scenario tree that includes all possible combinations of realizations of the endogenous and exogenous parameters. Although there are other ways to generate such a set (e.g., $\mathcal{R}_X \times \mathcal{R}_N$; see Appendix A.2), we focus our attention on this approach since the resulting scenario tree has a structure that can be exploited to significantly reduce the dimensionality of the model (as will be discussed in Chapter 4). For the remainder of this thesis, excluding Chapter 6, we assume that the scenario tree has been generated in this manner. The total number of scenarios (i.e., the cardinality of $\mathcal{R}$) is equal to the product of the number of endogenous scenarios and the number of exogenous scenarios,

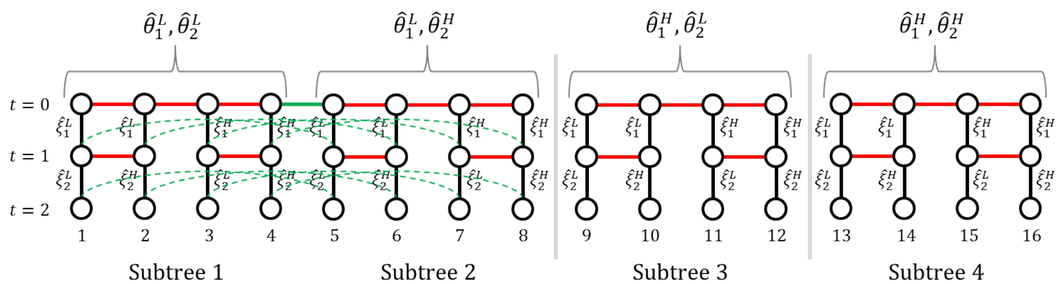
$$
S := |\mathcal{R}| = S_N \cdot S_X \tag{2.6}
$$

We index the set of scenarios by defining the ordered set of indices $\mathcal{S} := \{s \,:\, s = 1, 2, \ldots, S\}$. We use this set to restate Equation (2.5) as $\mathcal{R} = \{(\theta^s_{1,1}, \ldots, \theta^s_{I,H_I}, \xi^s_{1,1}, \ldots, \xi^s_{J,T}) \,:\, s \in \mathcal{S}\}$, and we equate the right-hand side of this expression with the right-hand side of Equation (2.5) to give the values of $\theta^s_{i,h}$ and $\xi^s_{j,t}$ for all $i \in \mathcal{I}$, $h \in \mathcal{H}_i$, $j \in \mathcal{J}$, $t \in \mathcal{T}$, and $s \in \mathcal{S}$.

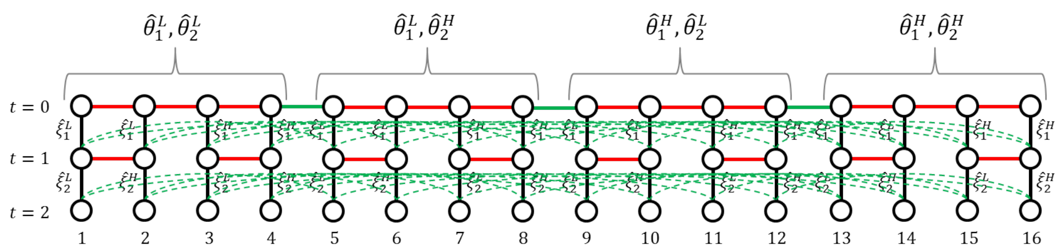
The generation of the composite scenario tree is shown in Figure 2.1. We consider the case where we have one exogenous parameter with two realizations (*low* or *high*) in each time period, two endogenous parameters each with two realizations (also *low* or *high*), and a time horizon consisting of two time periods (i.e., 3 stages). In generating the full set of scenarios, it follows that set $\mathcal{R}_N$ corresponds to the endogenous scenario tree in Figure 1.5, and set $\mathcal{R}_X$ corresponds to the exogenous scenario tree in Figure 1.2. By Equation (2.5), the composite scenario tree resulting from all possible combinations of realizations of these parameters will consist of the scenarios given by $\mathcal{R} = \mathcal{R}_N \times \mathcal{R}_X = \{(\hat{\theta}^L_1, \hat{\theta}^L_2, \hat{\xi}^L_1, \hat{\xi}^L_2), \ldots, (\hat{\theta}^H_1, \hat{\theta}^H_2, \hat{\xi}^H_1, \hat{\xi}^H_2)\}$. The number of scenarios in this composite tree is given by Equation (2.6), which yields $S = 4 \cdot 4 = 16$. Thus, $\mathcal{S} = \{1, 2, \ldots, 16\}$. Figure 2.1a clarifies the mathematical procedure for generating the composite scenario tree by providing the graphical analogue: we simply copy the exogenous scenario tree (Figure 1.2) for each possible combination of realizations of the endogenous parameters. This gives rise to multiple "subtrees" (four in this case). Viewed another way, we have essentially replaced each scenario in the endogenous scenario tree (Figure 1.5) with an exogenous subtree. Here we use the alternative form of the exogenous tree, Figure 1.2b, so that we can easily apply scenario decomposition later in this thesis. With the full set of scenarios in place, we then link these subtrees by adding first-period and endogenous non-anticipativity constraints (shown by solid and dotted green lines, respectively) which enforce equality between indistinguishable nodes. This process is partially illustrated in Figure 2.1b for the links between subtrees 1 and 2 only. By adding the remaining links between the subtrees, we end up with the complete composite scenario tree shown in Figure 2.1c. (We provide the reasoning behind our choice of these particular non-anticipativity constraints in Chapter 4.) It is clear how quickly these problems can grow, as the composite tree is significantly more complex than either Figure 1.2 or Figure 1.5 alone. Note that in the figures, as before, we are only considering one exogenous parameter and one endogenous parameter for each of the two sources, so we have dropped the j and h subscripts, respectively.



(a) **Step 1:** Copy the exogenous scenario tree for each possible combination of realizations of the endogenous parameters.



(b) **Step 2:** Link these 'subtrees' by adding first-period and endogenous non-anticipativity constraints (shown by solid and dotted green lines, respectively, between subtrees 1 and 2).



(c) Complete composite scenario tree.

Figure 2.1: Procedure for generating a 'composite' scenario tree. This tree captures all possible combinations of realizations for both the endogenous and exogenous uncertain parameters.

Notice that within each subtree, all scenarios have the same possible endogenous realizations, and these endogenous realizations are the only distinguishing characteristic between each of the subtrees. Thus, if the uncertainty in the endogenous parameters is not resolved by the end of the time horizon, all of the subtrees will be exactly identical (since all of the conditional, dotted green lines will have become solid lines, enforcing non-anticipativity between the corresponding nodes).

It is also interesting to note that if we assume expected values for each of the endogenous parameters, thereby neglecting the endogenous uncertainty, we recover the original exogenous tree

(Figure 1.2). If we instead assume expected values for each of the exogenous parameters, thereby neglecting the exogenous uncertainty, we recover the original endogenous tree (Figure 1.5).

The concept of subtrees will be used extensively in the definitions of parameters and sets later in this thesis, so we define parameter $Sub(s)$ to return the subtree number of each scenario in the composite tree. This number is calculated as the ceiling of the ratio of the scenario index s and the number of scenarios in each subtree, S_X:

$$Sub(s) := \left\lceil \frac{s}{S_X} \right\rceil \qquad \forall\, s \in \mathcal{S} \tag{2.7}$$

Note that S_X, defined in Equation (2.2), is the number of scenarios in each subtree since each subtree is simply an exogenous tree. For Figure 2.1c, scenarios 1–4 are in subtree 1, scenarios 5–8 are in subtree 2, scenarios 9–12 are in subtree 3, and scenarios 13–16 are in subtree 4. Accordingly, Equation (2.7) returns $Sub(2) = \lceil 2/4 \rceil = 1$, $Sub(6) = \lceil 6/4 \rceil = 2$, $Sub(10) = \lceil 10/4 \rceil = 3$, and $Sub(14) = \lceil 14/4 \rceil = 4$.

2.4 Realization Probabilities

Following directly from the theory presented in the previous sections, we now briefly discuss realization probabilities. For each realization r of exogenous parameter $xi_{j,t}$, there is a corresponding probability $\hat{\upsilon}^r_{j,t}$ of this value occurring. The realization values are defined in set $\Xi_{j,t} := \{\hat{\xi}^r_{j,t} : r = 1, 2, \dots, R_{j,t}\}$, and we now define the set of probabilities $\Upsilon_{j,t} := \{\hat{\upsilon}^r_{j,t} : r = 1, 2, \dots, R_{j,t}\}$. Note that this set is indexed in the same order as the realization values. For instance, if we consider the first realization, $r = 1$, the realization value is given by set $Xi_{j,t}$ as $\hat{\xi}^1_{j,t}$, and the corresponding probability is given by set $\Upsilon_{j,t}$ as $\hat{\upsilon}^1_{j,t}$. Also, note that since these realizations represent all possible outcomes for parameter $\xi_{j,t}$ from a discretized probability distribution, the probabilities must sum to 1; i.e., $\sum_{r=1}^{R_{j,t}} \hat{\upsilon}^r_{j,t} = 1 \ \forall\, j \in \mathcal{J},\ t \in \mathcal{T}$.

Each realization m of endogenous parameter $\theta_{i,h}$ also has a corresponding probability $\hat{\omega}^m_{i,h}$ that it will occur. The realization values are defined in set $\Theta_{i,h} := \{\hat{\theta}^m_{i,h} : m = 1, 2, \dots, M_{i,h}\}$, and we define the set of probabilities $\Omega_{i,h} := \{\hat{\omega}^m_{i,h} : m = 1, 2, \dots, M_{i,h}\}$. As is the case for the exogenous parameters, this set is indexed in the same order as the realization values. Thus, for $m = 1$, we have the realization value $\hat{\theta}^1_{i,h}$ from set $\Theta_{i,h}$, and the corresponding probability $\hat{\omega}^1_{i,h}$ from set $\Omega_{i,h}$. Again, these realizations represent all possible outcomes for parameter $\theta_{i,h}$ from a discretized probability distribution, so the probabilities must sum to 1; i.e., $\sum_{m=1}^{M_{i,h}} \hat{\omega}^m_{i,h} = 1 \ \forall\, i \in \mathcal{I},\ h \in \mathcal{H}_i$.

Recall that set $\mathcal{R}_X$ (Equation (2.1)), set $\mathcal{R}_N$ (Equation (2.3)), and set $\mathcal{R}$ (Equation (2.5)) give the realization values for each scenario $s \in \mathcal{S}_X$, $s \in \mathcal{S}_N$, and $s \in \mathcal{S}$, respectively. Since each of these realizations has a corresponding probability, we can find the set of realization probabilities for each scenario by simply substituting the realization values in each expression with their corresponding probabilities. Specifically, for the case where the uncertainty is purely exogenous, the set of realization probabilities for each scenario is defined by,

$$
\begin{aligned}
\mathcal{P}_X &:= \times_{t\in\mathcal{T}} \left(\times_{j\in\mathcal{J}} \Upsilon_{j,t}\right) \\
&= \left\{ \left(\hat{\upsilon}^1_{1,1}, \dots, \hat{\upsilon}^1_{J,T}\right), \dots, \left(\hat{\upsilon}^{R_{1,1}}_{1,1}, \dots, \hat{\upsilon}^{R_{J,T}}_{J,T}\right) \right\} \\
&= \left\{ \left(\upsilon^s_{1,1}, \dots, \upsilon^s_{J,T}\right) : \, s \in \mathcal{S}_X \right\}
\end{aligned}
\tag{2.8}
$$

where $\upsilon^s_{j,t}$ refers to the realization probability of exogenous parameter $\xi_{j,t}$ in scenario s. In the case where the uncertainty is purely endogenous, the set of realization probabilities for each scenario is defined by,

$$
\begin{aligned}
\mathcal{P}_N &:= \times_{i\in\mathcal{I}} \left(\times_{h\in\mathcal{H}_i} \Omega_{i,h}\right) \\
&= \left\{ \left(\hat{\omega}^1_{1,1}, \dots, \hat{\omega}^1_{I,H_I}\right), \dots, \left(\hat{\omega}^{M_{1,1}}_{1,1}, \dots, \hat{\omega}^{M_{I,H_I}}_{I,H_I}\right) \right\} \\
&= \left\{ \left(\omega^s_{1,1}, \dots, \omega^s_{I,H_I}\right) : \, s \in \mathcal{S}_N \right\}
\end{aligned}
\tag{2.9}
$$

where $\omega^s_{i,h}$ refers to the realization probability of endogenous parameter $\theta_{i,h}$ in scenario s. And for the case of primary interest, where there are both endogenous and exogenous uncertainties, the set of realization probabilities for each scenario is defined by,

$$
\begin{aligned}
\mathcal{P} :=& \mathcal{P}_N \times \mathcal{P}_X \\
=& \left\{ \left(\hat{\omega}^1_{1,1}, \dots, \hat{\omega}^1_{I,H_I}, \hat{\upsilon}^1_{1,1}, \dots, \hat{\upsilon}^1_{J,T}\right), \dots, \left(\hat{\omega}^{M_{1,1}}_{1,1}, \dots, \hat{\omega}^{M_{I,H_I}}_{I,H_I}, \hat{\upsilon}^{R_{1,1}}_{1,1}, \dots, \hat{\upsilon}^{R_{J,T}}_{J,T}\right) \right\} \\
=& \left\{ \left(\omega^s_{1,1}, \dots, \omega^s_{I,H_I}, \upsilon^s_{1,1}, \dots, \upsilon^s_{J,T}\right) : \, s \in \mathcal{S} \right\}
\end{aligned}
\tag{2.10}
$$

The probability of each scenario is given by p^s, and is equal to the product of all of the realization probabilities in scenario s:

$$
p^s := \left(\prod_{i\in\mathcal{I}} \prod_{h\in\mathcal{H}_i} \omega^s_{i,h} \right) \cdot \left(\prod_{t\in\mathcal{T}} \prod_{j\in\mathcal{J}} \upsilon^s_{j,t} \right) \qquad \forall \, s \in \mathcal{S} \tag{2.11}
$$

Since the elements sum to 1 in each set of realization probabilities ($\Upsilon_{j,t}$ and $\Omega_{i,h}$), and we are simply taking the product of each possible combination of all of these elements, the sum over all of these products must also be 1 (see Appendix A.3 for the simple proof). In other words, the total probability over all scenarios must sum to 1: $\sum_{s\in\mathcal{S}} p^s = 1$.

Chapter 3

Models

A simple MILP formulation for a deterministic multi-period planning problem is given in model (MPD). Variable vectors y_t represent investment and operation decisions that are made at the beginning of each time period t (e.g., whether or not to drill a particular oilfield, the processing capacity of a new offshore oil facility, etc.), and variable vectors x_t represent operation decisions that typically follow these investment decisions (e.g., the oil flow rate from a field to a newly-installed facility). Variable vectors w_t are commonly referred to as state variables and represent calculated quantities associated with each time period, such as intermediate flow rates and economic values like total operating cost. Vectors y_t, x_t, and w_t may each have integer and continuous components.

(MPD)

$$\min_{y,x} \phi_D = \sum_{t \in \mathcal{T}} \left({}^{y}c_t y_t + {}^{x}c_t x_t + {}^{w}c_t w_t \right) \tag{3.1}$$

$$\text{s.t.} \quad \sum_{\tau=1}^{t} \left({}^{y}A_{\tau,t} y_\tau + {}^{x}A_{\tau,t} x_\tau + {}^{w}A_{\tau,t} w_\tau \right) \leq a_t \qquad \forall\, t \in \mathcal{T} \tag{3.2}$$

$$y_t \in \mathcal{Y}_t, \;\; x_t \in \mathcal{X}_t, \;\; w_t \in \mathcal{W}_t \qquad \forall\, t \in \mathcal{T} \tag{3.3}$$

The objective function, Equation (3.1), minimizes the total cost associated with decisions y_t and x_t, and state variables w_t. For convenience, we adopt the notation of Goel and Grossmann (2006) and specify the corresponding cost coefficients through row vectors ${}^{y}c_t$, ${}^{x}c_t$, and ${}^{w}c_t$, respectively. Equation (3.2) represents constraints that govern the decisions in each time period $t \in \mathcal{T}$, as well as constraints that link decisions across time periods. This equation also includes equality constraints such as those that assign values to w_t. The constraint coefficients for variables y_t, x_t, and w_t are given by matrices ${}^{y}A_{\tau,t}$, ${}^{x}A_{\tau,t}$, and ${}^{w}A_{\tau,t}$, respectively, and the right-hand side is given by column vectors a_t. Bounds and integrality restrictions on the variables are specified by mixed-integer sets $\mathcal{Y}_t$, $\mathcal{X}_t$, and $\mathcal{W}_t$ in Equation (3.3).

In the following sections, we will show how this model is transformed into a multistage stochastic programming problem in the case of: (1) exogenous uncertainty, (2) endogenous uncertainty, and (3) both endogenous and exogenous uncertainties. These stochastic programming models (largely inspired by the work of Goel and Grossmann (2006) will be presented in deterministic-equivalent form using the non-anticipativity approach. For additional background, we refer the reader to Rockafellar and Wets (1991), Ruszczyński (1997), and Birge and Louveaux (2011).

3.1 MSSP Formulation for Exogenous Uncertainty

The multistage stochastic programming formulation of model (MPD) in the case of exogenous uncertainties is given in model (MSSP_X). Notice that variables y_t, x_t, and w_t have been indexed for each scenario $s \in \mathcal{S}_X$ to indicate the respective decisions and calculated quantities in each scenario. As we are now modeling under a multistage stochastic programming framework, the decision-making process is structured as shown in Figure 1.1. Specifically, variables y_t^s refer to here-and-now decisions, and variables x_t^s refer to recourse decisions. Recall that decisions y_t^s are implemented at the beginning of each time period t of scenario s. At some point after these decisions are made, but during t, the uncertainty in exogenous parameter $\xi_{j,t}$ is resolved. Recourse decisions x_t^s are then made as corrective action at the end of the period in response to this new information. Based on the values of y_t^s and x_t^s, state variables w_t^s are calculated.

($\mathbf{MSSP_X}$)

$$\min_{y,x} \phi_X = \sum_{s \in \mathcal{S}_X} p^s \sum_{t \in \mathcal{T}} \left({}^y c_t^s y_t^s + {}^x c_t^s x_t^s + {}^w c_t^s w_t^s \right) \tag{3.4}$$

$$\text{s.t.} \quad \sum_{\tau=1}^{t} \left({}^y A_{\tau,t}^s y_\tau^s + {}^x A_{\tau,t}^s x_\tau^s + {}^w A_{\tau,t}^s w_\tau^s \right) \leq a_t^s \qquad \forall\, t \in \mathcal{T},\;\; s \in \mathcal{S}_X \tag{3.5}$$

$$y_1^s = y_1^{s'} \qquad \forall\, (s, s') \in \mathcal{SP}_F \tag{3.6}$$

$$x_t^s = x_t^{s'} \qquad \forall\, (t, s, s') \in \mathcal{SP}_X \tag{3.7}$$

$$y_{t+1}^s = y_{t+1}^{s'} \qquad \forall\, (t, s, s') \in \mathcal{SP}_X \tag{3.8}$$

$$y_t^s \in \mathcal{Y}_t^s,\;\; x_t^s \in \mathcal{X}_t^s,\;\; w_t^s \in \mathcal{W}_t^s \qquad \forall\, t \in \mathcal{T},\;\; s \in \mathcal{S}_X \tag{3.9}$$

Notice that only fairly simple changes are required to convert the deterministic model (MPD) to the multistage stochastic programming model (MSSP_X). In particular, the objective function, Equation (3.4), now minimizes the total *expected* cost by taking the weighted sum of the costs in each scenario based on the probability of each scenario, p^s. The cost coefficients have been indexed for all $s \in \mathcal{S}_X$ to allow for the possibility of different cost realizations in each scenario. Additionally, the constraints governing the decisions in each time period, represented by Equation (3.5), are simply applied for each $s \in \mathcal{S}_X$. Note that like the cost coefficients, the constraint coefficients and right-hand side have also been indexed for s to allow for different realizations in each scenario. In other words, exogenous parameters $\xi_{j,t}$ may enter the model through the objective function and/or the constraints (via the constraint coefficients and/or the right-hand side).

The most significant difference between the models is the introduction of non-anticipativity constraints, given by Equations (3.6)–(3.8). Each scenario in model (MSSP_X) represents a different instance of the deterministic planning problem with different realizations for the uncertain parameters, and the non-anticipativity constraints link these scenarios together, as shown in Figure 1.2b.

Equation (3.6) enforces non-anticipativity between all scenarios at the beginning of the first time period. As previously stated, this is due to the fact that all scenarios are indistinguishable at this time, and we must make the same here-and-now decisions (first-stage decisions) in all scenarios. For the remainder of this thesis, we will simply refer to these constraints as *first-period NACs*. Note that when discussing indistinguishability, we refer specifically to the indistinguishability *between two scenarios* s and s'. Accordingly, we will consider pairs of indistinguishable scenarios (s, s') in each time period t for which we must enforce non-anticipativity. We will also define sets of these tuples in order to simplify the notation in our models. For the first-period NACs, the corresponding *first-period scenario pairs* are elements of set $\mathcal{SP}_F$, given by:

$$\mathcal{SP}_F := \mathcal{A} \tag{3.10}$$

where $\mathcal{A}$ is the set of scenario pairs for which s and s' are adjacent. This will be discussed in greater detail in Section 4.1 (see Equation (4.2)).

Equations (3.7) and (3.8) represent non-anticipativity constraints for all remaining stages. In particular, if scenarios s and s' are indistinguishable in time period t in terms of the resolution of exogenous uncertainty, we must make the same recourse decisions at the end of this period (enforced by Equation (3.7)), as well as the same here-and-now decisions at the beginning of the next time period, $t+1$ (enforced by Equation (3.8)). We will refer to these constraints as *exogenous NACs*. The corresponding set of *exogenous scenario pairs* is given by set $\mathcal{SP}_X$ and is defined as:

$$\mathcal{SP}_X := \Big\{(t, s, s') : \, t \in \mathcal{T} \setminus \{T\}, \;\; (s, s') \in \mathcal{A}, \;\; Sub(s) = Sub(s'), \;\; Q_t^{s,s'} = True\Big\} \tag{3.11}$$

where $Sub(s) = Sub(s')$ ensures that s and s' are in the same subtree, and $Q_t^{s,s'}$ is a Boolean parameter that indicates whether or not these scenarios are indistinguishable in time period t. This will be discussed in Section 4.2 (see Equations (4.3) and (4.4)).

Notice that at the beginning of the final time period, the NACs for the here-and-now decisions correspond to Equation (3.8) with $t = T - 1$. Also, specifically in the exogenous case, NACs never apply for the recourse decisions at the end of the final time period (final-stage decisions); this is because the leaf nodes must refer to independent states or else we would have duplicate scenarios in the tree (see Figure 1.2). It follows, then, that we can entirely exclude time period $t = T$ from the definition of set $\mathcal{SP}_X$, as indicated in Equation (3.11).

We also note that we never express NACs for state variables w_t^s, in *any* time period, since these variables are calculated based on the values of decision variables y_t^s and x_t^s. In other words, non-anticipativity for w_t^s is implicitly enforced by Equations (3.6)–(3.8).

Similar to the deterministic formulation, bounds and integrality restrictions on the variables are specified by the mixed-integer sets in Equation (3.9).

3.2 MSSP Formulation for Endogenous Uncertainty

The multistage stochastic programming formulation of model (MPD) in the case of endogenous uncertainties is given in model ($\mathrm{MSSP_N}$). This model has been adapted from Goel and Grossmann (2006) and is presented in hybrid mixed-integer linear disjunctive form.

Previously, we used vector y_t^s to represent all here-and-now decisions in each time period t of scenario s. In the case of endogenous uncertainties, however, this approach does not provide us with particularly detailed information. As can be seen in Figure 1.3, it is not immediately obvious which decisions are associated with a source $i \in \mathcal{I}$. This is a very important modeling consideration, since such decisions uniquely determine whether or not the uncertainty in parameter $\theta_{i,h}$ can be resolved in scenario s. Accordingly, we define vector $b_{i,t}^s$ to identify those binary decisions that are strictly associated with a particular source i (e.g., to drill an oilfield of uncertain size and initial deliverability). To keep the notation simple, we will continue to use y_t^s to represent all other here-and-now decisions.

It is often the case that the uncertainty in some (or all) endogenous parameters cannot be resolved within the first few time periods of the planning horizon. For instance, in an oilfield development planning problem, we may assume that any oilfield must be in production for a certain number of years before the size of the field can be established. Before that amount of time has passed, the sizes of all fields are uncertain, and any scenarios that differ only in the possible realizations of field sizes must be indistinguishable. Thus, for these initial time periods, the corresponding conditional NACs can be expressed as equality constraints (Tarhan et al., 2009; Colvin and Maravelias, 2010; Gupta and Grossmann, 2014a). To model this, we denote the number of initial 'equality' periods as $T_E^{i'}$, and partition the set of time periods $\mathcal{T}$ into the set of these initial periods $\mathcal{T}_E^{i'} := \{t : t = 1, \dots, T_E^{i'}\}$ and the set of remaining 'conditional' time periods $\mathcal{T}_C^{i'} := \{t : t = T_E^{i'} + 1, \dots, T\}$, where $T_E^{i'} < T$, for all $i' \in \mathcal{I}$. We use index i' in these definitions so as not to conflict with index i of $b_{i,t}^s$. Note that if $T_E^{i'} = 0$, the corresponding sets reduce to $\mathcal{T}_E^{i'} := \emptyset$ and $\mathcal{T}_C^{i'} := \mathcal{T}$. Further note that these parameters and subsets are defined for each $i' \in \mathcal{I}$ since the number of initial periods may not be the same for all sources of endogenous uncertainty.

(MSSP$_\mathbf{N}$)

$$\min_{b,y,x} \phi_N = \sum_{s \in \mathcal{S}_N} p^s \sum_{t \in \mathcal{T}} \left({}^y c_t^s y_t^s + {}^x c_t^s x_t^s + {}^w c_t^s w_t^s + \sum_{i \in \mathcal{I}} {}^b c_{i,t}^s b_{i,t}^s \right) \tag{3.12}$$

$$\text{s.t.} \quad \sum_{\tau=1}^{t} \left({}^y A_{\tau,t}^s y_\tau^s + {}^x A_{\tau,t}^s x_\tau^s + {}^w A_{\tau,t}^s w_\tau^s + \sum_{i \in \mathcal{I}} {}^b A_{i,\tau,t}^s b_{i,\tau}^s \right) \le a_t^s \qquad \forall\, t \in \mathcal{T},\ \ s \in \mathcal{S}_N \tag{3.13}$$

$$b_{i,1}^s = b_{i,1}^{s'} \qquad \forall\, (s,s') \in \mathcal{SP}_F,\ \ i \in \mathcal{I} \tag{3.14}$$

$$y_1^s = y_1^{s'} \qquad \forall\, (s,s') \in \mathcal{SP}_F \tag{3.6}$$

$$x_t^s = x_t^{s'} \qquad \forall\, (t,s,s') \in \mathcal{SP}_N,\ \ t \in \mathcal{T}_E^{i'},\ \ \{i'\} = \hat{\mathcal{D}}^{s,s'} \tag{3.15}$$

$$b_{i,t+1}^s = b_{i,t+1}^{s'} \qquad \forall\, (t,s,s') \in \mathcal{SP}_N,\ \ t \in \mathcal{T}_E^{i'},\ \ \{i'\} = \hat{\mathcal{D}}^{s,s'},\ \ i \in \mathcal{I} \tag{3.16}$$

$$y_{t+1}^s = y_{t+1}^{s'} \qquad \forall\, (t,s,s') \in \mathcal{SP}_N,\ \ t \in \mathcal{T}_E^{i'},\ \ \{i'\} = \hat{\mathcal{D}}^{s,s'} \tag{3.17}$$

$$\begin{bmatrix} Z_t^{s,s'} & & \\ x_t^s = x_t^{s'} & & \\ b_{i,t+1}^s = b_{i,t+1}^{s'} & \forall\, i \in \mathcal{I}, & t < T \\ y_{t+1}^s = y_{t+1}^{s'} & & t < T \end{bmatrix} \vee \left[\neg Z_t^{s,s'} \right] \qquad \forall\, (t,s,s') \in \mathcal{SP}_N,\ \ t \in \mathcal{T}_C^{i'},\ \ \{i'\} = \hat{\mathcal{D}}^{s,s'} \tag{3.18}$$

$$Z_t^{s,s'} \Leftrightarrow F(b_{i',1}^s, b_{i',2}^s, \dots, b_{i',t}^s) \qquad \forall\, (t,s,s') \in \mathcal{SP}_N,\ \ t \in \mathcal{T}_C^{i'},\ \ \{i'\} = \hat{\mathcal{D}}^{s,s'} \tag{3.19}$$

$$b_{i,t}^s \in \{0,1\}, \;\; y_t^s \in \mathcal{Y}_t^s, \;\; x_t^s \in \mathcal{X}_t^s, \;\; w_t^s \in \mathcal{W}_t^s \qquad \forall\, i \in \mathcal{I}, \;\; t \in \mathcal{T}, \;\; s \in \mathcal{S}_N \tag{3.20}$$

$$Z_t^{s,s'} \in \{True, False\} \qquad \forall\, (t,s,s') \in \mathcal{SP}_N, \;\; t \in \mathcal{T}_C^{i'}, \;\; \{i'\} = \hat{\mathcal{D}}^{s,s'} \tag{3.21}$$

The objective function, Equation (3.12), is very similar to that of model ($\mathrm{MSSP_X}$). Notice that the only differences from Equation (3.4) are the following: we have now introduced decision variables $b_{i,t}^s$ and the corresponding row vector of cost coefficients, ${}^b c_{i,t}^s$ (which requires a summation over all sources $i \in \mathcal{I}$), and the set of scenarios is now given by $\mathcal{S}_N$. Likewise, Equation (3.13) only differs from Equation (3.5) by the same changes, except the corresponding coefficient matrix is ${}^b A_{i,\tau,t}^s$. Endogenous parameters $\theta_{i,h}$ may enter the model through the objective function and/or the constraints, as was the case for exogenous parameters $\xi_{j,t}$ in model ($\mathrm{MSSP_X}$). First-period NACs still apply, and accordingly, we express them for our here-and-now decisions in Equation (3.14) and (from model ($\mathrm{MSSP_X}$)) Equation (3.6).

Each scenario pair of time period $t \in \mathcal{T}_E^{i'}$ in set $\mathcal{SP}_N$ represents two scenarios s and s' that are indistinguishable at that time in terms of the resolution of endogenous uncertainty. This set of *endogenous scenario pairs*, $\mathcal{SP}_N$, is given by:

$$\begin{aligned} \mathcal{SP}_N^{i',h,l} := \Big\{ (t,s,s') : \; & t \in \mathcal{T}, \;\; s, s' \in \left({}^N\mathcal{G}_{i',h}^l \cap \mathcal{U}_t^{i',h}\right), \\ & s' = \min_{\hat{s}'} \left(\hat{s}' \in \left({}^N\mathcal{G}_{i',h}^l \cap \mathcal{U}_t^{i',h}\right), \; \hat{s}' > s\right), \\ & s < \max_{\hat{s}} \left(\hat{s} \in \left({}^N\mathcal{G}_{i',h}^l \cap \mathcal{U}_t^{i',h}\right)\right), \\ & \{(i',h)\} = \mathcal{D}^{s,s'} \Big\} \;\; \forall\, l \in \mathcal{L}_{i',h}, \;\; i' \in \mathcal{I}, \;\; h \in \mathcal{H}_{i'} \end{aligned} \tag{3.22}$$

$$\mathcal{SP}_N := \bigcup_{i' \in \mathcal{I}} \left(\bigcup_{h \in \mathcal{H}_{i'}} \left(\bigcup_{l \in \mathcal{L}_{i',h}} \mathcal{SP}_N^{i',h,l} \right) \right) \tag{3.23}$$

where, in Equation (3.22), we first determine the set of scenario pairs in each time period t corresponding to endogenous parameter $\theta_{i',h}$ for all $i' \in \mathcal{I}$ and $h \in \mathcal{H}_{i'}$. We then take the union of all of these sets in Equation (3.23).

Given the complexity of Equation (3.22), before continuing, we briefly describe the primary aspects of this expression. Sets ${}^N\mathcal{G}_{i',h}^l$, indexed by $l \in \mathcal{L}_{i',h}$, represent endogenous scenario groups corresponding to $\theta_{i',h}$. For each endogenous parameter $\theta_{i',h}$, set $\mathcal{U}_t^{i',h}$ provides a sufficient subset of scenarios that are available for pairing from that parameter's respective groups in time period t. These sets, $\mathcal{U}_t^{i',h}$, are defined in a sequential manner in which we successively eliminate scenarios based on the pairs that have already been formed. We then obtain a sufficient subset of each group specific to time period t via sets ${}^N\mathcal{G}_{i',h}^l \cap \mathcal{U}_t^{i',h}$, which we refer to as *reduced* endogenous scenario groups. We pair off consecutive scenarios in each of these reduced groups, where set $\mathcal{D}^{s,s'}$ indicates the specific parameter $\theta_{i',h}$ for which s and s' differ in possible realizations. This will be discussed in detail in Section 4.3.

Accordingly, for each of the scenario pairs of time period $t \in \mathcal{T}_E^{i'}$ in set $\mathcal{SP}_N$, we enforce non-anticipativity between the respective scenarios s and s' as shown in Equations (3.15)–(3.17), exactly as we would in the exogenous case (see Equations (3.7) and (3.8) for comparison). Notice that the only differences here are that we are considering different scenario pairs, and the set of time periods is source dependent. The particular source i' of set $\mathcal{T}_E^{i'}$ is determined by set $\hat{\mathcal{D}}^{s,s'}$; specifically, this set indicates the source in which scenarios s and s' differ in the possible realization of some endogenous parameter (see Equation (4.21) in Section 4.3). Note that we will refer to these equality constraints as *fixed endogenous NACs*.

Each scenario pair of time period $t \in \mathcal{T}_C^{i'}$ in set $\mathcal{SP}_N$ represents two scenarios s and s' that *may* be indistinguishable at that time (where the particular source i' of set $\mathcal{T}_C^{i'}$ is determined by set $\hat{\mathcal{D}}^{s,s'}$). Recall that in the exogenous case, we know in advance whether two scenarios will differ in parameter realizations in time period t. For endogenous parameters, however, the timing of realizations depends on decisions $b_{i,t}^s$, so we can no longer use simple equality constraints to apply non-anticipativity. Instead, we *conditionally* enforce non-anticipativity between these scenarios (see Figure 1.5), as shown by the disjunctive constraints in Equation (3.18). Boolean variable $Z_t^{s,s'}$ indicates whether s and s' are indistinguishable by the end of time period t, and if so, the value is $True$ and the NACs are enforced. If they are distinguishable, the value is $False$ and the constraints are ignored. We will refer to these conditional constraints as *conditional endogenous NACs*. Note that since we make here-and-now decisions for the *next* time period $(t+1)$ based on indistinguishability information revealed up until the current time t, NACs for decisions $b_{i,t+1}^s$ and y_{t+1}^s must be restricted to $t < T$; this is, of course, because we cannot make new here-and-now decisions at the end of the time horizon. This restriction is implicit in the exogenous model because non-anticipativity does not apply at the end of the final time period.

Using a big-M reformulation (Trespalacios and Grossmann, 2014), we can rewrite the disjunctive constraints (3.18) as inequality constraints (3.24)–(3.26), where UB denotes the upper bound of the respective variable.[1]

$$-x_t^{UB}(1 - z_t^{s,s'}) \leq x_t^s - x_t^{s'} \leq x_t^{UB}(1 - z_t^{s,s'}) \quad \forall\,(t,s,s') \in \mathcal{SP}_N,\;\; t \in \mathcal{T}_C^{i'},\;\; \{i'\} = \hat{\mathcal{D}}^{s,s'} \tag{3.24}$$

$$-(1 - z_t^{s,s'}) \leq b_{i,t+1}^s - b_{i,t+1}^{s'} \leq (1 - z_t^{s,s'}) \quad \forall\,(t,s,s') \in \mathcal{SP}_N,\;\; t \in \mathcal{T}_C^{i'},\;\; t < T,\;\; \{i'\} = \hat{\mathcal{D}}^{s,s'},\;\; i \in \mathcal{I} \tag{3.25}$$

$$-y_{t+1}^{UB}(1 - z_t^{s,s'}) \leq y_{t+1}^s - y_{t+1}^{s'} \leq y_{t+1}^{UB}(1 - z_t^{s,s'}) \quad \forall\,(t,s,s') \in \mathcal{SP}_N,\;\; t \in \mathcal{T}_C^{i'},\;\; t < T,\;\; \{i'\} = \hat{\mathcal{D}}^{s,s'} \tag{3.26}$$

$$z_t^{s,s'} \in \{0,1\} \quad \forall\,(t,s,s') \in \mathcal{SP}_N,\;\; t \in \mathcal{T}_C^{i'},\;\; \{i'\} = \hat{\mathcal{D}}^{s,s'} \tag{3.27}$$

Note that $z_t^{s,s'}$, defined in Equation (3.27), is the binary equivalent of Boolean variable $Z_t^{s,s'}$; i.e., $(z_t^{s,s'} = 1) \Leftrightarrow (Z_t^{s,s'} = True)$ and $(z_t^{s,s'} = 0) \Leftrightarrow (Z_t^{s,s'} = False)$. The fixed endogenous NACs can be viewed as a special case of these constraints with $z_t^{s,s'} = 1$. Specifically, in the initial time periods, each double-sided inequality collapses into a single equality constraint, thereby providing

[1] We substitute variable upper bounds for big-M parameters; however, for a specific problem instance, tighter bounds can often be established.

us with a smaller, tighter formulation (Colvin and Maravelias, 2010).

The value of $Z_t^{s,s'}$ is determined by an uncertainty-resolution rule, as stated in general form in Equation (3.19) (Tarhan et al., 2013; Gupta and Grossmann, 2014a). This rule uses the values of all decisions $b_{i',\tau}^s$ up to and including the current time period to determine whether uncertainty has been resolved in a given source $i' \in \mathcal{I}$.

$$Z_t^{s,s'} \Leftrightarrow \left[\bigwedge_{\tau=1}^{t} (\neg b_{i',\tau}^s)\right] \qquad \forall\,(t,s,s') \in \mathcal{SP}_N,\;\; t \in \mathcal{T}_C^{i'},\;\; \{i'\} = \hat{\mathcal{D}}^{s,s'} \tag{3.28}$$

Prior to this time t, $Z_\tau^{s,s'} = True$ and the scenarios s and s' that differ in the possible realization of $\theta_{i',h}$ are indistinguishable. After the investment at time t, $Z_t^{s,s'} = False$ and the scenarios are distinguishable; thus, non-anticipativity constraints no longer apply between s and s'. Note that we previously used this concept in the discussion of Figure 1.4. Logic constraints (3.28) can be rewritten as linear integer inequality constraints (3.29) and (3.30) by applying the reformulations described in Williams (2013) and Raman and Grossmann (1991).

$$1 - \sum_{\tau=1}^{t} b_{i',\tau}^s \le z_t^{s,s'} \qquad \forall\,(t,s,s') \in \mathcal{SP}_N,\;\; t \in \mathcal{T}_C^{i'},\;\; \{i'\} = \hat{\mathcal{D}}^{s,s'} \tag{3.29}$$

$$z_t^{s,s'} \le 1 - b_{i',\tau}^s \qquad \forall\,(t,s,s') \in \mathcal{SP}_N,\;\; t \in \mathcal{T}_C^{i'},\;\; \tau \in \mathcal{T},\;\; \tau \le t,\;\; \{i'\} = \hat{\mathcal{D}}^{s,s'} \tag{3.30}$$

Making this replacement and substituting disjunctive constraints (3.18) with constraints (3.24)–(3.27) transforms model ($\mathrm{MSSP_N}$) into an MILP. Bounds and integrality restrictions on the variables are given in Equations (3.20), (3.21) and (3.27).

3.3 MSSP Formulation for Endogenous and Exogenous Uncertainties

In the case of both endogenous and exogenous uncertainties, the multistage stochastic programming formulation of model (MPD) is given by model (MSSP). This model is also adapted from the work of Goel and Grossmann (2006) and will be our primary focus for the remainder of this thesis. Just as the scenario tree for this class of problems is represented by a composite scenario tree (Figure 2.1c), the corresponding model can also be seen as a composite of the exogenous model ($\mathrm{MSSP_X}$) and the endogenous model ($\mathrm{MSSP_N}$). In particular, all of their respective NACs and logic constraints appear together in (MSSP).

(MSSP)

$$\min_{b,y,x} \phi = \sum_{s\in\mathcal{S}} p^s \sum_{t\in\mathcal{T}} \left({}^y c_t^s y_t^s + {}^x c_t^s x_t^s + {}^w c_t^s w_t^s + \sum_{i\in\mathcal{I}} {}^b c_{i,t}^s b_{i,t}^s \right) \tag{3.31}$$

$$\text{s.t.} \quad \sum_{\tau=1}^{t} \left({}^y A_{\tau,t}^s y_\tau^s + {}^x A_{\tau,t}^s x_\tau^s + {}^w A_{\tau,t}^s w_\tau^s + \sum_{i\in\mathcal{I}} {}^b A_{i,\tau,t}^s b_{i,\tau}^s \right) \le a_t^s \qquad \forall\, t \in \mathcal{T},\;\; s \in \mathcal{S} \tag{3.32}$$

$$b_{i,1}^s = b_{i,1}^{s'} \qquad \forall\,(s,s') \in \mathcal{SP}_F,\;\; i \in \mathcal{I} \tag{3.14}$$

$$y_1^s = y_1^{s'} \qquad \forall\, (s,s') \in \mathcal{SP}_F \tag{3.6}$$

$$x_t^s = x_t^{s'} \qquad \forall\, (t,s,s') \in \mathcal{SP}_X \tag{3.7}$$

$$b_{i,t+1}^s = b_{i,t+1}^{s'} \qquad \forall\, (t,s,s') \in \mathcal{SP}_X,\;\; i \in \mathcal{I} \tag{3.33}$$

$$y_{t+1}^s = y_{t+1}^{s'} \qquad \forall\, (t,s,s') \in \mathcal{SP}_X \tag{3.8}$$

$$x_t^s = x_t^{s'} \qquad \forall\, (t,s,s') \in \mathcal{SP}_N,\;\; t \in \mathcal{T}_E^{i'},\;\; \{i'\} = \hat{\mathcal{D}}^{s,s'} \tag{3.15}$$

$$b_{i,t+1}^s = b_{i,t+1}^{s'} \qquad \forall\, (t,s,s') \in \mathcal{SP}_N,\;\; t \in \mathcal{T}_E^{i'},\;\; \{i'\} = \hat{\mathcal{D}}^{s,s'},\;\; i \in \mathcal{I} \tag{3.16}$$

$$y_{t+1}^s = y_{t+1}^{s'} \qquad \forall\, (t,s,s') \in \mathcal{SP}_N,\;\; t \in \mathcal{T}_E^{i'},\;\; \{i'\} = \hat{\mathcal{D}}^{s,s'} \tag{3.17}$$

$$\begin{bmatrix} \quad Z_t^{s,s'} & \\ x_t^s = x_t^{s'} & \\ b_{i,t+1}^s = b_{i,t+1}^{s'} \; \forall\, i \in \mathcal{I}, & t < T \\ y_{t+1}^s = y_{t+1}^{s'} & t < T \end{bmatrix} \vee \left[\neg Z_t^{s,s'}\right] \qquad \forall\, (t,s,s') \in \mathcal{SP}_N,\;\; t \in \mathcal{T}_C^{i'},\;\; \{i'\} = \hat{\mathcal{D}}^{s,s'} \tag{3.18}$$

$$Z_t^{s,s'} \Leftrightarrow F(b_{i',1}^s, b_{i',2}^s, \ldots, b_{i',t}^s) \qquad \forall\, (t,s,s') \in \mathcal{SP}_N,\;\; t \in \mathcal{T}_C^{i'},\;\; \{i'\} = \hat{\mathcal{D}}^{s,s'} \tag{3.19}$$

$$b_{i,t}^s \in \{0,1\},\;\; y_t^s \in \mathcal{Y}_t^s,\;\; x_t^s \in \mathcal{X}_t^s,\;\; w_t^s \in \mathcal{W}_t^s \qquad \forall\, i \in \mathcal{I},\;\; t \in \mathcal{T},\;\; s \in \mathcal{S} \tag{3.34}$$

$$Z_t^{s,s'} \in \{True, False\} \qquad \forall\, (t,s,s') \in \mathcal{SP}_N,\;\; t \in \mathcal{T}_C^{i'},\;\; \{i'\} = \hat{\mathcal{D}}^{s,s'} \tag{3.21}$$

Like model ($\mathrm{MSSP_N}$), this formulation represents a *hybrid* mixed-integer linear *disjunctive* programming problem due to the presence of the conditional endogenous constraints (3.18) and logic constraints (3.19). The disjunctive constraints can be replaced by constraints (3.24)–(3.27), and if immediate resolution of uncertainty is assumed, the logic constraints can be replaced by inequalities (3.29) and (3.30). These steps transform model (MSSP) into standard mixed-integer linear form.

Notice that the objective function (3.31) and constraints (3.32) have only been updated from their respective counterparts in model ($\mathrm{MSSP_N}$) to reflect the fact that the set of scenarios is now given by $\mathcal{S}$. This is also true for the bounds and integrality restrictions specified in Equation (3.34). The only new addition to the model is Equation (3.33), which gives the exogenous non-anticipativity constraints for $b_{i,t}^s$, as these variables were not originally defined in the exogenous model ($\mathrm{MSSP_X}$). Exogenous parameters $\xi_{j,t}$ and endogenous parameters $\theta_{i,h}$ may enter the model through the objective function and/or the constraints.

It is interesting to note that if we assume expected values for the endogenous parameters, then we have $\mathcal{SP}_N = \emptyset,\; \mathcal{S} = \mathcal{S}_X$, and model (MSSP) reduces to the exogenous model ($\mathrm{MSSP_X}$). Similarly, if we assume expected values for the exogenous parameters, then we have $\mathcal{SP}_X = \emptyset,\; \mathcal{S} = \mathcal{S}_N$, and model (MSSP) reduces to the endogenous model ($\mathrm{MSSP_N}$).

The multistage stochastic programming problem (MSSP) may appear to be simply a larger version of the purely-exogenous and purely-endogenous formulations previously discussed; however, there is a great deal of complexity contained in scenario-pair sets $\mathcal{SP}_F$, $\mathcal{SP}_X$, and $\mathcal{SP}_N$. Specifically, we must carefully account for the presence of both types of uncertainty when defining these sets. Notice that in the exogenous formulation ($\mathrm{MSSP_X}$), NACs are applied in time period t for all pairs of scenarios that are indistinguishable in terms of the resolution of exogenous uncertainty. In the endogenous formulation ($\mathrm{MSSP_N}$), NACs are applied in time period t for all pairs of scenarios that must be indistinguishable, and conditionally applied for those that *may* be indistinguishable, in terms of the resolution of endogenous uncertainty. As can be seen in the composite scenario tree (Figure 2.1c), this is not the case when endogenous and exogenous uncertainties are both present. First-period NACs link all scenarios at the beginning of the first time period, as always, but exogenous NACs now link scenarios in time period t that are indistinguishable in terms of the resolution of exogenous uncertainty *and* are identical in all possible realizations of the endogenous parameters. In other words, exogenous NACs are applied between scenarios *within each subtree.* Endogenous NACs now link scenarios in time period t that differ in the possible realization of one endogenous parameter *and* are identical in all realizations of the exogenous parameters. Thus, endogenous NACs are applied between scenarios *in different subtrees.* This is an interesting modeling challenge and will be discussed in detail in the next chapter.

Chapter 4

Scenario Pairs and Reduction Properties

In defining each of the scenario-pair sets $\mathcal{SP}_F$, $\mathcal{SP}_X$, and $\mathcal{SP}_N$, we begin with a naïve approach in which we specify only $s, s' \in \mathcal{S}$ and $s \neq s'$, along with the additional indistinguishability conditions specific to either first-period NACs, exogenous NACs, or endogenous NACs; i.e.,

$$\{(s, s') : \; s, s' \in \mathcal{S}, \;\; s \neq s', \;\; \text{conditions for indistinguishability}\} \tag{4.1}$$

As stated in the following property, however, the condition $s \neq s'$ is not particularly restrictive and leaves us with many redundant scenario pairs.

Property 1. *Scenario pairs (s, s') and (s', s) refer to the same pair. Thus, it is sufficient to enforce non-anticipativity constraints for only pairs (s, s') where $s < s'$* (Goel and Grossmann, 2006)*.*

Proof. See Appendix B.1. A brief, qualitative proof can also be found in Goel and Grossmann (2006). □

This simple symmetry argument eliminates half of the scenario pairs generated by Equation (4.1). We place special emphasis on *reduction properties* such as Property 1 since NACs are expressed for each pair of scenarios, and the number of pairs can be extremely large in instances with a large number of scenarios. In the following sections, we will define additional reduction properties to exclude all redundant pairs from each of our set definitions. We begin with scenario-pair set $\mathcal{SP}_F$ for first-period NACs.

4.1 First-period Scenario Pairs

As was the case for purely-exogenous and purely-endogenous uncertainties, at the beginning of the first time period, no decisions have been implemented and no uncertainties have been resolved. Hence, all scenarios are indistinguishable at that time and we must make the same here-and-now decisions in all scenarios. To define the set of scenario pairs required for these non-anticipativity constraints, we rely on the following property.

Property 2. *For first-period NACs, it is sufficient to consider only scenario pairs* (s, s') *for which* s *and* s' *are adjacent.*

Proof. See Appendix B.2. $\square$

Accordingly, we define the set of all pairs of adjacent scenarios, $\mathcal{A}$:

$$\mathcal{A} := \{(s, s') : s, s' \in \mathcal{S}, \;\; s' = s + 1, \;\; s < S\} \tag{4.2}$$

Note that the condition $s < s'$ is implicit in this definition since we are only considering consecutive scenarios in the 'forward' direction. The set of all scenario pairs for first-period NACs is then simply equal to set $\mathcal{A}$, as we define in Equation (3.10).

$$\mathcal{SP}_F := \mathcal{A} \tag{3.10}$$

This is the minimum number of scenario pairs, as stated in the following proposition.

Proposition 1. *First-period scenario-pair set* $\mathcal{SP}_F$ *contains the minimum number of scenario pairs.*

Proof. See Appendix B.3. $\square$

Note that the respective scenario pairs in set $\mathcal{SP}_F$ are *non-unique*. In other words, different formulations with the same cardinality are possible; e.g., we may instead choose to link the first scenario to every other scenario. Such alternative pairing approaches have been shown to perform better in Lagrangean decomposition (Oliveira et al., 2013); however, for convenience, we limit our current discussion to the consecutive-pairing approach.

4.2 Exogenous Scenario Pairs

Excluding the beginning of the first time period, scenarios s and s' are indistinguishable in time period t if they are identical in the realizations of all exogenous parameters up to this point *and* they have all of the same possible realizations for the endogenous parameters. These scenario pairs are required for exogenous non-anticipativity constraints.

Rather than explicitly checking that each pair of scenarios has the same possible endogenous realizations, it is clear from Figure 2.1 that due to the manner in which we generate the scenario set, this condition is implicitly satisfied for any s and s' in the same subtree. Recall that this is because each subtree represents an exogenous scenario tree, and by definition, all scenarios in this tree must have the same endogenous realizations (see Section 2.3). Furthermore, different subtrees have different possible endogenous realizations, so s and s' can *only* be in the same subtree. This argument also allows us to invoke the following reduction property.

Property 2b. *For exogenous NACs, it is sufficient to consider only scenario pairs* (s, s') *for which* s *and* s' *are adjacent.*

Proof. See Appendix B.4. $\square$

Hence, we state that adjacent scenarios s and s' will be indistinguishable in the first time period if they have the same realizations for all exogenous parameters in this period and they are in the same subtree. Let Boolean parameter $Q_t^{s,s'}$ represent the indistinguishability of adjacent scenarios s and s' in time period t, where $Q_t^{s,s'} = True$ if the scenarios are indistinguishable, and $Q_t^{s,s'} = False$ otherwise. Then,

$$Q_1^{s,s'} := \begin{cases} True, & \text{if } \xi_{j,1}^{s} = \xi_{j,1}^{s'} \ \ \forall\, j \in \mathcal{J} \\ False, & \text{otherwise} \end{cases} \qquad \forall\, (s, s') \in \mathcal{A},\ \ Sub(s) = Sub(s') \tag{4.3}$$

where the subtree condition $Sub(s) = Sub(s')$ relies on the definition provided by Equation (2.7).

For all subsequent time periods, the scenarios are indistinguishable if they were indistinguishable in the previous time period, they have the same exogenous realizations in the current time period, and they are in the same subtree:

$$Q_t^{s,s'} := \begin{cases} True, & \text{if } Q_{t-1}^{s,s'} = True \ \text{ and } \ \xi_{j,t}^{s} = \xi_{j,t}^{s'} \ \ \forall\, j \in \mathcal{J} \\ False, & \text{otherwise} \end{cases}$$
$$t = 2, 3, \ldots, T,\ \ \forall\, (s, s') \in \mathcal{A},\ \ Sub(s) = Sub(s') \tag{4.4}$$

As an example, scenarios 1 and 2 in Figure 2.1c have the same realizations for the exogenous parameter in the first time period; i.e., $\xi_1^1 = \xi_1^2$. Thus, these scenarios are indistinguishable at the end of this period and $Q_1^{1,2} = True$. They have different realizations in the second time period (i.e., $\xi_2^1 \neq \xi_2^2$), so the scenarios are distinguishable at that time and $Q_2^{1,2} = False$. Since the leaf nodes in each subtree refer to independent states, it is in fact the case that all adjacent scenarios in the same subtree will be distinguishable by the end of the final time period; i.e., $Q_T^{s,s'} = False$. Thus, it is unnecessary to evaluate Equation (4.4) for $t = T$. We also note that because $Q_t^{s,s'}$ is the same for all subtrees, it is most efficient to calculate $Q_t^{s,s'}$ only for the first subtree and then to duplicate the results for all others.

The set of all scenario pairs (s, s') in each time period t, such that s and s' are indistinguishable in terms of the resolution of exogenous uncertainty and are identical in all possible realizations of the endogenous parameters, can then be defined as:

$$\mathcal{SP}_X := \{(t, s, s') :\ t \in \mathcal{T} \setminus \{T\},\ \ (s, s') \in \mathcal{A},\ \ Sub(s) = Sub(s'),\ \ Q_t^{s,s'} = True\} \tag{3.11}$$

Equation (3.11) is also applicable in purely-exogenous problems since, in that case, $Sub(s) = 1$ for all $s \in \mathcal{S}$. This is the reasoning behind the use of set $\mathcal{SP}_X$ in model ($\mathrm{MSSP_X}$).

We now define exogenous scenario 'groups' in each time period $t \in \mathcal{T} \setminus \{T\}$, where each group is a set of indistinguishable scenarios that refer to the same state.[1] Specifically, each group is the

[1]We will frequently refer to exogenous scenario groups *in* time period t. This will be understood to mean the *end* of time period t, after all realizations in that period have occurred.

direct result of splitting a single node into indistinguishable copies for each scenario, as discussed in the proof of Property 2b. For example, at the end of the first time period in Figure 2.1c, scenarios 1 and 2 refer to the same unique state and can be grouped together. Scenarios 3 and 4 refer to another unique state and can be placed into a second group. Continuing this process, we end up with 8 different groups of two scenarios each, as shown in Figure 4.1. Blue groups consist of scenarios with a *low* realization for exogenous parameter ξ_1, and green groups consist of scenarios with a *high* realization for that parameter. We typically do not define scenario groups for the final time period, since (as previously mentioned) adjacent leaf nodes in the same subtree are unique; in other words, there would be S groups of one scenario each in time period T (e.g., 16 groups of one scenario each in Figure 4.1).

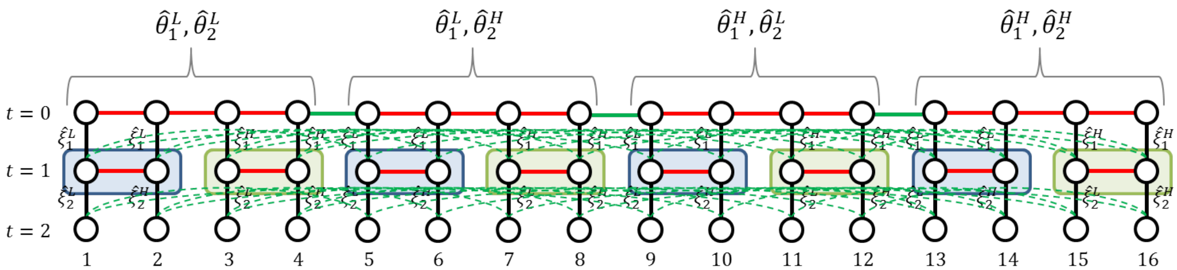


Figure 4.1: Exogenous scenario groups.

To generalize this grouping process, we first define parameter $G_X(t,s)$ to return the group number of each scenario $s \in \mathcal{S}$ in time period $t \in \mathcal{T} \setminus \{T\}$. Next, we assign the first scenario in each of these time periods to group 1 by specifying $G_X(t,1) := 1 \ \forall\, t \in \mathcal{T} \setminus \{T\}$. We then use Equation (4.5) to assign group numbers to all other scenarios:

$$G_X(t,s) := G_X(t,s-1) + \sum_{(t,s-1,s) \notin \mathcal{SP}_X} [1] \qquad \forall\, t \in \mathcal{T} \setminus \{T\}, \;\; s = 2,3,\ldots,S \tag{4.5}$$

The general idea behind this equation is that the group number of scenario s will be equal to the group number of the previous scenario $s-1$, given by $G_X(t,s-1)$, as long as these two adjacent scenarios are indistinguishable based on the definition of set $\mathcal{SP}_X$. If they are not indistinguishable in this sense (i.e., $(t,s-1,s) \notin \mathcal{SP}_X$), then the scenarios have different realizations for some of the uncertain parameters and scenario s belongs in a new group; thus, the group number is incremented by 1. For instance, at $t=1$ in Figure 4.1, scenario 1 is first assigned a group number of 1. Scenario 2 is indistinguishable from scenario 1, so must also be assigned to group 1. Scenario 3, however, is distinguishable from scenario 2 since $\xi_1^2 \neq \xi_1^3$, and $(1,2,3) \notin \mathcal{SP}_X$. Thus, we increment the group number and assign scenario 3 to group 2. We repeat this process for all remaining scenarios in this time period.

We index these groups by defining the set of indices $\mathcal{K}_t$,

$$\mathcal{K}_t := \{k : \; k = 1,2,\ldots,G_X(t,S)\} \qquad \forall\, t \in \mathcal{T} \setminus \{T\} \tag{4.6}$$

where $G_X(t,S)$ gives the total number of groups in time period t (since it is the group number for the final scenario in time period t). In Figure 4.1, this corresponds to $G_X(1,16) = 8$; therefore, $\mathcal{K}_1 := \{1, 2, \ldots, 8\}$.

We then use the group numbers to define the set of scenarios for each group:

$$ {}^X\mathcal{G}_t^k := \{s : \; s \in \mathcal{S}, \;\; G_X(t,s) = k\} \qquad \forall \, k \in \mathcal{K}_t, \;\; t \in \mathcal{T} \setminus \{T\} \tag{4.7} $$

For example, at $t = 1$ in Figure 4.1, scenario 1 has a group number of 1 (i.e., $G_X(1,1) = 1$) and scenario 2 has a group number of 1 (i.e., $G_X(1,2) = 1$). Accordingly, exogenous scenario group 1 in the first time period is given by ${}^X\mathcal{G}_1^l = \{1, 2\}$. We similarly define ${}^X\mathcal{G}_1^2 = \{3, 4\}$, ${}^X\mathcal{G}_1^3 = \{5, 6\}, \ldots, {}^X\mathcal{G}_1^8 = \{15, 16\}$.

The exogenous scenario-group definitions allow us to state the following proposition.

Proposition 2. *Exogenous scenario-pair set $\mathcal{SP}_X$ contains the minimum number of scenario pairs.*

Proof. See Appendix B.5. □

Like set $\mathcal{SP}_F$, the respective scenario pairs in set $\mathcal{SP}_X$ are non-unique. The concept of exogenous scenario groups will be used again in the next section to derive endogenous scenario-pair set $\mathcal{SP}_N$. As will be shown, the definition of this set is quite complex.

4.3 Endogenous Scenario Pairs

Excluding the beginning of the first time period, scenarios s and s' are indistinguishable in the initial time periods $t \in \mathcal{T}_E^{i'}$ if they differ in the possible realizations of one or more endogenous parameters *and* they are identical in the realizations of all exogenous parameters that have been realized up until that time. Recall that these scenarios must be indistinguishable here because the endogenous uncertainty cannot yet be resolved. These scenario pairs are used to generate fixed endogenous NACs.

For the remaining time periods $t \in \mathcal{T}_C^{i'}$, the uncertainty *can* be resolved at some point, but we do not know when this will occur (or if it will at all). Scenarios s and s' will be indistinguishable until this unknown point in time. Thus, we state that under the same conditions given for $t \in \mathcal{T}_E^{i'}$, scenarios s and s' in $t \in \mathcal{T}_C^{i'}$ *may* be indistinguishable. These scenario pairs are used to generate conditional endogenous NACs. Notice that due to the conditional nature of these constraints, set $\mathcal{SP}_N$ may contain several scenario pairs that we do not need. This is in sharp contrast to the exogenous scenario-pair set $\mathcal{SP}_X$, where every scenario pair is required because all of the NACs are fixed.

Before we derive the endogenous scenario-pair set $\mathcal{SP}_N$, it is possible to significantly strengthen the indistinguishability requirements. We begin with the following reduction property.

Property 3. *For endogenous NACs, it is sufficient to consider only scenario pairs (s, s') for which s and s' differ in the possible realization of a single endogenous parameter and are identical in the realizations of all exogenous parameters in all time periods.*

Proof. See Goel and Grossmann (2006). □

Example. Due to the complexity of Property 3, and its importance to this work, we provide an illustrative example of the proof. Consider Figure 4.2, where we have isolated scenarios 1, 5, and 13 from Figure 2.1c for an arbitrary time period $t = \tau$. Scenario 1 differs from scenario 5 in the possible realization of endogenous parameter θ_2. Scenario 5 differs from scenario 13 in the possible realization of endogenous parameter θ_1. Scenario 1 differs from scenario 13, however, in the possible realizations of *both* θ_1 *and* θ_2. The three scenarios have identical realizations for exogenous parameter ξ_t in all time periods.

Disregarding the possibility of initial 'equality' periods, we will have three conditional links between the scenarios, as shown at the top of Figure 4.2: $(1, 5)$ and $(5, 13)$, as shown in green, and $(1, 13)$, as shown in orange. There are four possible outcomes depending upon the way the uncertainty is resolved. In Case 1, both endogenous parameters have been realized by the end of this time period. Accordingly, the scenarios are distinguishable and non-anticipativity does not apply. In Case 2, only the value of θ_1 has been realized and NACs are enforced between scenarios 1 and 5. If we consider only variables y_τ^s, the corresponding NAC is $y_\tau^1 = y_\tau^5$. Similarly, in Case 3, only the value of θ_2 has been realized and NACs are enforced between scenarios 5 and 13; e.g., $y_\tau^5 = y_\tau^{13}$. When neither of the parameters has been realized in Case 4, all three conditional links are enforced: $y_\tau^1 = y_\tau^5$, $y_\tau^5 = y_\tau^{13}$, and $y_\tau^1 = y_\tau^{13}$.

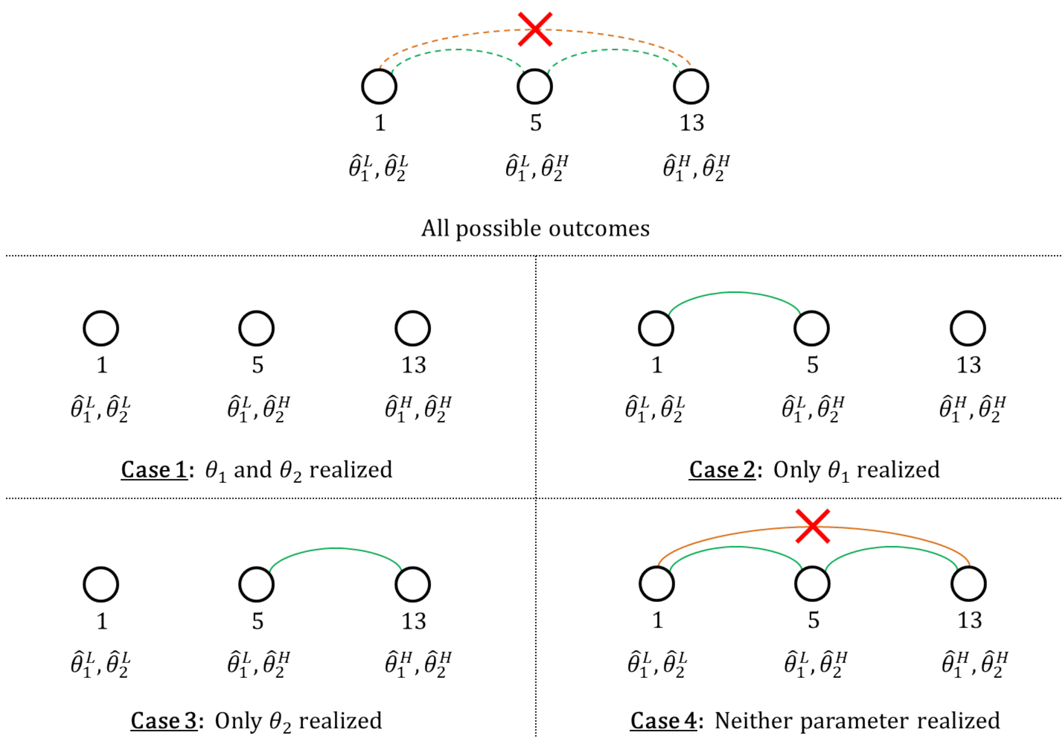


Figure 4.2: Illustration of Property 3.

Notice that Case 4 is the *only* case in which we apply non-anticipativity for scenario pair $(1, 13)$, and it applies only at the same time as the non-anticipativity for pairs $(1, 5)$ and $(5, 13)$. Thus, by a simple transitivity argument, it is clear that constraint $y_\tau^1 = y_\tau^{13}$ is implied by constraints $y_\tau^1 = y_\tau^5$

and $y_\tau^5 = y_\tau^{13}$. Accordingly, scenario pair $(1, 13)$ can be excluded entirely. This leaves us with two pairs that differ only in the possible realization of a single endogenous parameter.

Recall that scenarios 1, 5, and 13 have identical realizations for the exogenous parameter in all time periods. We now extend this example to include scenario 2, which has a different exogenous realization in the second period. Here, we illustrate the second part of Property 3; specifically, that the corresponding scenario pairs (s, s') consist of scenarios s and s' that are identical in the realizations of all exogenous parameters *in all time periods*, rather than just identical in the exogenous realizations that have been *revealed up until the current time period*. This is shown in Figure 4.3.

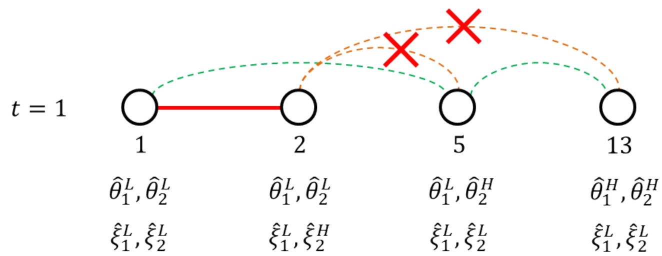


Figure 4.3: Property 3 as applied to endogenous and exogenous uncertainties.

Figure 4.3 includes additional scenario pairs $(2, 5)$ and $(2, 13)$, as shown in orange. Notice, however, that scenario 2 is identical to scenario 1 aside from the different exogenous realization in the second time period (i.e., $\xi_2^1 \neq \xi_2^2$). This means that in the first period, non-anticipativity for pairs $(2, 5)$ and $(1, 5)$ will apply at the same time, non-anticipativity for pairs $(2, 13)$ and $(1, 13)$ will apply at the same time,[2] and we have the exogenous non-anticipativity constraint $y_\tau^1 = y_\tau^2$ between scenarios 1 and 2 (shown in red).

Thus, by transitivity, pair $(2, 5)$ can be eliminated since constraints $y_\tau^1 = y_\tau^2$ and $y_\tau^1 = y_\tau^5$ imply $y_\tau^2 = y_\tau^5$. Likewise, pair $(2, 13)$ can also be eliminated since constraints $y_\tau^1 = y_\tau^2$, $y_\tau^1 = y_\tau^5$, and $y_\tau^5 = y_\tau^{13}$ imply $y_\tau^2 = y_\tau^{13}$. Notice that we have eliminated any endogenous scenario pairs (s, s') for which s and s' are not identical in the realizations of all exogenous parameters in all time periods.

As proved rigorously in Goel and Grossmann (2006), Property 3 always holds, provided that the set of scenarios consists of all possible combinations of realizations of the endogenous parameters. The authors also showed that this property extends to the general case where there are multiple parameters associated with each source of endogenous uncertainty (as we have considered throughout this thesis with the use of parameter $\theta_{i,h}$).

By Property 3, we may now state that scenarios s and s' are indistinguishable in time period t if they differ in the possible realization of *exactly one* endogenous parameter and they are identical in the realizations of all exogenous parameters *in all time periods*. We first address the latter part of this statement.

Recall from the previous section that scenarios in the same subtree must have the same endogenous realizations. Thus, for s and s' to differ in *any* endogenous realizations, they must belong

[2] Recall from the discussion surrounding Figure 4.2 that scenario pair $(1, 13)$ is implied by pairs $(1, 5)$ and $(5, 13)$.

to different subtrees. Furthermore, for these scenarios to have exactly the same exogenous realizations, they must have the same *position* in both subtrees; for example, the first scenario in both, as in scenarios 1 and 5. This is because we generate the composite tree by starting with a single exogenous tree that has no duplicate scenarios. It follows that when we duplicate the exogenous tree for each possible combination of realizations of the endogenous parameters, scenarios in the same position in different subtrees have originated from the same scenario. Therefore, they must have all of the same exogenous realizations. Because there were no duplicates in the original exogenous tree, these are the only scenarios for which this holds.

We define parameter $Pos(s)$ to return the position of scenario s from the viewpoint of its respective subtree; in other words, the index that scenario s would have if it were in subtree 1:

$$Pos(s) := s - S_X(Sub(s) - 1) \qquad \forall\, s \in \mathcal{S} \tag{4.8}$$

Equation (4.8) calculates this normalized scenario index for s by subtracting off the appropriate number of scenarios according to the subtree that s belongs to. Recall that S_X is just the number of scenarios in each subtree. As a simple example, consider scenarios 1, 5, 9, and 13 in Figure 2.1c. Since these scenarios refer to the first scenario in each subtree, respectively, Equation (4.8) gives $Pos(1) = 1 - 4(1-1) = 1$, $Pos(5) = 5 - 4(2-1) = 1$, $Pos(9) = 9 - 4(3-1) = 1$, and $Pos(13) = 13 - 4(4-1) = 1$.

Thus, to indicate that s and s' are identical in all exogenous realizations, but differ in at least one possible endogenous realization, it is sufficient to state $Pos(s) = Pos(s')$, with $s < s'$. Note that this implies that the two scenarios are in different subtrees, so it is unnecessary to specify $Sub(s) \neq Sub(s')$.

We now address the first part of Property 3; namely, that scenarios s and s' differ in the possible realization of *exactly one* endogenous parameter. To do so, we define sets $\mathcal{D}^{s,s'}$, composed of pairs of indices (i', h), to indicate the endogenous parameters $\theta_{i',h}$ for which scenarios s and s' differ in possible realizations:[3]

$$\mathcal{D}^{s,s'} := \left\{ (i', h) : \, i' \in \mathcal{I}, \;\; h \in \mathcal{H}_{i'}, \;\; \theta^{s}_{i',h} \neq \theta^{s'}_{i',h} \right\} \qquad \forall\, s, s' \in \mathcal{S}, \;\; s < s', \;\; Pos(s) = Pos(s') \tag{4.9}$$

Property 3 then requires that $|\mathcal{D}^{s,s'}| = 1$ for all endogenous scenario pairs. In other words, the corresponding set of pairs for all time periods is given by:

$$\mathcal{SP}_{N^3} := \left\{ (t, s, s') : \, t \in \mathcal{T}, \;\; s, s' \in \mathcal{S}, \;\; s < s', \;\; Pos(s) = Pos(s'), \;\; |\mathcal{D}^{s,s'}| = 1 \right\} \tag{4.10}$$

Note that the same pairs are present in each period.

As pointed out by Gupta and Grossmann (2011), however, when we consider 3 or more possible realizations for any of the endogenous parameters, there are additional redundant scenario pairs that are not removed by this property. This is illustrated in Figure 4.4. Here we consider a group

[3] Recall that we index the sources with i' so as not to conflict with index i of $b^{s}_{i,t}$ in models (MSSP$_{\mathrm{N}}$) and (MSSP).

of three scenarios ($\hat{s}, \hat{s}'$, and $\hat{s}''$), in an arbitrary time period $t = \tau$, that all differ in the possible realization of a single endogenous parameter $\theta_{\hat{\imath},\hat{h}}$. These scenarios will be distinguishable in time period τ if parameter $\theta_{\hat{\imath},\hat{h}}$ has been realized (Case 1), or indistinguishable if the parameter has not yet been realized (Case 2).

Using Property 3, we generate three scenario pairs: $(\hat{s}, \hat{s}')$ and $(\hat{s}', \hat{s}'')$, as shown in green, and $(\hat{s}, \hat{s}'')$, as shown in orange. Since the corresponding NACs all apply at the same time or are all ignored at the same time, it is clear that scenario pair $(\hat{s}, \hat{s}'')$ is redundant and can be eliminated. This follows directly from the simple transitivity arguments previously used in the example of Property 3. Because $|\mathcal{D}^{\hat{s},\hat{s}''}| = 1$ and yet $(\hat{s}, \hat{s}'')$ is redundant, it is also clear that we must rely on an alternative approach to exclude such scenario pairs.

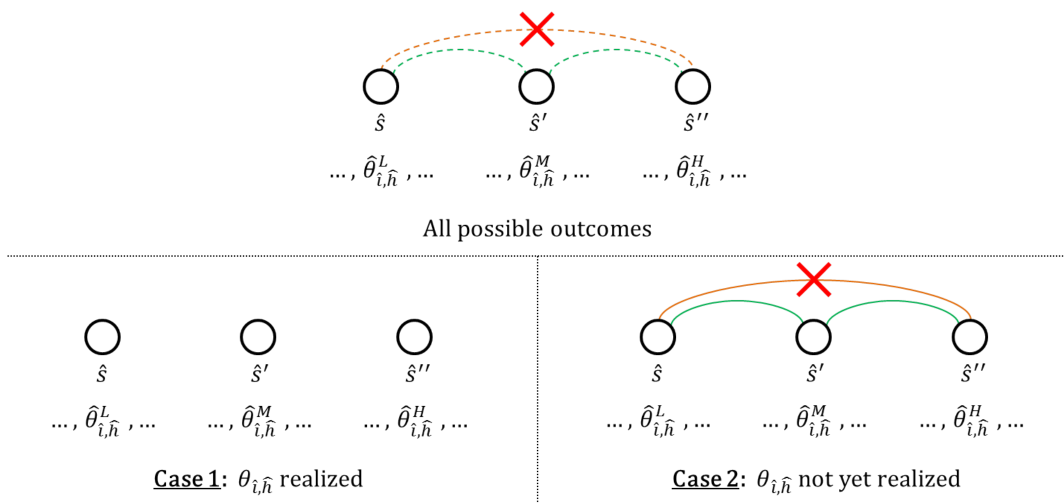


Figure 4.4: Property 3 fails to eliminate all redundant scenario pairs when there are 3 or more possible realizations for any of the endogenous parameters.

A simple remedy for this, as proposed by Gupta and Grossmann (2011), is to first generate all 'groups' of scenarios like that shown in Figure 4.4, and then link consecutive scenarios in each of these groups.[4] Each group is the set of all scenarios that differ only in the possible realization of a single endogenous parameter h of source i' (i.e., $\theta_{i',h}$). As previously noted, these scenarios will be indistinguishable as long as this parameter is unrealized.

Thus, for each $i' \in \mathcal{I}$ and $h \in \mathcal{H}_{i'}$, we define parameter $G_N(i', h, s)$ to identify the index of the group that scenario $s \in \mathcal{S}$ belongs to. We refer to this as the group number, represented here by index l, and propose the following algorithm to assign group numbers to all scenarios. Notice that unlike the exogenous case, an algorithm is required here because the groups consist of nonconsecutively-indexed scenarios.

Endogenous Scenario-Group Algorithm

Step 1 Initialize the group numbers to zero for all scenarios; i.e., $G_N(i', h, s) := 0 \ \forall \ i' \in \mathcal{I}, \ h \in$

[4] The scope of Gupta and Grossmann (2011) is limited to purely-endogenous MSSP problems with no initial 'equality' time periods and only one parameter associated with each source of uncertainty.

$\mathcal{H}_{i'}$, $s \in \mathcal{S}$. Also, define a group counter, GroupCount, to keep track of the current group number in each iteration.

Step 2 For each endogenous parameter, define all groups of scenarios that differ in the possible realization of *only* this parameter. This is done as follows.

For each $i' \in \mathcal{I}$ and $h \in \mathcal{H}_{i'}$:

Step 2a Reset the group counter (i.e., GroupCount := 0).

Step 2b Fix s to the next available scenario in $\mathcal{S}$ (i.e., s has not already been assigned to a group, so $G_N(i', h, s) = 0$), and then search for all other scenarios from which s differs in the possible realization of only $\theta_{i',h}$. Such scenarios must be in the same group as s.

Specifically, for $s = 1, 2, \ldots, S$, where $G_N(i', h, s) = 0$:

(i) Increment the group counter (i.e., GroupCount := GroupCount + 1).

(ii) Set the group number of scenario s to the current group number:

$$G_N(i', h, s) := \text{GroupCount} \tag{4.11}$$

(iii) Search for scenarios $s' \in \mathcal{S}$ that differ from s in the possible realization of the same endogenous parameter; i.e., $\mathcal{D}^{s,s'} = \{(i', h)\}$. For each s' that satisfies this condition, set the group number of that scenario to the same group number as scenario s; i.e.,

$$G_N(i', h, s') := G_N(i', h, s) \qquad \forall\, s' \in \mathcal{S},\;\; s' > s,\;\; Pos(s') = Pos(s),\;\; \mathcal{D}^{s,s'} = \{(i', h)\} \tag{4.12}$$

For instance, in Figure 4.4, assume that scenario $\hat{s}$ is in group $\hat{l}$ corresponding to endogenous parameter $\theta_{\hat{\imath},\hat{h}}$. Also, assume that $s = \hat{s}$. In this step of the algorithm, we would first identify $\hat{s}'$ as belonging to the same group as $\hat{s}$, and then the same for $\hat{s}''$, since $\mathcal{D}^{\hat{s},\hat{s}'} = \mathcal{D}^{\hat{s},\hat{s}''} = \{(\hat{\imath}, \hat{h})\}$. Thus, we would have $G_N(\hat{\imath}, \hat{h}, \hat{s}) = G_N(\hat{\imath}, \hat{h}, \hat{s}') = G_N(\hat{\imath}, \hat{h}, \hat{s}'') = \hat{l}$.

Notice that, aside from the fact that the s index is fixed, the restrictions on the scenarios in Equation (4.12) are the same as those for Property 3 (see the definition of set $\mathcal{SP}_{N^3}$ in Equation (4.10)), with the condition $\mathcal{D}^{s,s'} = \{(i', h)\}$ in place of $|\mathcal{D}^{s,s'}| = 1$. This condition is inspired by Gupta and Grossmann (2011) and implies that $|\mathcal{D}^{s,s'}| = 1$.

Step 2c Use the final group number to define the set of indices for all groups corresponding to $\theta_{i',h}$:

$$\mathcal{L}_{i',h} := \{l : l = 1, 2, \ldots, \text{GroupCount}\} \tag{4.13}$$

We index the endogenous scenario groups as $l \in \mathcal{L}_{i',h}$.

For each endogenous parameter $\theta_{i',h}$, the group-number parameter gives the particular index $\hat{l}$ for each $s \in \mathcal{S}$ (i.e., $G_N(i', h, s) = \hat{l}$). We can use this information to define the set of scenarios for each group:

$$ {}^{N}\mathcal{G}^{l}_{i',h} := \{s:\ s\in\mathcal{S},\ \ G_N(i',h,s)=l\} \qquad \forall\, l\in\mathcal{L}_{i',h},\ \ i'\in\mathcal{I},\ \ h\in\mathcal{H}_{i'} \tag{4.14} $$

Note that it is unnecessary to define these groups for every time period, since endogenous realizations are not explicitly associated with any particular time t.

We now define the corresponding set of endogenous scenario pairs, which is at least as restrictive as $\mathcal{SP}_{N^3}$ (we will prove this momentarily), by first linking consecutive scenarios in each group. This is handled separately for each group, as shown in Equation (4.15).

$$ \begin{aligned} \mathcal{SP}^{i',h,l}_{N^4} := \Big\{(t,s,s') :\ & t\in\mathcal{T},\ \ s,s'\in {}^{N}\mathcal{G}^{l}_{i',h}, \\ & s' = \min_{\hat{s}'}\Big(\hat{s}'\in {}^{N}\mathcal{G}^{l}_{i',h},\ \hat{s}'>s\Big), \\ & s < \max_{\hat{s}}\Big(\hat{s}\in {}^{N}\mathcal{G}^{l}_{i',h}\Big), \\ & \{(i',h)\} = \mathcal{D}^{s,s'}\Big\}\ \forall\, l\in\mathcal{L}_{i',h},\ \ i'\in\mathcal{I},\ \ h\in\mathcal{H}_{i'} \end{aligned} \tag{4.15} $$

Although seemingly complex, the expression $s' = \min_{\hat{s}'}\Big(\hat{s}'\in {}^{N}\mathcal{G}^{l}_{i',h},\ \hat{s}'>s\Big)$ simply ensures that scenario s' is the next-highest-indexed scenario immediately following scenario s. The expression $s < \max_{\hat{s}}\Big(\hat{s}\in {}^{N}\mathcal{G}^{l}_{i',h}\Big)$ simply excludes the highest-indexed scenario from the group, since there is no scenario following it with which to form a pair. This is *the same concept* used to define the set of adjacent scenarios, $\mathcal{A}$, previously defined in Equation (4.2) and used in our consecutive pairing approach for first-period and exogenous scenario pairs. The endogenous case is merely a more general formulation that allows us to pair off consecutive scenarios that are nonconsecutively indexed. To prove that this is the case, consider the following: if we replace ${}^{N}\mathcal{G}^{l}_{i',h}$ with $\mathcal{S}$ in the two expressions under discussion, we arrive at $s'=s+1$ from the first and $s<S$ from the second. These are the same two conditions that appear in the definition of set $\mathcal{A}$.

Returning to Equation (4.15), for a given scenario pair (s,s'), the indices i' and h are given by $\{(i',h)\} = \mathcal{D}^{s,s'}$ and correspond to the specific endogenous parameter $\theta_{i',h}$ for which scenarios s and s' differ in possible realizations. Also, notice that the pairs for each group are explicitly generated for every time period, even though they are the same in each period (the reasoning here will become apparent later in this section). Finally, to offer a brief insight into the use of this equation, consider an arbitrary group $\hat{l}$ in the context of Figure 4.4: ${}^{N}\mathcal{G}^{\hat{l}}_{\hat{\imath},\hat{h}} = \{\hat{s},\hat{s}',\hat{s}''\}$.[5] By Equation (4.15), we generate scenario pairs $(\hat{s},\hat{s}')$ and $(\hat{s}',\hat{s}'')$ for each time period; the third, redundant pair $(\hat{s},\hat{s}'')$ is implicitly eliminated. (More specifically, for an arbitrary time period $t=\tau$, we will have tuples $(\tau,\hat{s},\hat{s}'),(\tau,\hat{s}',\hat{s}'')\in\mathcal{SP}^{\hat{\imath},\hat{h},\hat{l}}_{N^4}$, and $(\tau,\hat{s},\hat{s}'')\notin\mathcal{SP}^{\hat{\imath},\hat{h},\hat{l}}_{N^4}$.)

After evaluating Equation (4.15), there will be one set of pairs for each endogenous scenario group. The union of all of these sets gives the complete set of endogenous scenario pairs, as shown in Equation (4.16).

[5] Note that we cannot provide an example in the context of Figure 2.1c, since there are only 2 possible realizations for each endogenous parameter in that case.

$$\mathcal{SP}_{N^4} := \bigcup_{i' \in \mathcal{I}} \left(\bigcup_{h \in \mathcal{H}_{i'}} \left(\bigcup_{l \in \mathcal{L}_{i',h}} \mathcal{SP}_{N^4}^{i',h,l} \right) \right) \tag{4.16}$$

Since this set is at least as restrictive as $\mathcal{SP}_{N^3}$, as previously noted, we claim that $\mathcal{SP}_{N^4} \subseteq \mathcal{SP}_{N^3}$. We now formally state Property 4, by which we prove this claim.

Property 4. *For endogenous NACs, it is sufficient to consider only scenario pairs (s, s') for which s and s' are consecutive scenarios in an endogenous scenario group.*

Proof. See Appendix B.6. □

The following proposition states that, under special circumstances, the proposed approach leads to the minimum number of endogenous scenario pairs.

Proposition 3. *In the case of purely endogenous uncertainty, with no initial 'equality' periods and only one parameter associated with each source, the approach described in Property 4 gives the minimum number of endogenous scenario pairs.*

Proof. See Appendix B.7. □

In the general case considered here, however, it is clear that Proposition 3 does not apply. For instance, with both endogenous and exogenous parameters present in the model, some of the endogenous NACs can be implied through the use of exogenous NACs. A simple example of this can be seen with scenario pairs $(1, 5)$ and $(2, 6)$ at the end of the first time period/beginning of the second time period in Figure 2.1c. We isolate the corresponding scenarios (1, 2, 5, and 6) in Figure 4.5 to clearly illustrate the issue. Notice that if we consider only variables y_τ^s, we have the exogenous NACs $y_2^1 = y_2^2$ and $y_2^5 = y_2^6$ for the beginning of the second period (shown in red). We also have the conditional endogenous NACs $y_2^1 = y_2^5$ (shown in green) and $y_2^2 = y_2^6$ (shown in orange), which are enforced together as long as endogenous parameter θ_2 is unrealized. Recall that the exogenous NACs *always* hold. We can thus use the two exogenous constraints to rewrite the first endogenous constraint as $y_2^2 = y_2^6$. This, of course, is the second endogenous constraint. Accordingly, we can eliminate the endogenous scenario pair $(2, 6)$ since it is already implied by existing pairs.

Because all scenarios in an exogenous scenario group refer to the same state at that point in time, it is only necessary to consider a single endogenous scenario pair between any two exogenous scenario groups. This transitivity argument is formally stated in the following reduction property.

Property 5. *For any two exogenous scenario groups in time period $t = \tau$ (say, ${}^X\mathcal{G}_\tau^{\hat{k}}$ and ${}^X\mathcal{G}_\tau^{\tilde{k}}$), it is sufficient to consider only* one *endogenous scenario pair (s, s') such that s is in one group and s' is in the other (i.e., $s \in {}^X\mathcal{G}_\tau^{\hat{k}}$ and $s' \in {}^X\mathcal{G}_\tau^{\tilde{k}}$, or vice versa).*

Proof. See Appendix B.8. □

Since we require only one endogenous scenario pair between each exogenous scenario group, it is sufficient to consider only a *subset* of scenarios when generating these endogenous pairs. Specifically,

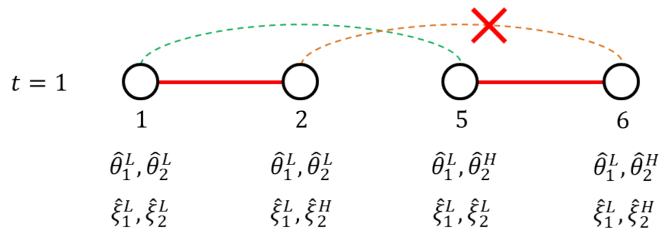


Figure 4.5: Illustration of Property 5.

for each time period $t \in \mathcal{T} \setminus \{T\}$, rather than considering all scenarios in $\mathcal{S}$, we select a single 'representative' scenario from each exogenous scenario group. This gives us a set of *unique* scenarios, $\tilde{\mathcal{U}}_t$, in each period. We use the term 'unique' because all of the scenarios in set $\tilde{\mathcal{U}}_t$ have different realizations for the exogenous parameters up until that point in time and/or different possible realizations for the endogenous parameters.

In selecting these 'representative' scenarios, we must ensure that the corresponding scenario pairs can satisfy Property 3; i.e., $Pos(s) = Pos(s')$, where $s < s'$, and $|\mathcal{D}^{s,s'}| = 1$. We do this by selecting one scenario from each exogenous scenario group *in the first subtree*, and then selecting only scenarios *with the same position* in every other subtree. This procedure is repeated for all $t \in \mathcal{T} \setminus \{T\}$.

For example, consider $t = 1$ in Figure 4.1. If we select scenario 1 from the first group in subtree 1, we must also select scenario 5 from subtree 2, scenario 9 from subtree 3, and scenario 13 from subtree 4. The resulting pairs can satisfy Property 3 since $Pos(1) = Pos(5) = Pos(9) = Pos(13) = 1$. Similarly, if we select scenario 4 from the second group in subtree 1, we must also select scenario 8 from subtree 2, scenario 12 from subtree 3, and scenario 16 from subtree 4. The corresponding pairs can satisfy Property 3 since $Pos(4) = Pos(8) = Pos(12) = Pos(16) = 4$. The set of unique scenarios in this case is then given by $\tilde{\mathcal{U}}_1 = \{1, 4, 5, 8, 9, 12, 13, 16\}$.

For convenience, we simply select the lowest-indexed scenario from each exogenous scenario group (i.e., the first scenario in each group), as shown in Equation (4.17). Specifically, $\tilde{\mathcal{U}}_t$ is expressed as the union of all of these single-scenario sets:

$$\tilde{\mathcal{U}}_t := \bigcup_{k \in \mathcal{K}_t} \left\{ s : \; s = \min_{\hat{s}} \left(\hat{s} \in {}^X\mathcal{G}_t^k \right) \right\} \qquad \forall \, t \in \mathcal{T} \setminus \{T\} \tag{4.17}$$

We let $\tilde{\mathcal{U}}_T := \mathcal{S}$, since there are no exogenous scenario groups defined for $t = T$. Notice that the time index, which was not strictly required in Equation (4.15), will now play a significant role in the definition of the set of endogenous scenario pairs.

In order to define the set of pairs for each endogenous scenario group, $\mathcal{SP}_{N^5}^{i',h,l}$, corresponding to the addition of Property 5, we first restate our earlier definition corresponding to Property 4 (see Equation (4.15)). Our only change is to replace set ${}^N\mathcal{G}_{i',h}^l$ with set ${}^N\mathcal{G}_{i',h}^l \cap \mathcal{S}$, as follows:

$$\begin{aligned}\mathcal{SP}^{i',h,l}_{N^4} := \Big\{(t,s,s') :\ & t\in\mathcal{T},\ \ s,s'\in\left({}^N\mathcal{G}^l_{i',h}\cap\mathcal{S}\right),\\ & s' = \min_{\hat{s}'}\left(\hat{s}'\in\left({}^N\mathcal{G}^l_{i',h}\cap\mathcal{S}\right),\ \hat{s}'>s\right),\\ & s<\max_{\hat{s}}\left(\hat{s}\in\left({}^N\mathcal{G}^l_{i',h}\cap\mathcal{S}\right)\right),\\ & \{(i',h)\}=\mathcal{D}^{s,s'}\Big\}\ \forall\, l\in\mathcal{L}_{i',h},\ \ i'\in\mathcal{I},\ \ h\in\mathcal{H}_{i'}\end{aligned} \tag{4.18}$$

Because $\mathcal{S}$ refers to the complete set of scenarios, the intersection of ${}^N\mathcal{G}^l_{i',h}$ and $\mathcal{S}$ is redundant; there are no scenarios removed from each group, and accordingly, Equation (4.18) is equivalent to Equation (4.15). For Property 5, however, we simply replace set $\mathcal{S}$ in this intersection with a subset of unique scenarios, $\tilde{\mathcal{U}}_t$. The resulting set, ${}^N\mathcal{G}^l_{i',h}\cap\tilde{\mathcal{U}}_t$, further restricts $\mathcal{SP}^{i',h,l}_{N^4}$ such that the endogenous scenario pairs can only be formed among *unique* scenarios in each of the endogenous scenario groups in each time period. This further-restricted set is defined as $\mathcal{SP}^{i',h,l}_{N^5}$ in Equation (4.19).

$$\begin{aligned}\mathcal{SP}^{i',h,l}_{N^5} := \Big\{(t,s,s') :\ & t\in\mathcal{T},\ \ s,s'\in\left({}^N\mathcal{G}^l_{i',h}\cap\tilde{\mathcal{U}}_t\right),\\ & s' = \min_{\hat{s}'}\left(\hat{s}'\in\left({}^N\mathcal{G}^l_{i',h}\cap\tilde{\mathcal{U}}_t\right),\ \hat{s}'>s\right),\\ & s<\max_{\hat{s}}\left(\hat{s}\in\left({}^N\mathcal{G}^l_{i',h}\cap\tilde{\mathcal{U}}_t\right)\right),\\ & \{(i',h)\}=\mathcal{D}^{s,s'}\Big\}\ \forall\, l\in\mathcal{L}_{i',h},\ \ i'\in\mathcal{I},\ \ h\in\mathcal{H}_{i'}\end{aligned} \tag{4.19}$$

Note that we will refer to sets ${}^N\mathcal{G}^l_{i',h}\cap\tilde{\mathcal{U}}_t$ as *reduced* endogenous scenario groups, since for each $l\in\mathcal{L}_{i',h}$, $i'\in\mathcal{I}$, and $h\in\mathcal{H}_{i'}$, this intersection produces a subset of group ${}^N\mathcal{G}^l_{i',h}$ specific to time period $t\in\mathcal{T}$. (For the case of $t=T$, it is also worth noting that ${}^N\mathcal{G}^l_{i',h}\cap\tilde{\mathcal{U}}_T = {}^N\mathcal{G}^l_{i',h}$ since $\tilde{\mathcal{U}}_T := \mathcal{S}$.)

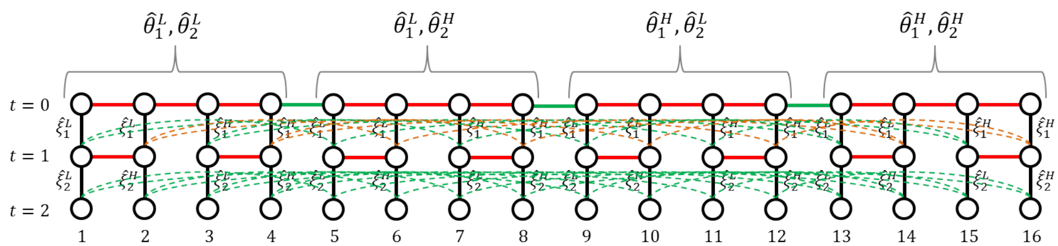
We then take the union of all of the sets of pairs from Equation (4.19) in order to produce the complete set of endogenous scenario pairs, $\mathcal{SP}_{N^5}$, as shown in Equation (4.20). Note that this is the same approach previously used in Equation (4.16) in the context of Property 4.

$$\mathcal{SP}_{N^5} := \bigcup_{i'\in\mathcal{I}}\left(\bigcup_{h\in\mathcal{H}_{i'}}\left(\bigcup_{l\in\mathcal{L}_{i',h}}\mathcal{SP}^{i',h,l}_{N^5}\right)\right) \tag{4.20}$$

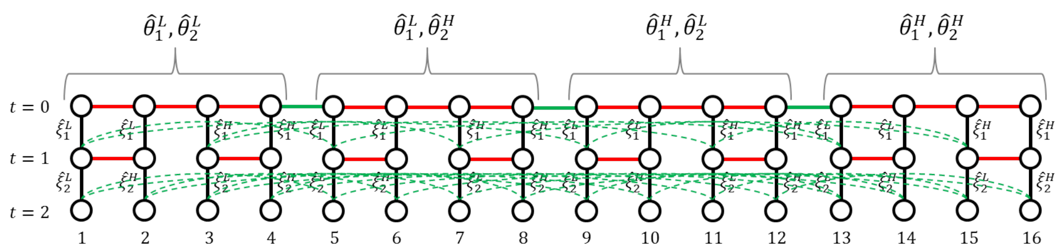
We state that $\mathcal{SP}_{N^5}\subseteq\mathcal{SP}_{N^4}$ based on the proof of Property 5 (see Appendix B.8); in other words, Property 5 may eliminate additional pairs that cannot be removed by Property 4. This conclusion can also be reached by comparing Equation (4.19) to Equation (4.18).

For illustrative purposes, we apply Property 5 to Figure 2.1c in order to remove endogenous scenario pair $(2,6)$ and all other similar pairs at the end of the first time period/beginning of the second time period. Here, we have endogenous scenario groups ${}^N\mathcal{G}^l_1 = \{1,9\}$, ${}^N\mathcal{G}^2_1 = \{2,10\}$, ${}^N\mathcal{G}^3_1 = \{3,11\}$, $\ldots$, ${}^N\mathcal{G}^8_1 = \{8,16\}$ corresponding to θ_1, and ${}^N\mathcal{G}^l_2 = \{1,5\}$, ${}^N\mathcal{G}^2_2 =$

$\{2,6\}$, ${}^N\mathcal{G}_2^3 = \{3,7\}$, $\ldots, {}^N\mathcal{G}_2^8 = \{12,16\}$ corresponding to θ_2. The set of unique scenarios from Property 5 is given by $\tilde{\mathcal{U}}_1 = \{1,3,5,7,9,11,13,15\}$. The intersection ${}^N\mathcal{G}_{i',h}^l \cap \tilde{\mathcal{U}}_1$ then yields the following: ${}^N\mathcal{G}_1^l \cap \tilde{\mathcal{U}}_1 = \{1,9\}$, ${}^N\mathcal{G}_1^2 \cap \tilde{\mathcal{U}}_1 = \emptyset$, ${}^N\mathcal{G}_1^3 \cap \tilde{\mathcal{U}}_1 = \{3,11\}, \ldots, {}^N\mathcal{G}_1^8 \cap \tilde{\mathcal{U}}_1 = \emptyset$, and ${}^N\mathcal{G}_2^l \cap \tilde{\mathcal{U}}_1 = \{1,5\}$, ${}^N\mathcal{G}_2^2 \cap \tilde{\mathcal{U}}_1 = \emptyset$, ${}^N\mathcal{G}_2^3 \cap \tilde{\mathcal{U}}_1 = \{3,7\}, \ldots, {}^N\mathcal{G}_2^8 \cap \tilde{\mathcal{U}}_1 = \emptyset$, respectively. For the groups listed, this corresponds to scenario pairs $(1,5)$, $(1,9)$, $(3,7)$, and $(3,11)$ (or, more specifically, tuples $(1,1,5)$, $(1,1,9)$, $(1,3,7)$, $(1,3,11) \in \mathcal{SP}_{N^5}$). The respective pairs are illustrated in Figure 4.6, along with all remaining (non-listed) pairs for the end of the first time period and the end of the second time period. Notice that at $t = 1$, all conditional endogenous NACs involving non-unique scenarios have been removed. Also, note that this reduction does not apply in the final time period; at that time, there are no exogenous scenarios groups that we can exploit (see Figure 4.1), and all pairs in $\mathcal{SP}_{N^4}$ are present in $\mathcal{SP}_{N^5}$ for $t = T$. This can easily be seen by comparing Equation (4.18) to Equation (4.19) with $\tilde{\mathcal{U}}_T := \mathcal{S}$.



(a) Before applying Property 5, there are several redundant scenario pairs at the end of the first time period/beginning of the second time period (shown in orange).



(b) After applying Property 5, the redundant scenario pairs have been eliminated.

Figure 4.6: Property 5 as applied to Figure 2.1c.

In certain cases (such as Figure 2.1c), the addition of Property 5 leads to the minimum number of endogenous scenario pairs. This is formally stated in the following proposition.

Proposition 4. *In the case of both endogenous and exogenous uncertainties, with no initial 'equality' periods and only one parameter associated with each source, the approach described in Property 4 and supplemented by Property 5 gives the minimum number of endogenous scenario pairs.*

Proof. See Appendix B.9. □

As was the case with Proposition 3, this proposition does not apply in the general case considered here. This is because we may have: (1) endogenous parameters that cannot be realized in some of the initial time periods; and/or (2) multiple endogenous parameters associated with some of the sources of uncertainty. Both of these possibilities have a similar effect on the model.

For the first case, we have *fixed endogenous NACs*, as previously introduced in Section 3.2 (see Equations (3.15)–(3.17)). An example of this is shown in Figure 4.7. Here we consider scenarios 1, 5, 9, and 13 from Figure 2.1c and assume that endogenous parameter θ_2 cannot be realized in the first time period. The four scenarios have identical realizations for the exogenous parameter, and we have four endogenous scenario pairs: $(1, 5)$ and $(9, 13)$, as indicated by solid green lines; $(1, 9)$, as indicated by a dotted green line; and $(5, 13)$, as indicated by a dotted orange line. Notice that scenarios 1 and 5 differ in the possible realization of θ_2 but must be indistinguishable in the first time period because θ_2 cannot be realized at that time. The same is true of scenarios 9 and 13.

If we consider only variables y^s_τ, we have the fixed endogenous NACs $y^1_2 = y^5_2$ and $y^9_2 = y^{13}_2$ for the beginning of the second period. We also have the conditional endogenous NACs $y^1_2 = y^9_2$ and $y^5_2 = y^{13}_2$, which must be enforced together as long as endogenous parameter θ_1 is unrealized. It follows that we can use the two fixed endogenous constraints to rewrite the first conditional endogenous constraint as $y^5_2 = y^{13}_2$. Notice that this is the second conditional endogenous constraint. Accordingly, we can eliminate the endogenous scenario pair $(5, 13)$ since it is already implied by existing pairs. Recall that this result is very similar to what we previously observed in Figure 4.5 with Property 5.

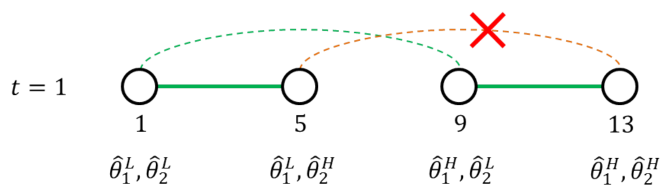


Figure 4.7: Illustration of Property 6 for endogenous parameters that cannot be realized in some of the initial time periods.

For the second case, we have multiple endogenous parameters associated with some of the sources of uncertainty. We use Figure 4.8 to illustrate this and consider 4 scenarios ($\hat{s}$, $\hat{s}'$, $\hat{s}''$, and $\hat{s}'''$) in an arbitrary time period $t = \tau$. There are 2 endogenous parameters ($h = 1$ and $h = 2$) associated with a single source $\hat{\imath}$. It is assumed that the scenarios have identical realizations for all exogenous parameters.

By our existing reduction properties, we generate four scenario pairs: $(\hat{s}, \hat{s}')$, $(\hat{s}, \hat{s}'')$, and $(\hat{s}'', \hat{s}''')$, as shown in green, and $(\hat{s}', \hat{s}''')$, as shown in orange. Each of these pairs consists of scenarios s and s' that differ in the possible realization of an endogenous parameter of *the same source* $\hat{\imath}$. This means that if the uncertainty in source $\hat{\imath}$ has been resolved by the end of time period τ, then all of the scenarios will be distinguishable, and the corresponding NACs will be jointly ignored (Case 1). If the uncertainty has not yet been resolved, then all of the scenarios will be indistinguishable, and the NACs will be jointly enforced (Case 2). Notice that from a modeling perspective, it is only necessary for us to consider Case 2, where the NACs can be viewed as equality constraints. From this viewpoint, it is clear that scenario pair $(\hat{s}', \hat{s}''')$ is redundant and can be eliminated. Recall that this discussion is very similar to what we previously observed in Figure 4.4

in relation to Property 4.

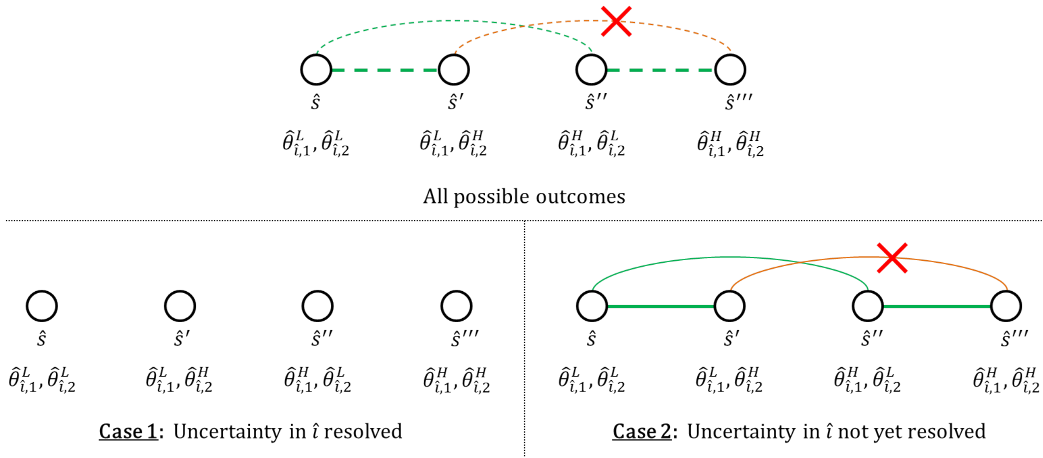


Figure 4.8: Illustration of Property 6 for multiple parameters associated with a single source of endogenous uncertainty.

What we see in these two cases is that some of the remaining endogenous NACs can in fact be implied by other endogenous NACs. To eliminate the corresponding redundant scenario pairs, we extend our definition of unique scenarios.

First, recall that set $\tilde{\mathcal{U}}_t$ from Property 5 provides a sufficient subset of scenarios (in place of the complete set of scenarios, $\mathcal{S}$) that can be considered when generating endogenous scenario pairs. This is based on the presence of exogenous scenario pairs. In a similar manner, provided that the endogenous scenario pairs are generated in sequential order, we may use the existing endogenous pairs to eliminate additional scenarios from set $\tilde{\mathcal{U}}_t$ at each step. For example, in the context of Figure 4.7, after generating scenario pairs $(1,5)$ and $(9,13)$ from the endogenous scenario groups corresponding to θ_2, it is clear that there is no further need to consider scenarios 5 and 13; thus, we may remove these scenarios from the groups for θ_1.

We use this concept to define set $\mathcal{U}_t^{i',h}$ for each endogenous parameter $\theta_{i',h}$ and time period $t \in \mathcal{T}$. Each set indicates the unique scenarios available for forming pairs from the groups corresponding to $\theta_{i',h}$, taking into account all endogenous pairs formed *before this point*. The specific definitions for these sets, as well as the order in which to define them, are given by the unique scenarios algorithm, which we present in Appendix B.10. Note that there is no need to index sets $\mathcal{U}_t^{i',h}$ for $l \in \mathcal{L}_{i',h}$, since the groups corresponding to $\theta_{i',h}$ each contain different scenarios, and any reductions would thus have no effect until we begin forming pairs from the groups of the *next* endogenous parameter.

Next, we formally state the final reduction property, by which we justify the use of the unique scenarios algorithm.

Property 6. *For endogenous NACs, it is sufficient to consider only scenario pairs (s, s') for which s and s' are unique, as defined by the unique scenarios algorithm.*

Proof. See Appendix B.11. □

This leads to the following proposition.

Proposition 5. *In the general case considered throughout this thesis, the approach described in Property 4 and supplemented by Property 5 and Property 6 gives the minimum number of endogenous scenario pairs.*

Proof. See Appendix B.12. □

We now define the set of all scenario pairs (s, s') in each time period t, such that s and s' differ in the possible realization of one endogenous parameter and are identical in all exogenous realizations, with additional redundant pairs eliminated by Properties 4–6. We apply the same general approach as described in Equations (4.15) and (4.19). Specifically, for each endogenous parameter $\theta_{i',h}$, we first link consecutive scenarios in each of the associated endogenous scenario groups $l \in \mathcal{L}_{i',h}$. We define a separate set for each of these groups in Equation (3.22). Note that in keeping with the notation of the previous sets in this section (e.g., $\mathcal{SP}_{N^4}^{i',h,l}$ and $\mathcal{SP}_{N^5}^{i',h,l}$), this set should be named $\mathcal{SP}_{N^6}^{i',h,l}$; however, since we will make no further modifications to the following definition, we will simply refer to this set as $\mathcal{SP}_{N}^{i',h,l}$.

$$\begin{aligned}\mathcal{SP}_{N}^{i',h,l} := \Big\{(t,s,s') :\ & t \in \mathcal{T},\ \ s,s' \in \left({}^{N}\mathcal{G}_{i',h}^{l} \cap \mathcal{U}_t^{i',h}\right),\\ & s' = \min_{\hat{s}'}\left(\hat{s}' \in \left({}^{N}\mathcal{G}_{i',h}^{l} \cap \mathcal{U}_t^{i',h}\right),\ \hat{s}' > s\right),\\ & s < \max_{\hat{s}}\left(\hat{s} \in \left({}^{N}\mathcal{G}_{i',h}^{l} \cap \mathcal{U}_t^{i',h}\right)\right),\\ & \{(i',h)\} = \mathcal{D}^{s,s'}\Big\}\ \forall\, l \in \mathcal{L}_{i',h},\ \ i' \in \mathcal{I},\ \ h \in \mathcal{H}_{i'}\end{aligned} \tag{3.22}$$

Notice that the only change from Equation (4.19) is that we have replaced the reduced endogenous scenario groups ${}^{N}\mathcal{G}_{i',h}^{l} \cap \tilde{\mathcal{U}}_t$ with a further reduced set, ${}^{N}\mathcal{G}_{i',h}^{l} \cap \mathcal{U}_t^{i',h}$, based on Property 6. A brief example of the use of Equation (3.22) is provided at the end of Appendix B.10.

Finally, we take the union of all of the individual scenario-pair sets in Equation (3.23) to obtain set $\mathcal{SP}_N$, the complete set of endogenous scenario pairs. (Again, note that since we will make no further modifications to the following definition, we will refer to this set as $\mathcal{SP}_N$ rather than $\mathcal{SP}_{N^6}$.) This is simply an updated form of Equation (4.20) in which we have replaced set $\mathcal{SP}_{N^5}^{i',h,l}$ with $\mathcal{SP}_{N}^{i',h,l}$.

$$\mathcal{SP}_N := \bigcup_{i' \in \mathcal{I}} \left(\bigcup_{h \in \mathcal{H}_{i'}} \left(\bigcup_{l \in \mathcal{L}_{i',h}} \mathcal{SP}_{N}^{i',h,l} \right)\right) \tag{3.23}$$

Since Property 6 may eliminate additional scenario pairs that cannot be removed by Property 5, we state that $\mathcal{SP}_N \subseteq \mathcal{SP}_{N^5}$ by the proof of Property 6 (see Appendix B.11). This can also be seen by comparing Equation (3.22) to Equation (4.19). It follows that $\mathcal{SP}_N \subseteq \mathcal{SP}_{N^5} \subseteq \mathcal{SP}_{N^4} \subseteq \mathcal{SP}_{N^3}$. Like sets $\mathcal{SP}_F$ and $\mathcal{SP}_X$, the respective scenario pairs in set $\mathcal{SP}_N$ are also non-unique.

In the case of purely endogenous uncertainty, it is worth noting that $\tilde{\mathcal{U}}_t = \mathcal{S}$ for all $t \in \mathcal{T}$ (since, in each time period, scenario 1 will be assigned to exogenous scenario group 1, and every other scenario will be assigned to a separate group by Equation (4.5) (due to $\mathcal{SP}_X = \emptyset$)). Thus, Equations (3.22) and (3.23) are also applicable for purely-endogenous problems, as previously suggested with the use of $\mathcal{SP}_N$ in model ($\mathrm{MSSP_N}$).

We now formulate one final set in this section. Recall that in model (MSSP) (and accordingly, model ($\mathrm{MSSP_N}$)), there are many cases where we require the *source* i' of the endogenous parameter for which scenarios s and s' differ in possible realizations. The reason, of course, is that there is some information that is specific to the source itself.

For example, we may make an investment in a source to reveal uncertain parameter values, and there may be a certain number of initial time periods (i.e., a lead time) before we can observe these values. The investment decision $b^s_{i',t}$ and hence the indistinguishability of scenarios are both specific to the source i'. We require this index to evaluate our uncertainty-resolution rule (see Equation (3.19)). The set of initial 'equality' time periods $\mathcal{T}_E^{i'}$ and thus the remaining 'conditional' periods $\mathcal{T}_C^{i'}$ are also both specific to the source. Accordingly, the index i' is required in all endogenous non-anticipativity constraints. To further emphasize our point, notice that $b^s_{i',t}$, $\mathcal{T}_E^{i'}$, and $\mathcal{T}_C^{i'}$ are *not* indexed for any particular parameter h.

The previously-defined set $\mathcal{D}^{s,s'}$ indicates the specific *parameter* $\theta_{i',h}$ for which s and s' differ in possible realizations. We now define set $\hat{\mathcal{D}}^{s,s'}$ to indicate only the associated *source*, i':

$$\hat{\mathcal{D}}^{s,s'} := \left\{ i' : i' \in \mathcal{I},\ \left(\exists\, h \in \mathcal{H}_{i'} : (i', h) \in \mathcal{D}^{s,s'} \right) \right\} \qquad \forall\, s, s' \in \mathcal{S},\ \ s < s',\ \ Pos(s) = Pos(s') \tag{4.21}$$

where we specify that there exists *at least one* endogenous parameter h associated with source i' for which scenarios s and s' differ in possible realizations. (Due to Property 3, there will be *exactly one* endogenous parameter h in each case.)

4.4 Conclusions

In the previous three sections, we have presented 6 theoretical reduction properties that eliminate all redundant scenario pairs. This, in turn, eliminates all redundant non-anticipativity constraints, which can significantly reduce the dimensionality of our multistage stochastic programming model, (MSSP), as compared to the case where no reduction properties are applied.

Note that in the reduced form of the model, first-period scenario-pair set $\mathcal{SP}_F$ is defined in Equations (3.10) and (4.2), exogenous scenario-pair set $\mathcal{SP}_X$ is defined in Equation (3.11), and endogenous scenario-pair set $\mathcal{SP}_N$ is defined in Equations (3.22) and (3.23). The NACs in model (MSSP) are expressed in terms of these sets. In other words, with the stated definitions, this model is in reduced form, and no further reduction is possible.[6]

[6] The same can be said of models ($\mathrm{MSSP_X}$) and ($\mathrm{MSSP_N}$), which are simply special cases of model (MSSP). Also, note that this statement applies to the general formulations considered in this thesis; further reduction may be possible in specific problem instances.

Chapter 5

Solution Methods and Numerical Results

Even after eliminating redundant scenario pairs with Properties 1–6, model (MSSP) is often still too large to solve directly with commercial MILP solvers. We thus rely on alternative solution methods. Specifically, we consider a novel sequential scenario decomposition heuristic and Lagrangean decomposition.

5.1 Sequential Scenario Decomposition Heuristic

The first alternative solution method that we will discuss is a heuristic that we refer to as sequential scenario decomposition (SSD). The basic idea behind this algorithm is that we sequentially solve endogenous MILP subproblems to determine the binary investment decisions, fix these decisions to satisfy the corresponding first-period and exogenous NACs, and then solve the resulting model to obtain a feasible solution to the original problem.

More specifically, we start at $t = 1$ in model (MSSP) and select one scenario from each exogenous scenario group. This subset of scenarios will be connected by only first-period and endogenous NACs, since we have effectively removed all of the exogenous constraints by disregarding many of the scenarios. We then solve this endogenous MILP subproblem (a modified form of model ($\mathrm{MSSP_N}$)) and extract the binary investment decisions from the solution. Returning to the original problem, we fix the respective binary first-stage decisions in all scenarios, and for all other time periods, we fix the binary here-and-now decisions in all scenarios that belong to the same exogenous scenario groups as the subproblem scenarios. We then proceed to the next time period and repeat this process (excluding the consideration of binary first-stage decisions, as these have already been fixed), selecting only scenarios that have not been considered in any previous subproblem. We continue until we reach $t = T - 1$; this is the last subproblem, as we are solving for binary here-and-now decisions for the next time period, and there are no such decisions for $t = T$. After this process is complete, all binary investment decisions will be fixed in model (MSSP). This means that the scenario tree is fixed and we no longer have conditional constraints. The solution of this model gives a feasible solution to the original problem. In Figure 5.1, we demonstrate the first iteration of the algorithm.

The primary motivation for this procedure is that the subproblems should be considerably easier to solve than the original model. Furthermore, as shown in Figure 5.1, the first "easy" subproblem includes all of the unique scenarios in the first time period; thus, at the beginning of the planning horizon, we have the same level of information as the original model. The quality of information gradually deteriorates as we proceed forward in time since (by design) some required scenarios are not considered until later subproblems. For instance, in Figure 5.1, scenarios 3, 7, 11, and 15 are excluded from the first subproblem and thus the model is unaware of the possibility of a *high* demand in the second time period. This demand is accounted for in the next subproblem, after investment decisions have already been fixed in all scenarios at the beginning of the first and second time periods and in half of the scenarios at the beginning of the third time period, based on partial information (see Figure 5.1c). To our benefit, however, this is typically not a significant concern. In problems with endogenous uncertainty, investment decisions are often made early in the planning horizon, at which point we still have "mostly complete" information. Hence, the subproblem data may not be extensive enough to determine optimal values for the continuous variables, but should be sufficient to approximate the optimal "yes" or "no" investment decisions.

We assume that fixing binary decisions for time period t does not render any later-period subproblems infeasible. Note that when we refer to "binary decisions," we are referring to *all* binary here-and-now decisions $b^s_{i,t}$, as well as any binary components of variable vector y^s_t. For convenience of notation, however, we will represent all such binary decisions as $b^s_{i,t}$ in this section. We next present the algorithm.

Sequential Scenario Decomposition Algorithm

Step 1 Generate all parameters and sets required for model (MSSP).

Step 2 Determine the set of scenarios $\mathcal{S}^{\hat{t}}_{SSD}$ for each subproblem $\hat{t} \in \mathcal{T} \setminus \{T\}$. This is done as follows: for each subproblem $\hat{t}$, select *one* scenario from each exogenous scenario group in this time period (i.e., $s \in \tilde{\mathcal{U}}_{\hat{t}}$), excluding all scenarios in previous subproblems (i.e., $s \notin \bigcup_{\hat{\tau}\in\mathcal{T},\,\hat{\tau}<\hat{t}} \mathcal{S}^{\hat{\tau}}_{SSD}$). We exclude the final time period because there are no exogenous scenario groups defined for $t = T$, and we cannot make new here-and-now decisions at the end of the time horizon. Set $\mathcal{S}^{\hat{t}}_{SSD}$ is then given by:

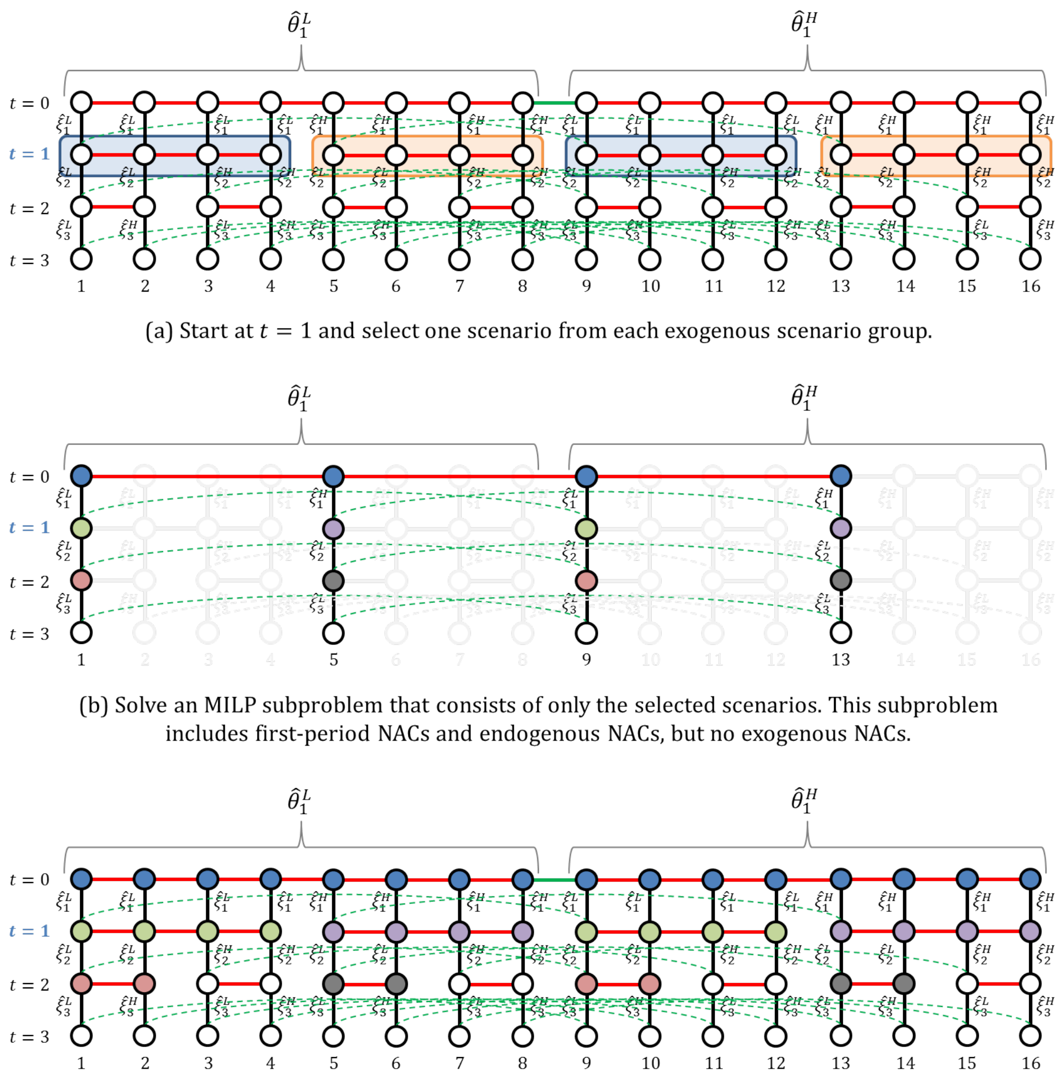
$$\mathcal{S}^{\hat{t}}_{SSD} := \left\{ s : \, s \in \tilde{\mathcal{U}}_{\hat{t}} \setminus \bigcup_{\hat{\tau}\in\mathcal{T},\,\hat{\tau}<\hat{t}} \mathcal{S}^{\hat{\tau}}_{SSD} \right\} \qquad \forall\, \hat{t} \in \mathcal{T}, \;\; \hat{t} < T \tag{5.1}$$

In Figure 5.1, the set of scenarios for the first subproblem is given by $\mathcal{S}^1_{SSD} := \{s : \, s \in \{1, 5, 9, 13\} \setminus \emptyset\} = \{1, 5, 9, 13\}$. For the second subproblem (not shown), $\mathcal{S}^2_{SSD} := \{s : \, s \in \{1, 3, 5, 7, 9, 11, 13, 15\} \setminus \{1, 5, 9, 13\}\} = \{3, 7, 11, 15\}$.

Step 3 For $\hat{t} = 1, 2, \ldots, T-1$:

Step 3a Redefine set $\mathcal{A}$ (Equation (4.2)), and thus set $\mathcal{SP}_F$ (Equation (3.10)), using the set of scenarios for subproblem $\hat{t}$ (i.e., $\mathcal{S} := \mathcal{S}^{\hat{t}}_{SSD}$ and $S := |\mathcal{S}^{\hat{t}}_{SSD}|$).

Step 3b Generate subproblem $\hat{t}$, as shown in Figure 5.1b. This is a modified form of model (MSSP_N), where $\mathcal{S}_N := \mathcal{S}^{\hat{t}}_{SSD}$ and the non-anticipativity constraints for $b^s_{i,t}$ depend on the subproblem number, $\hat{t}$. Accordingly, replace first-period NACs (3.14), fixed



(a) Start at $t = 1$ and select one scenario from each exogenous scenario group.

(b) Solve an MILP subproblem that consists of only the selected scenarios. This subproblem includes first-period NACs and endogenous NACs, but no exogenous NACs.

(c) Extract the binary investment decisions from the subproblem solution, and fix these decisions in the scenarios of the original problem in order to satisfy first-period and exogenous NACs.

Figure 5.1: Sequential scenario decomposition heuristic (first subproblem).

endogenous NACs (3.16), and conditional endogenous NACs (3.25) (assuming a big-M reformulation is used) with constraints (5.2)–(5.4), respectively. The idea behind these modifications is that first-period NACs for $b^s_{i,t}$ are no longer needed after the corresponding decisions are fixed in the first subproblem. Thus, Equation (5.2) enforces them for only the first subproblem, $\hat{t} = 1$. Similarly for the endogenous constraints, at time t, decisions $b^s_{i,t}$ will have been fixed in all earlier time periods $t < \hat{t}$ by previous subproblems; hence, we consider these constraints for only $\hat{t} \le t < T$ in Equations (5.3) and (5.4).

$$b^s_{i,1} = b^{s'}_{i,1} \qquad \hat{t} = 1,\ \forall\,(s,s') \in \mathcal{SP}_F,\ \ i \in \mathcal{I} \tag{5.2}$$

$$b^s_{i,t+1} = b^{s'}_{i,t+1} \qquad \forall\,(t,s,s') \in \mathcal{SP}_N,\ \ t \in \mathcal{T}^{i'}_E,\ \ t \ge \hat{t},\ \ \{i'\} = \hat{\mathcal{D}}^{s,s'},\ \ i \in \mathcal{I} \tag{5.3}$$

$$\begin{aligned} -(1 - z^{s,s'}_t) \le b^s_{i,t+1} - b^{s'}_{i,t+1} \le (1 - z^{s,s'}_t) \\ \forall\,(t,s,s') \in \mathcal{SP}_N,\ \ t \in \mathcal{T}^{i'}_C,\ \ \hat{t} \le t < T,\ \ \{i'\} = \hat{\mathcal{D}}^{s,s'},\ \ i \in \mathcal{I} \end{aligned} \tag{5.4}$$

Next, solve subproblem $\hat{t}$. Note that we preserve all endogenous NACs in each time period, so there is no need to update set $\mathcal{SP}_N$.

In some cases, the heuristic subproblem may be too difficult to solve directly. One viable option here is Lagrangean decomposition; specifically, we may apply the endogenous scenario grouping approach described in Gupta and Grossmann (2014a), since these are purely-endogenous problems.

Step 3c If $\hat{t} = 1$, this is the first subproblem. Use the binary first-stage decisions from this subproblem (i.e., $b^{\hat{s}}_{i,1}\ \ \forall\, i \in \mathcal{I},\ \hat{s} \in \mathcal{S}^{\hat{t}}_{SSD}$) to fix the binary first-stage decisions in *all* scenarios (i.e., $b^s_{i,1}\ \ \forall\, i \in \mathcal{I},\ s \in \mathcal{S}$). This is shown in Figure 5.1c, where the nodes at the beginning of the first time period (originally white in Figure 5.1a) have now been shaded in blue. Because the first-stage decisions must be identical in all scenarios, we arbitrarily use only the decisions from the first scenario, $\hat{s} = 1$, instead of considering all $\hat{s} \in \mathcal{S}^{\hat{t}}_{SSD}$. Decisions are fixed as shown in Equation (5.5). Note that this step allows us to satisfy the first-period NACs in the original problem, (MSSP).

$$b^s_{i,1} := b^{\hat{s}}_{i,1} \qquad \hat{s} = 1,\ \forall\, i \in \mathcal{I},\ \ s \in \mathcal{S} \tag{5.5}$$

Step 3d Fix binary here-and-now decisions in all other time periods. This is done as follows: for each subproblem scenario $\hat{s} \in \mathcal{S}^{\hat{t}}_{SSD}$, start at $t = \hat{t}$ and fix decisions $b^{\hat{s}}_{i,\hat{t}+1}$ in all scenarios in the same exogenous scenario group as $\hat{s}$. We use the condition $G_X(t,s) = G_X(t,\hat{s})$ to check that scenario $s \in \mathcal{S}$ is in the same group as $\hat{s}$.[1] For each remaining time period $t < T$, we repeat this process of fixing decisions $b^{\hat{s}}_{i,t+1}$ in the respective scenario groups. In Figure 5.1c, for instance, scenario $\hat{s} = 1$ is considered in the first subproblem. Since scenarios $s = 1$, 2, 3, and 4 are all in the same group as $\hat{s}$ in

[1] Rather than the conditions $s \in \mathcal{S}$, $G_X(t,s) = G_X(t,\hat{s})$ in Equation (5.6), we could state $s \in {}^X\mathcal{G}^{\hat{k}}_t$, where $\hat{k}$ is the group number corresponding to $\hat{s}$, given by $\hat{k} = G_X(t,\hat{s})$.

time period 1, we fix their binary decisions in that period based on those of $\hat{s}$. In time period 2, scenarios $s = 1$ and 2 are in the same group as $\hat{s}$, and we fix their binary decisions in an identical manner. This is represented by a change in color of the nodes as compared to Figure 5.1a. Note that we solve each subproblem for the full time horizon $t \in \mathcal{T}$, but we only fix decisions for $\hat{t} \leq t < T$ since the decisions for all previous time periods have already been fixed in the previous subproblems. This step allows us to satisfy the exogenous NACs in the original problem, (MSSP).

$$b^{s}_{i,t+1} := b^{\hat{s}}_{i,t+1} \qquad \forall\, i \in \mathcal{I},\ t \in \mathcal{T},\ \hat{t} \leq t < T, \\ s \in \mathcal{S},\ \hat{s} \in \mathcal{S}^{\hat{t}}_{SSD},\ G_X(t,s) = G_X(t,\hat{s}) \tag{5.6}$$

Step 4 At this point, the binary here-and-now decisions $b^{s}_{i,t}$ have been fixed for all $i \in \mathcal{I}$, $t \in \mathcal{T}$, and $s \in \mathcal{S}$. (In Figure 5.1, this would occur after one more iteration.) Thus, in model (MSSP), drop all NACs related to these decisions. This includes the first-period NACs given by Equation (3.14), the exogenous NACs given by Equation (3.33), the fixed endogenous NACs given by Equation (3.16), and the conditional endogenous NACs given by either Equation (3.25) or the third line of Equation (3.18). Note that the scenario tree is fixed at this point, since indistinguishability can be determined by directly calculating $Z^{s,s'}_{t}$ (and thus $z^{s,s'}_{t}$) from the known values of $b^{s}_{i,t}$.

Next, redefine set $\mathcal{A}$ and set $\mathcal{SP}_F$ using the complete set of scenarios (i.e., $S := |\mathcal{R}|$ by Equation (2.6) and $\mathcal{S} := \{s : s = 1, 2, \ldots, S\}$). Then, solve the resulting form of model (MSSP). This provides a feasible, but not necessarily optimal, solution to (MSSP).

Note that since the scenario tree is fixed in the final form of model (MSSP), the timing of all realizations is known in advance; thus, all uncertainties can be viewed as exogenous. This model, however, is not in the form of a purely-exogenous stochastic program (i.e., ($\mathrm{MSSP_X}$)). For large instances where a direct-solution approach is impractical, we have two basic options: (1) preserve the structure and apply Lagrangean decomposition, as discussed in the next section; or (2) reformulate the problem into the form of model ($\mathrm{MSSP_X}$), as shown graphically in Section 5.3.1. In the latter case, we can take advantage of effective solution methods for purely-exogenous MSSP problems, such as the branch-and-fix coordination scheme by Escudero et al. (2009).

This heuristic can be used to obtain an initial upper bound in a Lagrangean decomposition algorithm, as discussed in the next section.

5.2 Lagrangean Decomposition

From Figure 2.1c, it is clear that if we remove all non-anticipativity constraints, then the scenario tree decomposes into independent scenarios. This is shown in Figure 5.2. The appealing aspect of this structure is that independent scenario subproblems should be considerably easier to solve than the full model. Such reasoning is the primary motivation behind Lagrangean decomposition, in which 'complicating' (i.e., 'linking') constraints are dualized in order to achieve a similar relaxation of the original model (Carøe and Schultz, 1999; Goel and Grossmann, 2006; Gupta and Grossmann, 2011; Escudero et al., 2016b). In this context, the complicating constraints are the NACs.

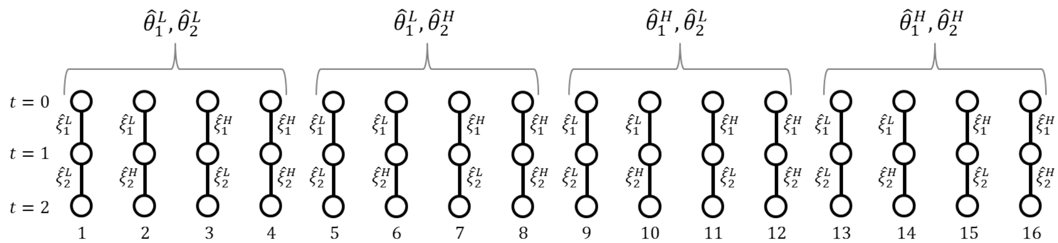


Figure 5.2: A scenario tree decomposes into independent scenarios when all NACs are removed.

As described in Gupta and Grossmann (2014a), in the case of standard Lagrangean decomposition for MSSP problems with endogenous uncertainties, the first step is to relax all of the conditional endogenous NACs. We then form the Lagrangean relaxation (Guignard, 2003) by dualizing the first-period and fixed endogenous NACs. This entails moving these constraints to the objective function as penalty terms multiplied by Lagrange multipliers. In our case, we must also dualize the exogenous NACs (Goel and Grossmann, 2006). We use a simplified form of model (MSSP) to illustrate this, (MSSPS), where for simplicity we keep only decision variables y_t^s. We also assume that the set of initial 'equality' periods is identical for all sources of endogenous uncertainty; i.e., $\mathcal{T}_E^{i'} = \mathcal{T}_E$, and thus $\mathcal{T}_C^{i'} = \mathcal{T}_C$, for all $i' \in \mathcal{I}$.

(MSSPS)

$$\min_y \phi = \sum_{s\in\mathcal{S}} p^s \sum_{t\in\mathcal{T}} {}^y c_t^s y_t^s \tag{5.7}$$

$$\text{s.t.} \quad \sum_{\tau=1}^{t} {}^y A_{\tau,t}^s y_\tau^s \le a_t^s \qquad \forall\, t \in \mathcal{T},\ \ s \in \mathcal{S} \tag{5.8}$$

$$y_1^s = y_1^{s'} \qquad \forall\, (s,s') \in \mathcal{SP}_F \tag{3.6}$$

$$y_{t+1}^s = y_{t+1}^{s'} \qquad \forall\, (t,s,s') \in \mathcal{SP}_X \tag{3.8}$$

$$y_{t+1}^s = y_{t+1}^{s'} \qquad \forall\, (t,s,s') \in \mathcal{SP}_N,\ \ t \in \mathcal{T}_E \tag{3.17}$$

$$-y_{t+1}^{UB}(1 - z_t^{s,s'}) \le y_{t+1}^s - y_{t+1}^{s'} \le y_{t+1}^{UB}(1 - z_t^{s,s'})\ \forall\, (t,s,s') \in \mathcal{SP}_N,\ t \in \mathcal{T}_C,\ t < T \tag{3.26}$$

$$Z_t^{s,s'} \Leftrightarrow F(y_1^s, y_2^s, \dots, y_t^s) \qquad \forall\, (t,s,s') \in \mathcal{SP}_N,\ \ t \in \mathcal{T}_C \tag{5.9}$$

$$y_t^s \in \mathcal{Y}_t^s \qquad \forall\, t \in \mathcal{T},\ \ s \in \mathcal{S} \tag{5.10}$$

$$Z_t^{s,s'} \in \{True, False\} \qquad \forall\, (t,s,s') \in \mathcal{SP}_N,\ \ t \in \mathcal{T}_C \tag{3.21}$$

$$z_t^{s,s'} \in \{0,1\} \quad \forall\, (t,s,s') \in \mathcal{SP}_N,\ \ t \in \mathcal{T}_C \tag{3.27}$$

In the simplified Lagrangean relaxation problem, (MSSPS-LR), we remove endogenous constraints (3.26), (5.9), (3.21), and (3.27), and dualize constraints (3.6), (3.8), and (3.17).

(MSSPS-LR)

$$\begin{aligned} \min_{y} \phi_{LR}(\lambda) = \sum_{s \in \mathcal{S}} p^s \sum_{t \in \mathcal{T}} {}^{y}c_t^s y_t^s + \sum_{(s,s') \in \mathcal{SP}_F} {}^{F}\lambda_1^{s,s'} (y_1^s - y_1^{s'}) \\ + \sum_{(t,s,s') \in \mathcal{SP}_X} {}^{X}\lambda_t^{s,s'} (y_{t+1}^s - y_{t+1}^{s'}) + \sum_{\substack{(t,s,s') \in \mathcal{SP}_N \\ t \in \mathcal{T}_E}} {}^{N}\lambda_t^{s,s'} (y_{t+1}^s - y_{t+1}^{s'}) \end{aligned} \tag{5.11}$$

$$\text{s.t.} \quad \sum_{\tau=1}^{t} {}^{y}A_{\tau,t}^s y_\tau^s \le a_t^s \qquad \forall\, t \in \mathcal{T}, \;\; s \in \mathcal{S} \tag{5.8}$$

$$y_t^s \in \mathcal{Y}_t^s \qquad \forall\, t \in \mathcal{T}, \;\; s \in \mathcal{S} \tag{5.10}$$

Notice that all complicating constraints have now been either removed or dualized, and Equations (5.8) and (5.10) apply only to individual scenarios. Further notice, however, that the objective function, Equation (5.11), still contains variables y_t^s and $y_t^{s'}$, so we cannot yet decompose the problem by scenario. We expand this expression, swap indices s and s' in certain summations, and then simplify in order to rewrite the objective function as Equation (5.12) (see Appendix C.1 for further details).

$$\begin{aligned} \min_{y} \phi_{LR}(\lambda) = \sum_{s \in \mathcal{S}} \Bigg(p^s \sum_{t \in \mathcal{T}} {}^{y}c_t^s y_t^s + y_1^s \Bigg(\sum_{(s,s') \in \mathcal{SP}_F} {}^{F}\lambda_1^{s,s'} - \sum_{(s',s) \in \mathcal{SP}_F} {}^{F}\lambda_1^{s',s} \Bigg) \\ + \sum_{t \in \mathcal{T} \setminus \{T\}} y_{t+1}^s \Bigg(\sum_{(t,s,s') \in \mathcal{SP}_X} {}^{X}\lambda_t^{s,s'} - \sum_{(t,s',s) \in \mathcal{SP}_X} {}^{X}\lambda_t^{s',s} \Bigg) \\ + \sum_{t \in \mathcal{T}_E} y_{t+1}^s \Bigg(\sum_{(t,s,s') \in \mathcal{SP}_N} {}^{N}\lambda_t^{s,s'} - \sum_{(t,s',s) \in \mathcal{SP}_N} {}^{N}\lambda_t^{s',s} \Bigg) \Bigg) \end{aligned} \tag{5.12}$$

The variables in the objective function now involve only scenario s, and all other terms are constants. Accordingly, the problem can be decomposed into independent scenario subproblems that can be solved in parallel. This is done in an iterative fashion, as shown in Figure 5.3 (adapted from Gupta and Grossmann, 2011. In each iteration, we first solve the subproblems with fixed multipliers to obtain a lower bound to the original problem (MSSPS). The lower bound is simply equal to the sum of the subproblem objective function values, and an upper bound is determined by a simple heuristic. In this heuristic, we selectively fix decisions from the subproblems in the original problem to obtain a feasible solution (see Appendix C.2 for complete details). The solution from the sequential scenario decomposition heuristic may be used as an initial upper bound; however, this is not required. We then apply the subgradient method (Fisher, 1985) to update the multipliers for the Lagrangean problem,[2] and repeat this process until the difference between the upper bound and lower bound lies within a pre-specified tolerance or until a maximum iteration limit is reached. Note that if we are unable to sufficiently close the gap, it may be necessary to implement a branch-and-bound algorithm such as the one proposed by Goel and Grossmann (2006) and Goel et al.

[2] There are many alternative multiplier-update procedures. See Escudero et al. (2013b), as well as Oliveira et al. (2013) and the references therein.

(2006).

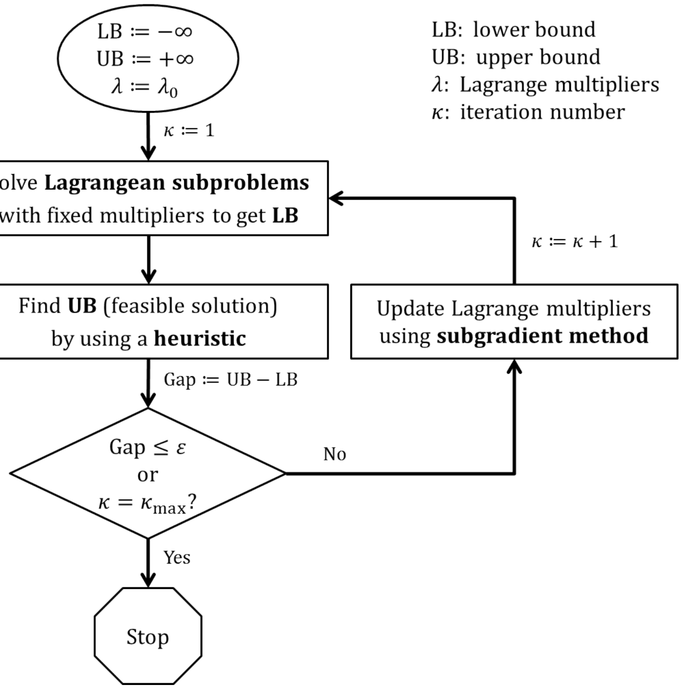


Figure 5.3: Algorithm for Lagrangean decomposition.

5.3 Numerical Results

5.3.1 Motivating Example

Consider the simple process network shown in Figure 5.4, as adapted from Goel and Grossmann (2006). In this example, a product A is produced in Process III which has an existing capacity of 3 tons/hr and a known yield of 70%. This process requires a feed of chemical B that is currently purchased. The demand of product A is uncertain but must be satisfied for each time period in the planning horizon. If the demand cannot be met by production, product A is purchased from a competitor.

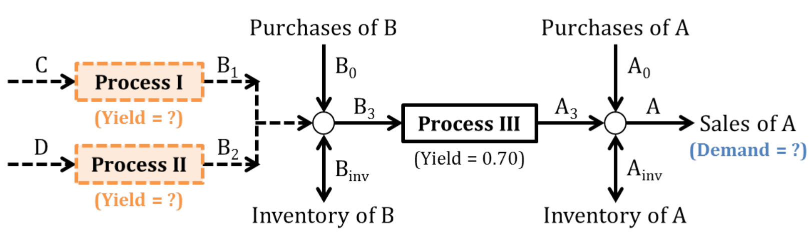


Figure 5.4: Process network for the motivating example.

Due to the high price of B, it is proposed that some (or all) of this chemical be manufactured from raw material C in a new process, Process I, or from raw material D in a second new process, Process II. These processes are not exclusive, and neither, one, or both may be installed.

Table 5.1: Model statistics for the motivating example.

Problem Type	Scenarios	Constraints	Continuous Variables	Binary Variables
Fullspace	16	5,985	913	240
Reduced Model	16	1,472	913	120
SSD (Sub 1)	8	784	457	64
SSD (Final)	16	1,355	913	24
LD	4	286	229	24

The yield of Process I is uncertain, with possible realizations $\{0.69, 0.81\}$, both with an equal probability of 0.5. The yield of Process II is also uncertain, with possible realizations $\{0.62, 0.85\}$, both with equal probabilities. The objective is to determine the optimal investment and operation decisions over a 2-year planning horizon in order to maximize the total expected profit from the sales of A. Over this time horizon, the demand of product A has possible realizations $\{1.10, 3.10\}$ tons/hr in time period 1 and $\{2.25, 4.25\}$ tons/hr in time period 2, each with probability 0.5. We do not provide the remaining problem data here; however, this data is available upon request.

Regarding the *types* of uncertainty, the yields of Process I and Process II represent endogenous parameters (θ_1 and θ_2, respectively), since they are uncertain until the units are installed and operated. For simplicity, we assume that the units are operated immediately after they are installed. The demand of product A is an exogenous parameter (ξ_t), as it is a market value that will be realized automatically in each time period. There are 4 possible combinations of realizations for the endogenous parameters and 4 possible combinations of realizations for the exogenous parameters. This gives rise to a 3-stage, 16-scenario stochastic programming problem. Note that this corresponds to the composite scenario tree previously introduced in Figure 2.1c. We first use a direct-solution approach to solve the *fullspace model* (i.e., model (MSSP) with no reduction properties applied) and the *reduced model* (i.e., model (MSSP) in its current form, with all reduction properties applied). We then apply the sequential scenario decomposition (SSD) heuristic and Lagrangean decomposition (LD). The corresponding model statistics are provided in Table 5.1.

First, we observe that by applying Properties 1–6 through the set definitions proposed in Chapter 4, we are able to reduce the total number of constraints from 5,985 to 1,472; a 75% reduction, based solely on the removal of redundant NACs. We are also able to eliminate half of the binary variables (specifically, the indistinguishability variables $z_t^{s,s'}$ associated with redundant conditional endogenous NACs). This effect can be even more pronounced in larger problem instances, as will be shown in the next section.

Furthermore, the SSD heuristic requires only one subproblem to fix all of the binary here-and-now decisions. This subproblem consists of scenarios 1, 3, 5, 7, 9, 11, 13, and 15 (see Figure 2.1c). Recall that we fix the respective binary decisions in these scenarios, and all remaining scenarios, in order to satisfy the corresponding first-period and exogenous non-anticipativity constraints.

Table 5.1 also indicates that there are still binary variables in the final SSD problem, SSD (Final). This is due to the indistinguishability variables $z_t^{s,s'}$, which are simply calculated quantities given the fixed values of $b_{i,t}^s$. We may choose to either fix these variables prior to generating the model, or allow the solver to perform these calculations. For convenience, we choose the

Table 5.2: Numerical results for the motivating example.

Problem Type	Total Expected Profit ($MM)		Optimality Gap	Solution Time (s)
	Lower Bound	Upper Bound		
Fullspace	5.069	5.069	0%	0.08
Reduced Model	5.069	5.069	0%	0.06
SSD	5.069	–	–	0.11
LD	5.069	5.069	0.006%	11.70

latter option in this case. Note that when there are no other integer variables in the problem, and indistinguishability is determined by Equations (3.29) and (3.30), we may obtain the optimal solution of the final SSD problem by solving its LP relaxation. This is because, by these inequalities, $z_t^{s,s'}$ must be 0 or 1 if $b_{i,t}^s$ is also binary.

The problem size reported for Lagrangean decomposition corresponds to the size of each Lagrangean subproblem. For this example, we decompose the problem such that each subproblem corresponds to one subtree (i.e., 4 scenarios with all non-anticipativity constraints intact), rather than one individual scenario. Thus, we must dualize only 3 sets of first-period NACs, which gives 4 independent subproblems of 4 scenarios each.

Note that this Lagrangean-decomposition strategy is inspired by the scenario clustering approach of Escudero et al. (2016b); we will use a similar strategy for our LD implementations in the following two sections as well. While, in principle, we may also merge indistinguishable nodes within each subtree such that all non-dualized first-period and exogenous NACs are implicitly enforced (i.e., for each subtree, we may adopt the standard form shown in Figure 1.2a), this would require significantly more complex notation which we wish to avoid.

We solve the motivating example in GAMS 24.3.3, with CPLEX 12.6.0.1, on a machine with a 2.50 GHz Intel Core i5 CPU and 4 GB of RAM. The optimal solution is to install Process I at the beginning of the first time period with a capacity of 3.704 tons/hr and perform no expansions. The total expected profit is $5.069 MM. The computational results are given in Table 5.2. Note that each reported solution time reflects only the solver time and does *not* include the model generation time. Given the complexity of the parameters and sets defined in Chapter 4, it is also worth noting that the generation time for the reduced model is less than one minute for all example problems in this chapter.

We observe that the SSD heuristic obtains the optimal solution as a lower bound, and after 14 iterations, the Lagrangean decomposition algorithm converges to the optimal solution. Figure 5.5 shows the best bounds obtained by LD at each iteration of the algorithm. Since this is a very simple example, it is faster in this case to directly solve the reduced model than it is to solve subproblems in the alternative solution methods. For larger instances, these alternative methods yield considerable savings in computational time, as will be seen in the next section.

The optimal structure of the composite scenario tree is shown in Figure 5.6. Notice that, starting from the superstructure form in Figure 2.1c, the dotted green lines have transitioned into solid green lines for active NACs and have disappeared entirely for inactive NACs. We also show

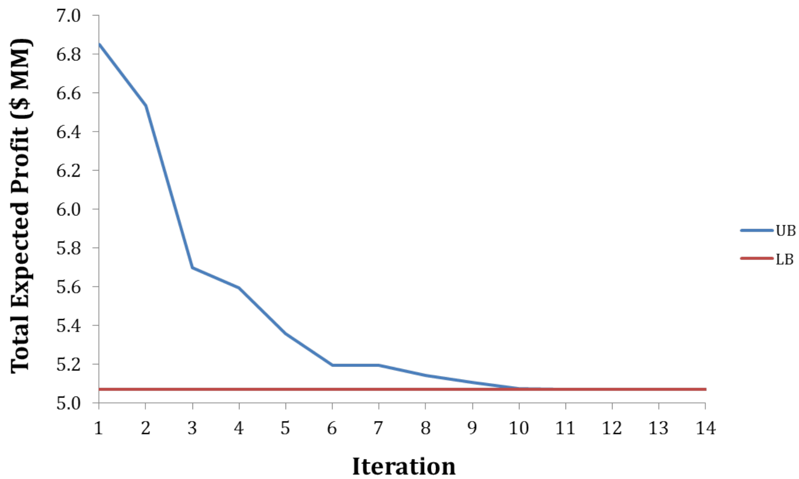


Figure 5.5: Best bounds on the optimal solution of the motivating example, as obtained by Lagrangean decomposition.

that by taking advantage of the active NACs and the known timing of the endogenous realizations, we are able to recover the standard form of the scenario tree. This form is significantly easier to interpret, as can be seen in Figure 5.6.

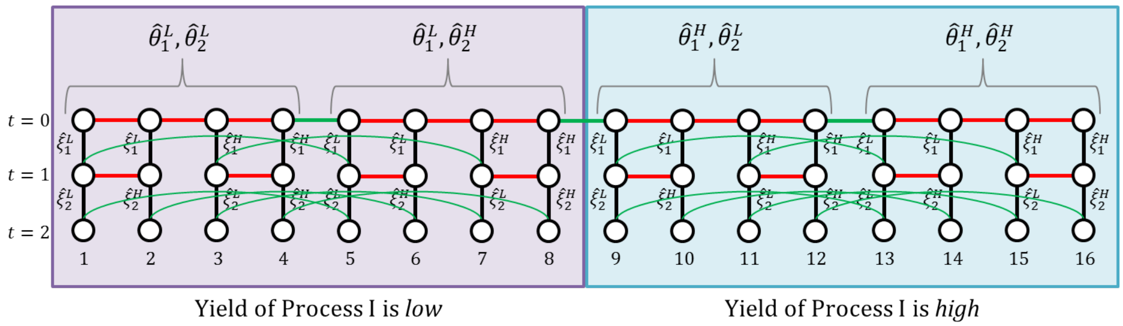
Note that unlike the two-stage case, the value of the stochastic solution (VSS) is not a trivial calculation for multistage problems. We do not perform this calculation here; however, we refer the reader to Escudero et al. (2007) and Maggioni et al. (2014) for further information on this topic.

5.3.2 Example 1: Capacity Expansion of a Process Network

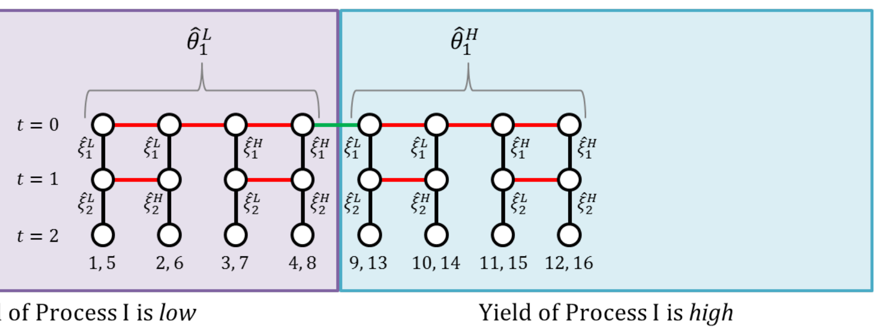
We now consider a larger instance of the motivating example. Specifically, we extend the time horizon to 8 years and consider 2 possible realizations for the demand of product A in each period, as well as 3 possible realizations for the yield of Process I and Process II. This gives 9 possible combinations of realizations for the endogenous parameters, and 256 possible combinations of realizations for the exogenous parameters. The result is a 9-stage stochastic programming problem with 2,304 scenarios. The corresponding model statistics for the fullspace model, the reduced model, and the SSD and LD problems are provided in Table 5.3. Notice in particular that the fullspace model for this instance has more than *176 million* constraints and approximately *4.8 million* binary variables. Such a model is clearly intractable in its current state. With the application of the reduction properties, we are able to reduce the number of constraints to about 838,000 — a *99.5% reduction*. The number of binary variables is also reduced to about 61,000, which is a 98.7% reduction.

Because of the longer time horizon, there are now more subproblems required for the sequential scenario decomposition heuristic. Each of these problems is significantly larger than the *one* in the motivating example; however, the model growth is slightly non-intuitive. Specifically, note the decrease in the problem size in the second subproblem, SSD (Sub 2). The reason for this is as follows.

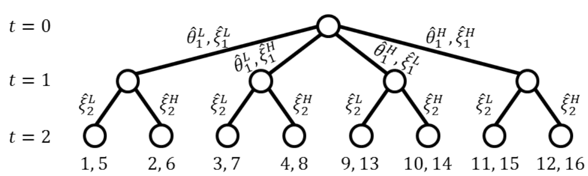
At $\hat{t} = 1$, there are 9 subtrees, each containing 2 exogenous scenario groups. We select *one*



(a) Optimal structure of the composite scenario tree. Notice the symmetry: the yield of Process I is *low* on the left and *high* on the right.



(b) Since the yield of Process II (θ_2) is unrealized during the time horizon, we can merge indistinguishable scenarios based on active NACs.



(c) Noting that the yield of Process I (θ_1) is realized in the first time period, we may now recover the standard form of the scenario tree.

Figure 5.6: Optimal structure of the composite scenario tree for the motivating example, and the procedure for converting this tree into its equivalent standard form.

Table 5.3: Model statistics for Example 1.

Problem Type	Scenarios	Constraints	Continuous Variables	Binary Variables
Fullspace	2,304	176,203,009	518,401	4,755,456
Reduced Model	2,304	838,318	518,401	61,416
SSD (Sub 1)	18	9,196	4,051	624
SSD (Sub 2)	18	8,569	4,051	492
SSD (Sub 3)	36	15,943	8,101	828
SSD (Sub 4)	72	29,395	16,201	1,344
SSD (Sub 5)	144	53,611	32,401	2,064
SSD (Sub 6)	288	96,475	64,801	2,880
SSD (Sub 7)	576	170,683	129,601	3,264
SSD (Final)	2,304	771,595	518,401	6,120
LD	256	78,862	57,601	6,144

scenario from each of these groups. In other words, we consider 18 scenarios in the first subproblem, and at this point, no binary here-and-now decisions have been fixed. In the second subproblem, $\hat{t} = 2$, we first consider 36 scenarios (i.e., 9 subtrees, each containing 4 exogenous scenario groups). The binary decisions have already been fixed in 18 of these scenarios. Accordingly, we neglect these 18 scenarios and consider only the remaining 18. Notice that this is *the same number of scenarios* as the first subproblem (see Appendix C.3 for further details); however, the binary here-and-now decisions for the previous time periods have already been fixed. This means that there will be fewer constraints and binary variables, as can be seen in Table 5.3.

In the third subproblem, $\hat{t} = 3$, we first consider 72 scenarios. The binary decisions have already been fixed in 36 of them, so we consider only the remaining 36. Notice that at this point, the number of scenarios in each subproblem begins to double (see Appendix C.3). The problem size, however, does *not* double. This can be seen in the corresponding number of binary variables reported in Table 5.3. Since in each subproblem the binary decisions in all previous time periods have already been fixed, we are able to effectively slow the problem growth.

We emphasize that the SSD subproblems are significantly smaller than the reduced model. At most, we consider 576 scenarios in subproblem 7. This is only 25% of the total number of scenarios. Moreover, this particular subproblem contains only 20% of the constraints of the reduced model and 5% of the binary variables.

In the Lagrangean decomposition algorithm, we again decompose the problem by subtrees (rather than by individual scenarios). This gives 9 subproblems of 256 scenarios each. Like for the SSD heuristic, these subproblems are considerably larger than those in the motivating example.

We solve this problem instance in GAMS 24.3.3, with CPLEX 12.6.0.1, on a machine with a 2.50 GHz Intel Core i5 CPU and 4 GB of RAM. The results are summarized in Table 5.4.

As would be expected, the fullspace model cannot be loaded into memory. After applying the reduction properties, however, we are in fact able to solve this instance to a 0.99% optimality gap in about 1 hour. The best feasible solution obtained from the reduced model is $142.411 MM.

As shown in Table 5.4, the SSD heuristic provides the same feasible solution as the reduced

Table 5.4: Numerical results for Example 1.

Problem Type	Total Expected Profit ($MM)		Optimality Gap	Solution Time (s)
	Lower Bound	Upper Bound		
Fullspace	–	–	–	–
Reduced Model	142.411	143.828	0.99%	3,670
SSD	142.411	–	–	61
LD	142.411	144.424	1.41%	913

model in just 61 seconds. We use this value as the initial lower bound for the Lagrangean decomposition algorithm. After 20 iterations, the lower bound does not improve, and we obtain an upper bound of $144.424 MM. This then provides us with bounds on the optimal solution; specifically, within a 1.41% optimality gap. The total time for lower- and upper-bound generation for the alternative solution methods is 974 seconds — a 73% reduction from solving the reduced model directly.

5.3.3 Example 2: Oilfield Development Planning

We consider a modified form of the MILP described in Gupta and Grossmann (2014a) (see Case (i)) for maximizing the total expected NPV in the development planning of an offshore oilfield. There are 3 oilfields; 3 potential Floating Production, Storage, and Offloading vessels (FPSOs); and 9 possible field-FPSO connections. A total of 30 wells can be drilled over a 5-year planning horizon: 7 for field I, 11 for field II, and 12 for field III. There is also a 3-year lead time for FPSO construction and a 1-year lead time for FPSO expansion. Fields II and III have a known recoverable oil volume (size); however, the size of field I is uncertain. Specifically, there are 2 possible realizations for the size of field I, both with equal probabilities. The oil and gas prices are also uncertain, with 2 possible realizations with equal probabilities in each time period. These prices are assumed to be correlated. The network superstructure for this problem instance is shown in Figure 5.7.

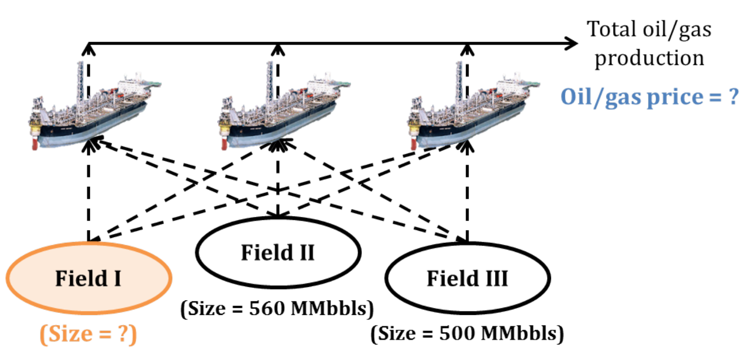


Figure 5.7: Network superstructure for the oilfield development planning problem. (FPSO images from www.rigzone.com.)

Notice that the size of field I is an endogenous parameter, since this information cannot be realized until we drill the field and begin producing from it. The oil and gas prices are exogenous

Table 5.5: Model statistics for Example 2.

Problem Type	Scenarios	Constraints	Continuous Variables	Binary Variables
Fullspace	64	303,553	67,393	9,024
Reduced Model	64	122,849	67,393	8,912
SSD (Sub 1)	4	7,531	4,213	558
SSD (Sub 2)	4	7,357	4,213	474
SSD (Sub 3)	8	14,539	8,425	900
SSD (Sub 4)	16	28,687	16,849	1,704
SSD (Final)	64	116,603	67,393	6,416
LD	2	3,725	2,107	278

Table 5.6: Numerical results for Example 2.

Problem Type	Total Expected NPV ($\$10^9$)		Optimality Gap	Solution Time (s)
	Lower Bound	Upper Bound		
Reduced Model	6.997	10.569	51.06%	40,818
SSD	7.166	–	–	41
LD	7.166	7.180	0.20%	14

parameters, as they are market values that will be realized automatically in each time period. We have 2 possible combinations of realizations for the endogenous parameters and 32 possible combinations of realizations for the exogenous parameters. Using the scenario-generation procedure described in Chapter 2, this gives rise to a 6-stage, 64-scenario stochastic programming problem. The corresponding model statistics are shown in Table 5.5. In particular, notice that the fullspace model consists of 303,553 constraints and 9,024 binary variables. After applying the theoretical reduction properties, there are 122,849 constraints and 8,912 binary variables. This is a 60% reduction in the number of constraints. The number of binary variables is reduced by approximately 1%.

Model statistics for the heuristic and Lagrangean decomposition are also provided in Table 5.5. Note that for the Lagrangean decomposition algorithm, we again choose not to decompose the problem by individual scenarios. However, rather than decomposing by subtrees, as this leads to very difficult subproblems, we instead consider 32 subproblems of 2 adjacent scenarios each.

The problem was modeled in GAMS 24.3.3 and solved with CPLEX 12.6.0.1 on a machine with a 2.93 GHz Intel Core i7 CPU and 12 GB of RAM. Table 5.6 summarizes the results for the different solution approaches. In the case of solving the reduced model directly, the optimality gap cannot be improved past 51% after more than 11 hours. In contrast, the sequential scenario decomposition heuristic finds a high-quality feasible solution ($7.166 billion) in only *41 seconds*. We initialize the lower bound of the Lagrangean decomposition algorithm to this objective value. After only 14 seconds, the LD algorithm finds a high-quality upper bound ($7.180 billion); the lower bound does not improve. This implies that the SSD solution is within 0.20% of the optimum. Notice that we obtain this information in *less than one minute* of CPU time.

The network structure corresponding to the best feasible solution ($7.166 billion, as obtained by the SSD heuristic) is shown in Figure 5.8. This solution indicates that we begin installing the necessary infrastructure in the first year. This includes FPSO I and FPSO II, as well as 3 of the 9 possible field-FPSO connections: field I to FPSO I, field II to FPSO I, and field III to FPSO II. Notice that due to the inherent risk in the size of field I, FPSO I is shared among fields I and II rather than devoting a separate FPSO solely to field I.

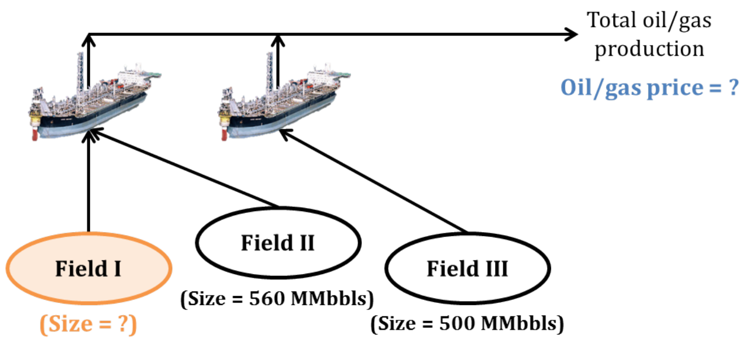


Figure 5.8: Network structure for the best feasible solution of Example 2. (FPSO images from www.rigzone.com.)

The corresponding drilling schedule is shown in Figure 5.9. Since it takes 3 years for the FPSOs to be fully operational, drilling cannot begin until the fourth year. For Field II, we drill 10 wells in year 4 and 1 well in year 5. Similarly for Field III, we drill 10 wells in year 4 and 2 wells in year 5. For Field I, however, we wait until year 5 and then drill 7 wells. The strategy here is to drill fields of known size first (as this carries less risk), and then drill the field with an uncertain size. Notice that by the end of the planning horizon, we have drilled the maximum number of wells in all fields.

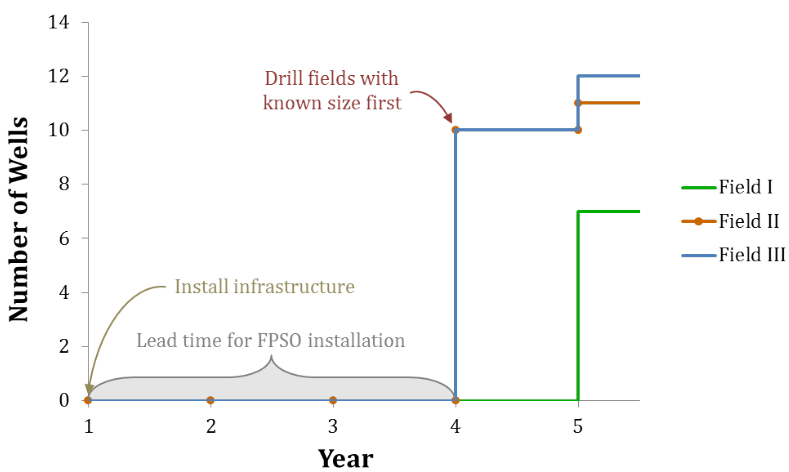


Figure 5.9: Drilling schedule for the best feasible solution of Example 2.

5.4 Conclusions

In this chapter, we have proposed two solution methods for multistage stochastic programs with endogenous and exogenous uncertainties: a novel sequential scenario decomposition heuristic and Lagrangean decomposition. We have evaluated the performance of these approaches, as well as the impact of our theoretical reduction properties from Chapter 4, on a process network planning problem and an oilfield development planning problem. Overall, we have demonstrated orders-of-magnitude reduction in problem size and solution times as a direct result of this work.

Chapter 6

A Graph-Theory Approach for NAC Reduction

6.1 Introduction

In multistage stochastic programming, a well-known modeling challenge is that the number of non-anticipativity constraints (NACs) increases rapidly as one increases the number of stages or the number of scenarios (Birge and Louveaux, 2011). In fact, as we have shown in Chapter 5, it is not uncommon for NACs to represent the majority of constraints in large problem instances. It should not be surprising, then, that some authors have invested a considerable amount of effort into developing new formulations that eliminate redundant NACs. In the case that the uncertainty is purely exogenous, this process is fairly straightforward since the scenario tree is fixed. One common strategy for such problems is the NAC aggregation approach proposed by Birge and Louveaux (2011).

For the case of (Type 2) endogenous uncertainties, however, the scenario tree is decision dependent; hence, NAC elimination is considerably more involved. Previous works in the literature, namely Goel and Grossmann (2006), Gupta and Grossmann (2011), and Chapter 4 of this thesis, have proposed effective theoretical reduction properties for this class of problems. Colvin and Maravelias (2008, 2009, 2010) have also proposed a number of reduction properties for the pharmaceutical clinical-trial scheduling problem. One primary assumption in these existing properties, though, is that the set of scenarios corresponds to *all* possible combinations of realizations of the uncertain parameters (i.e., a Cartesian product over all sets of realizations). In large, real-world problems with many uncertain parameters, the Cartesian-product approach can easily lead to instances with several thousands of scenarios or more. These problems are generally intractable. Furthermore, if some of the uncertain parameters are correlated, many of the scenarios in these large problems may in fact refer to infeasible outcomes. For example, based on geological conditions, it may be extremely unlikely for two nearby oilfields to have vastly different oil recoveries. A scenario with a low realization for one field and a high realization for the other would therefore represent a physically infeasible outcome that should not be included in the model.

In this chapter, we consider mixed-integer linear multistage stochastic programming problems involving endogenous and exogenous uncertainties, where the scenario sets are pre-specified and do not necessarily correspond to Cartesian products. Since the previous reduction properties in the literature do not apply here, efficient generation of the non-anticipativity constraints becomes a challenge. Boland et al. (2008) was the first work (of which we are aware) to raise these concerns. The concepts in that paper were further developed in Boland et al. (2016), which introduced the idea of a *non-anticipativity graph* and proposed a greedy, polynomial-time algorithm for generating minimum-cardinality NAC sets. Similar work was published by Hooshmand Khaligh and MirHassani (2016b) around the same time. It is also worth noting the more recent strategy by Christian and Cremaschi (2016), referred to as the Sample Non-Anticipativity Constraint algorithm, that claims to generate the minimum number of NACs by mapping the set of scenarios to an integer lattice and progressively adding edges (i.e., NACs) until the lattice forms a minimum spanning tree. This particular work was inspired by Apap and Grossmann (2015), and it is unclear how it differs from existing publications in the area.

In the current chapter, we extend the work of Boland et al. (2016) and Hooshmand Khaligh and MirHassani (2016b) to problems with both endogenous and exogenous uncertainties. We begin in Section 6.2 with a more general multistage stochastic programming model, and in Section 6.3, we discuss how our existing reduction properties change in the absence of Cartesian products. We introduce the concept of the non-anticipativity graph in Section 6.4. Finally, in Section 6.5, we present the graph algorithm for scenario-pair generation.

6.2 A More General Multistage Stochastic Programming Model

In the absence of Cartesian products, we can no longer guarantee that we will have a predictably-structured scenario tree like Figure 2.1. Our tree may, instead, look like Figure 6.1. To handle such cases, we propose a more general form of our multistage stochastic programming model (MSSP), which we will refer to as (MSSPG).

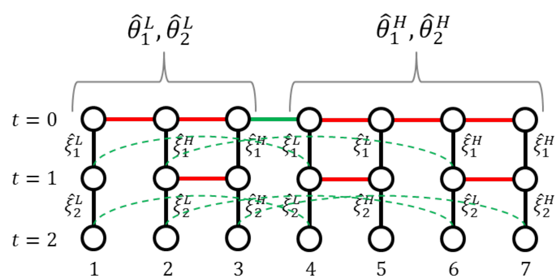


Figure 6.1: A composite scenario tree generated from an arbitrary scenario set.

(MSSPG)

$$\min_{b,y,x} \phi = \sum_{s\in\mathcal{S}} p^s \sum_{t\in\mathcal{T}} \left({}^y c_t^s y_t^s + {}^x c_t^s x_t^s + {}^w c_t^s w_t^s + \sum_{i\in\mathcal{I}} {}^b c_{i,t}^s b_{i,t}^s \right) \tag{6.1}$$

$$\text{s.t.} \quad \sum_{\tau=1}^{t} \left({}^y A_{\tau,t}^s y_\tau^s + {}^x A_{\tau,t}^s x_\tau^s + {}^w A_{\tau,t}^s w_\tau^s + \sum_{i\in\mathcal{I}} {}^b A_{i,\tau,t}^s b_{i,\tau}^s \right) \le a_t^s \qquad \forall\, t\in\mathcal{T},\ \ s\in\mathcal{S} \tag{6.2}$$

$$b^s_{i,1} = b^{s'}_{i,1} \qquad \forall\,(s,s') \in \mathcal{SP}_F,\;\; i \in \mathcal{I} \tag{6.3}$$
$$y^s_1 = y^{s'}_1 \qquad \forall\,(s,s') \in \mathcal{SP}_F \tag{6.4}$$
$$x^s_t = x^{s'}_t \qquad \forall\,(t,s,s') \in \mathcal{SP}_X \tag{6.5}$$
$$b^s_{i,t+1} = b^{s'}_{i,t+1} \qquad \forall\,(t,s,s') \in \mathcal{SP}_X,\;\; i \in \mathcal{I} \tag{6.6}$$
$$y^s_{t+1} = y^{s'}_{t+1} \qquad \forall\,(t,s,s') \in \mathcal{SP}_X \tag{6.7}$$
$$x^s_t = x^{s'}_t \qquad \forall\,(t,s,s') \in \mathcal{SP}_N,\;\; t \in \mathcal{T}^{i'}_E,\;\; \{i'\} = \hat{\mathcal{D}}^{s,s'}_{\min} \tag{6.8}$$
$$b^s_{i,t+1} = b^{s'}_{i,t+1} \qquad \forall\,(t,s,s') \in \mathcal{SP}_N,\;\; t \in \mathcal{T}^{i'}_E,\;\; \{i'\} = \hat{\mathcal{D}}^{s,s'}_{\min},\;\; i \in \mathcal{I} \tag{6.9}$$
$$y^s_{t+1} = y^{s'}_{t+1} \qquad \forall\,(t,s,s') \in \mathcal{SP}_N,\;\; t \in \mathcal{T}^{i'}_E,\;\; \{i'\} = \hat{\mathcal{D}}^{s,s'}_{\min} \tag{6.10}$$
$$\begin{bmatrix} \quad Z^{s,s'}_t & \\ x^s_t = x^{s'}_t & \\ b^s_{i,t+1} = b^{s'}_{i,t+1}\; \forall\, i \in \mathcal{I}, & t < T \\ y^s_{t+1} = y^{s'}_{t+1} & t < T \end{bmatrix} \vee \left[\neg Z^{s,s'}_t\right] \qquad \forall\,(t,s,s') \in \mathcal{SP}_N,\;\; t \in \mathcal{T}^{i'}_C,\;\; \{i'\} = \hat{\mathcal{D}}^{s,s'}_{\min} \tag{6.11}$$
$$Z^{s,s'}_t \Leftrightarrow F(b^s_{i'',1}, b^s_{i'',2}, \ldots, b^s_{i'',t}) \qquad \forall\,(t,s,s') \in \mathcal{SP}_N,\;\; t \in \mathcal{T}^{i'}_C,\;\; \{i'\} = \hat{\mathcal{D}}^{s,s'}_{\min},\;\; i'' \in \hat{\mathcal{D}}^{s,s'} \tag{6.12}$$
$$b^s_{i,t} \in \{0,1\},\;\; y^s_t \in \mathcal{Y}^s_t,\;\; x^s_t \in \mathcal{X}^s_t,\;\; w^s_t \in \mathcal{W}^s_t \qquad \forall\, i \in \mathcal{I},\;\; t \in \mathcal{T},\;\; s \in \mathcal{S} \tag{6.13}$$
$$Z^{s,s'}_t \in \{True, False\} \qquad \forall\,(t,s,s') \in \mathcal{SP}_N,\;\; t \in \mathcal{T}^{i'}_C,\;\; \{i'\} = \hat{\mathcal{D}}^{s,s'}_{\min} \tag{6.14}$$

Notice that only fairly simple changes are required to obtain this model. First and most simply, in the indistinguishability rule, Equation (6.12), we must now consider binary investment decisions for all $i'' \in \hat{\mathcal{D}}^{s,s'}$. This is because, in general, the scenario pairs under consideration may differ in many endogenous parameters corresponding to several different sources. We note that in the original form of this equation, Equation (3.19), this appears as $\{i'\} = \hat{\mathcal{D}}^{s,s'}$, which shares the same index i' as $\mathcal{T}^{i'}_C$.

This then brings us to the discussion of the second set of changes. In the previous chapters in this thesis, we consider the case where scenarios s and s' differ only in the possible realization of *one* endogenous parameter. There, for the particular source i' of sets $\mathcal{T}^{i'}_E$ and $\mathcal{T}^{i'}_C$, we simply have $\{i'\} = \hat{\mathcal{D}}^{s,s'}$. In the more general case considered here, however, it is not as straightforward to identify the source i'. If two scenarios differ in the possible realizations of *multiple* endogenous parameters associated with the *same* source, we will still have $\{i'\} = \hat{\mathcal{D}}^{s,s'}$. But if the scenarios differ in the possible realizations of multiple endogenous parameters associated with *different* sources, we can only guarantee that the scenarios will be indistinguishable for the *minimum* number of initial 'equality' time periods $T^{i'}_E$ among those respective sources. For convenience, we will refer to $T^{i'}_E$ as the *lead time* for source $i' \in \mathcal{I}$.

For example, if s and s' differ in two sources, one with a lead time of 3 years and another with a lead time of 5 years, we can only be certain that the scenarios will be indistinguishable for the first 3 years. We will refer to this minimum lead time as $T^{s,s'}_{E_{\min}}$. To define this parameter for each pair (s,s'), we construct a sequence of the lead times corresponding to the sources associated with scenarios s and s', and then determine the minimum value in that sequence:

$$T_{E_{\min}}^{s,s'} := \min\,(T_E^{i'})_{i' \in \hat{\mathcal{D}}^{s,s'}} \qquad \forall\, s, s' \in \mathcal{S},\;\; s < s' \tag{6.15}$$

We use this information to define set $\hat{\mathcal{D}}_{\min}^{s,s'}$, which indicates the particular source i' that has the minimum lead time; i.e., the $i' \in \hat{\mathcal{D}}^{s,s'}$ for which $T_E^{i'} = T_{E_{\min}}^{s,s'}$. Since more than one source may satisfy this condition, we simply select the lowest-indexed source that has the minimum lead time:

$$\hat{\mathcal{D}}_{\min}^{s,s'} := \left\{ i' : \, i' = \min_{\hat{\imath}'} \left(\hat{\imath}' \in \hat{\mathcal{D}}^{s,s'},\; T_E^{\hat{\imath}'} = T_{E_{\min}}^{s,s'} \right) \right\} \qquad \forall\, s, s' \in \mathcal{S},\;\; s < s' \tag{6.16}$$

This then allows us to refer to sets $\mathcal{T}_E^{i'}$ and $\mathcal{T}_C^{i'}$ with $\{i'\} = \hat{\mathcal{D}}_{\min}^{s,s'}$ (rather than $\{i'\} = \hat{\mathcal{D}}^{s,s'}$). Note that this statement applies regardless of whether the two scenarios differ in the possible realizations of endogenous parameters associated with the same source *or* different sources. We make this replacement in Equations (3.15)–(3.19) and (3.21) to produce Equations (6.8)–(6.12) and (6.14), respectively. All other changes to model (MSSP) are concealed in scenario-pair sets $\mathcal{SP}_X$ and $\mathcal{SP}_N$, which we will discuss in the next section.

One final note here is that we must redefine sets $\mathcal{D}^{s,s'}$ to indicate the endogenous parameters $\theta_{i',h}$ for which scenarios s and s' differ in possible realizations:

$$\mathcal{D}^{s,s'} := \left\{ (i', h) : \, i' \in \mathcal{I},\;\; h \in \mathcal{H}_{i'},\;\; \theta_{i',h}^{s} \neq \theta_{i',h}^{s'} \right\} \qquad \forall\, s, s' \in \mathcal{S},\;\; s < s' \tag{6.17}$$

as well as sets $\hat{\mathcal{D}}^{s,s'}$ to indicate the associated *sources*, i':

$$\hat{\mathcal{D}}^{s,s'} := \left\{ i' : \, i' \in \mathcal{I},\;\; \left(\exists\, h \in \mathcal{H}_{i'} : \, (i', h) \in \mathcal{D}^{s,s'} \right) \right\} \qquad \forall\, s, s' \in \mathcal{S},\;\; s < s' \tag{6.18}$$

Note that the "position" condition $Pos(s) = Pos(s')$ present in the original definitions (Equations (4.9) and (4.21), respectively) has been removed in Equations (6.17) and (6.18) because, in general, the structure of each subtree may not be the same. We maintain the condition $s < s'$, however, to avoid needlessly doubling the size of these sets. We emphasize that these updated definitions are *required* for model (MSSPG), as well as for the definitions of $T_{E_{\min}}^{s,s'}$ and $\hat{\mathcal{D}}_{\min}^{s,s'}$.

6.3 Scenario Pairs and Reduction Properties in the Absence of Cartesian Products

When the scenario set is generated by methods other than a Cartesian product, the definitions for the corresponding set of scenarios (i.e., $\mathcal{R}_X$, $\mathcal{R}_N$, and $\mathcal{R}$), the number of scenarios in these sets (i.e., S_X, S_N, and S), and the associated probabilities (i.e., $\mathcal{P}_X$, $\mathcal{P}_N$, and $\mathcal{P}$) no longer apply. These definitions can easily be substituted for those given by the particular scenario generation method being used. New definitions for our scenario-pair sets are not quite as trivial, however.

First, we note that Property 1, Property 2a, and Property 2b from Chapter 4 apply regardless of the structure of the scenario set. For the case of Property 2b, one caveat is that scenarios must first be sorted with lexicographical ordering (as described in Chapter 2) and then renumbered if they are not already indexed in sequential order. Property 5 also still applies, and the proof for this property (originally presented for the Cartesian product case in Appendix B.8) can easily be extended to the general case considered here.

It follows from this discussion that set $\mathcal{SP}_F$ for first-period scenario pairs is valid in its original form (Equation (3.10)). For exogenous scenario-pair set $\mathcal{SP}_X$, however, some modifications are required. First, we redefine Boolean parameter $Q_t^{s,s'}$ as follows:

$$Q_1^{s,s'} := \begin{cases} True, & \text{if } \xi_{j,1}^{s} = \xi_{j,1}^{s'} \ \forall\, j \in \mathcal{J} \\ False, & \text{otherwise} \end{cases} \qquad \forall\, s, s' \in \mathcal{S},\ \ s < s' \tag{6.19}$$

$$Q_t^{s,s'} := \begin{cases} True, & \text{if } Q_{t-1}^{s,s'} = True \ \text{ and } \ \xi_{j,t}^{s} = \xi_{j,t}^{s'} \ \forall\, j \in \mathcal{J} \\ False, & \text{otherwise} \end{cases}$$
$$t = 2, 3, \ldots, T-1,\ \ \forall\, s, s' \in \mathcal{S},\ \ s < s' \tag{6.20}$$

Note that we have replaced the conditions $(s, s') \in \mathcal{A},\ Sub(s) = Sub(s')$ with $s, s' \in \mathcal{S},\ s < s'$. There are two reasons for this. First, the subtree condition $Sub(s) = Sub(s')$ no longer applies because the original definition, Equation (2.7), is based on the assumption that there are S_X scenarios in every subtree. In the general case, each subtree may in fact contain a different number of scenarios. Second, when dealing with endogenous scenario pairs, we had previously used the condition $Pos(s) = Pos(s')$ to ensure that s and s' were identical in the realizations of all exogenous parameters. However, because each subtree may have a different structure, this is no longer valid and we must instead explicitly verify exogenous indistinguishability. This would not be possible if we allow only adjacent scenarios (i.e., $(s, s') \in \mathcal{A}$), so we must also relax this condition, ultimately leaving us with $s, s' \in \mathcal{S},\ s < s'$.

In a similar manner, we restate the definition of set $\mathcal{SP}_X$ as:

$$\mathcal{SP}_X := \Big\{(t, s, s') :\ t \in \mathcal{T} \setminus \{T\},\ \ (s, s') \in \mathcal{A},\ \ |\mathcal{D}^{s,s'}| = 0,\ \ Q_t^{s,s'} = True\Big\} \tag{6.21}$$

where we have replaced the $Sub(s) = Sub(s')$ condition in Equation (3.11) with $|\mathcal{D}^{s,s'}| = 0$. This simply ensures that the scenarios have the same possible realizations for all endogenous parameters, which implies that they are in the same subtree.

Unlike the definition of set $\mathcal{SP}_X$, we cannot simply modify conditions in the original definition of set $\mathcal{SP}_N$ to obtain a set that is valid in the general case. The fundamental problem here is described in the following proposition.

Proposition 6. *Property 3* (Goel and Grossmann, 2006) *states that for endogenous NACs, it is sufficient to consider only scenario pairs (s, s') for which s and s' differ in the possible realization of a single endogenous parameter and are identical in the realizations of all exogenous parameters in all time periods. However, this property cannot be applied in the case where set $\mathcal{R}_N$ does not correspond to a scenario tree constructed from all possible combinations of realizations of the endogenous parameters.*

Proof. This can be shown by a simple counter example. First, assume that Property 3 applies and consider scenarios $\hat{s}$ and $\hat{s}'$ in an arbitrary time period $t = \tau$ in Figure 6.2. These scenarios differ in the possible realizations of two endogenous parameters, θ_1 and θ_2. Since the realization of either parameter will distinguish the scenarios, non-anticipativity applies only in the case where neither parameter is realized (Case 2).

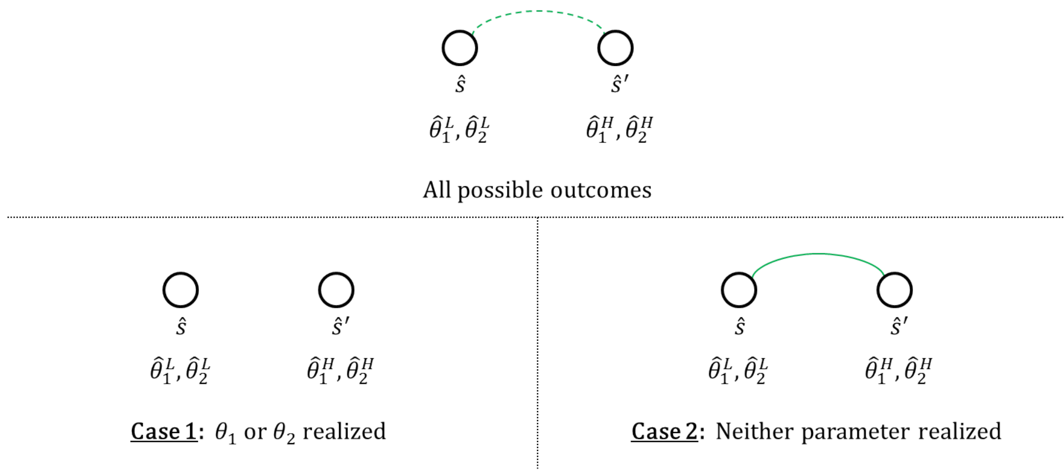


Figure 6.2: Required non-anticipativity constraints between 2 scenarios that differ in the possible realizations of 2 endogenous parameters.

Further assume that these are the only two scenarios in the corresponding problem instance. It is then clear that non-anticipativity must be explicitly enforced between scenarios $\hat{s}$ and $\hat{s}'$ in Case 2, since there are no other constraints that can be used to imply these NACs. Recall, however, that the scenarios differ in the possible realizations of *two* endogenous parameters (i.e., $|\mathcal{D}^{\hat{s},\hat{s}'}| = 2$), which violates Property 3. Scenario pair $(\hat{s}, \hat{s}')$ would thus be removed, rendering Case 2 unenforceable. It follows that Property 3 must be invalid in the current context. Note that the same argument can be made for the endogenous and exogenous case, as can be seen in Figure 6.1 with a composite scenario tree for which none of the conditional constraints can be enforced if Property 3 is applied. □

This limitation has a direct influence on some of our other reduction properties, as stated in Proposition 7.

Proposition 7. *Properties 4 and 6, which were introduced in Chapter 4*[1] *and can remove additional redundant scenario pairs by exploiting endogenous scenario groups, also do not apply in the current case.*

Proof. Property 3 is a necessary condition for Properties 4 and 6 (see Chapter 4). Thus, it follows directly from Proposition 6 that these properties do not apply here. □

Due to Propositions 6 and 7, it is clear that special considerations are required to eliminate redundant endogenous scenario pairs in the absence of these reduction properties. As the first step to this end, we consider a graph representation for these scenario pairs.

[1] The basis for Property 4 was originally proposed in Gupta and Grossmann (2011).

6.4 Non-anticipativity Graph for Endogenous Scenario Pairs

To represent the endogenous NACs, we adopt the use of a *non-anticipativity graph*, as shown in Figure 6.3. This is an undirected graph $\mathbb{G} := (\mathcal{V}, \mathcal{E})$ with vertex set $\mathcal{V}$ and edge set $\mathcal{E}$. Here, the vertices are the scenarios s and the edges are the scenario pairs (s, s'). Although this idea has recently been proposed by both Boland et al. (2016) and Hooshmand Khaligh and MirHassani (2016b), for convenience, we will continue this discussion in terms that more closely resemble the latter publication.

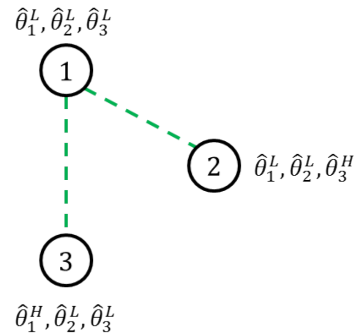


Figure 6.3: A simple non-anticipativity graph.

To fully define the non-anticipativity graph, we must ensure that, in each time period, there is at least one valid path between all vertices s and s' that differ in the possible realizations of one or more endogenous parameters *and* are identical in the realizations of all exogenous parameters that have been realized up until the end of that period. The reason for this is that if two scenarios are indistinguishable at the end of time period t, we must have a connection between the two in order to enforce the same recourse decisions in both scenarios at the end of this period and the same here-and-now decisions at the beginning of the following period. The goal is to limit the number of these connections to the minimum number required. Note that if we were to instead naively insert a new edge into the graph between every pair of vertices, we would likely end up with a considerable amount of redundancy, which we wish to avoid.

Let $P^{s,s'}$ denote a path between vertices s and s', where $s \neq s'$. Specifically, we define $P^{s,s'}$ as a sequence of vertices, such as $P^{2,3} = (2, 1, 3)$ in the case of Figure 6.3. Note that in this particular walk, the edges (i.e., arcs) are $(2, 1)$ and $(1, 3)$.

Since some of the edges in an endogenous non-anticipativity graph are *conditional*, we must ensure that all edges in $P^{s,s'}$ exist in cases where non-anticipativity applies between s and s'. Thus, in order to be a valid path, $P^{s,s'}$ must consist only of:

1. *fixed* edges (s'', s'''), for which the uncertainty in s'' and s''' cannot be resolved in the current time period; i.e., $t \in \mathcal{T}_E^{i'}$, where $\{i'\} = \hat{\mathcal{D}}_{\min}^{s'',s'''}$; and/or

2. *conditional* edges (s'', s'''), for which the uncertainty in s'' and s''' may be resolved in the current time period and s'' and s''' differ *exclusively* in possible realizations of endogenous parameters associated with *the same sources* as s and s'; i.e., $t \in \mathcal{T}_C^{i'}$, where $\{i'\} = \hat{\mathcal{D}}_{\min}^{s'',s'''}$,

and $\hat{\mathcal{D}}^{s'',s'''} \subseteq \hat{\mathcal{D}}^{s,s'}$.[2]

In both cases, s'' and s''' must be identical in the realizations of all exogenous parameters that have been realized up until the end of the current time period; i.e., $Q_t^{s'',s'''} = True$. Note that the subset condition $\hat{\mathcal{D}}^{s'',s'''} \subseteq \hat{\mathcal{D}}^{s,s'}$ has been adopted from Hooshmand Khaligh and MirHassani (2016b).

It is also worth noting that it is unnecessary to explicitly check the time period (i.e., $t \in \mathcal{T}_C^{i'}$, $\{i'\} = \hat{\mathcal{D}}_{\min}^{s'',s'''}$) for conditional edges, since if $t \notin \mathcal{T}_C^{i'}$, we have $t \in \mathcal{T}_E^{i'}$, in which case all edges are permissible. Thus, in summary, each edge (s'', s''') in a valid path $P^{s,s'}$ must satisfy the condition $Q_t^{s'',s'''} = True$, as well as $(t \in \mathcal{T}_E^{i'},\ \{i'\} = \hat{\mathcal{D}}_{\min}^{s'',s'''})$ *or* $(\hat{\mathcal{D}}^{s'',s'''} \subseteq \hat{\mathcal{D}}^{s,s'})$.

To further clarify some of these complex requirements, consider Figure 6.4. Here, we have the 3 scenarios from Figure 6.3 in an arbitrary time period $t = \tau$. There are 3 endogenous parameters (θ_1, θ_2, and θ_3, each associated with a different source), and 2 possible realizations for each parameter (*low* (L) or *high* (H)). All edges are conditional. In forming a path between vertices 2 and 3, we must first consider that these scenarios differ in the possible realizations of θ_1 and θ_3; i.e., $\hat{\mathcal{D}}^{2,3} = \{1, 3\}$. This means that any edge in a path between vertices 2 and 3 can *only* consist of scenarios (s'', s''') for which s'' and s''' differ in the possible realization of θ_1, θ_3, or both. The reasoning here is that the path exists only in the case that both of these parameters are unrealized, so edges exclusively associated with either of these parameters will be sufficient. (Note that if there happened to be an edge associated with θ_2, and we attempted to use it in our path, it would disappear upon the realization of θ_2, thereby leaving us with an incomplete path.) Accordingly, notice that edge $(1, 2)$ satisfies this requirement since $\hat{\mathcal{D}}^{1,2} = \{3\}$ and $\{3\} \subseteq \{1, 3\}$. The same is true for edge $(1, 3)$, since $\hat{\mathcal{D}}^{1,3} = \{1\}$ and $\{1\} \subseteq \{1, 3\}$. The discussion is slightly more complex if there is also exogenous uncertainty in the example; however, the basic ideas remain the same.

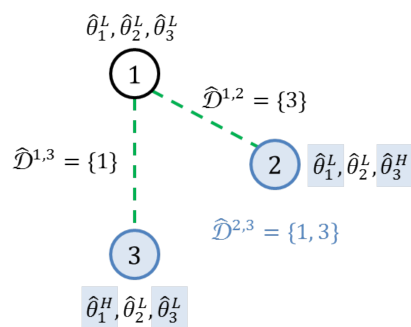


Figure 6.4: Characteristics of a valid path between vertices 2 and 3 from Figure 6.3.

Note that because we do not consider endogenous scenario groups in this chapter (unlike Chapter 4), for convenience, we will primarily consider differences between scenarios at the *source* level rather than at the *parameter* level. In other words, we will rely much more heavily on sets $\hat{\mathcal{D}}^{s,s'}$

[2] Note that the suitability of each edge in a path is verified in the context of an *undirected* form of the edge. Specifically, we store each edge as (s'', s'''), where $s'' < s'''$, with the understanding that it can be traversed from s'' to s''' or s''' to s''. For example, in Figure 6.3, we may travel from vertex 2 to vertex 1 (i.e., directed edge $(2, 1)$), but we store this edge as $(1, 2)$. This practice prevents the duplication of information and ensures that we only have to check that $\hat{\mathcal{D}}^{s'',s'''} \subseteq \hat{\mathcal{D}}^{s,s'}$, rather than also checking whether $\hat{\mathcal{D}}^{s''',s''} \subseteq \hat{\mathcal{D}}^{s,s'}$.

than $\mathcal{D}^{s,s'}$. This allows us to naturally account for multiple endogenous parameters associated with each source of uncertainty, since any parameters associated with the same source can only be realized at the same time, and therefore any edges (s'', s''') for which s'' and s''' differ in the possible realizations of these parameters can potentially be used together in the same path.

6.5 Graph Algorithm for Scenario-pair Generation

6.5.1 General Algorithm

We now propose a general algorithm for scenario-pair generation in multistage stochastic programs with both endogenous and exogenous uncertainties. This is an extension of the polynomial-time algorithm for purely-endogenous uncertainty recently proposed by Hooshmand Khaligh and Mir-Hassani (2016b).

Graph Algorithm for Scenario-Pair Generation

Step 1 Generate first-period and exogenous scenario-pair sets $\mathcal{SP}_F$ and $\mathcal{SP}_X$, respectively, by Equations (3.10) and (6.21).

Step 2 Initialize the set of vertices $\mathcal{V}_t$ to the set of 'unique' scenarios obtained from Property 5 (i.e., $\mathcal{V}_t := \tilde{\mathcal{U}}_t$), initialize the set of edges $\mathcal{E}_t$ to the empty set (i.e., $\mathcal{E}_t := \emptyset$), and then consider the empty graph $\mathbb{G}_t := (\mathcal{V}_t, \mathcal{E}_t)$ for all $t \in \mathcal{T}$. For convenience, we will consider a separate non-anticipativity graph for each time period.

Step 3 Define the sets of all *potential* edges (s, s'), where s and s' differ in the possible realizations of endogenous parameters from η different sources (i.e., $|\hat{\mathcal{D}}^{s,s'}| = \eta$) and are identical in the realizations of all exogenous parameters that have been realized up until the end of the current time period (i.e., $Q_t^{s,s'} = True$). We will define two separate versions of these sets, ${}^{E}\mathcal{E}_t^{\eta}$ and ${}^{C}\mathcal{E}_t^{\eta}$, for edges corresponding to fixed endogenous NACs and conditional endogenous NACs, respectively:

$$
\begin{aligned}
{}^{E}\mathcal{E}_t^{\eta} := \Big\{ (s, s') :\; & s, s' \in \mathcal{S},\;\; s < s',\;\; |\hat{\mathcal{D}}^{s,s'}| = \eta,\;\; t \in \mathcal{T}_E^{i'},\;\; \{i'\} = \hat{\mathcal{D}}^{s,s'}_{\min}, \\
& Q_t^{s,s'} = True \Big\} \qquad \forall\, \eta \in \mathcal{I},\;\; t \in \mathcal{T}
\end{aligned}
\tag{6.22}
$$

$$
\begin{aligned}
{}^{C}\mathcal{E}_t^{\eta} := \Big\{ (s, s') :\; & s, s' \in \mathcal{S},\;\; s < s',\;\; |\hat{\mathcal{D}}^{s,s'}| = \eta,\;\; t \in \mathcal{T}_C^{i'},\;\; \{i'\} = \hat{\mathcal{D}}^{s,s'}_{\min}, \\
& Q_t^{s,s'} = True \Big\} \qquad \forall\, \eta \in \mathcal{I},\;\; t \in \mathcal{T}
\end{aligned}
\tag{6.23}
$$

For example, ${}^{E}\mathcal{E}_1^{2}$ would indicate the set of potential *fixed* edges (s, s') in the first time period that are associated with 2 different sources. (More precisely, this refers to vertices s and s' in the first time period that are identical in the realizations of all exogenous parameters in this period and differ in the possible realizations of endogenous parameters from 2 different sources for which the uncertainty cannot yet be resolved.) Set ${}^{C}\mathcal{E}_1^{2}$ would indicate a similar case for potential *conditional* edges. Notice that the only difference between the two definitions is the presence of either $t \in \mathcal{T}_E^{i'}$ or $t \in \mathcal{T}_C^{i'}$.

Step 4 For each time period $t \in \mathcal{T}$:

Step 4a For $\eta = 1, 2, \ldots, I$, consider each potential edge $(s, s') \in {}^{E}\mathcal{E}_t^{\eta}$:

(i) Check whether there exists a valid path between vertices s and s' in $\mathbb{G}_t$. Specifically:

(1) Generate a new, temporary graph $\tilde{\mathbb{G}}_t := (\tilde{\mathcal{V}}_t, \tilde{\mathcal{E}}_t)$, where set $\tilde{\mathcal{E}}_t$ contains all edges in $\mathcal{E}_t$ that could be used in a valid path between s and s':

$$\tilde{\mathcal{E}}_t := \left\{(s'', s''') : (s'', s''') \in \mathcal{E}_t, \ \left(t \in \mathcal{T}_E^{i'}, \ \{i'\} = \hat{\mathcal{D}}_{\min}^{s'',s'''}\right) \text{ or } \left(\hat{\mathcal{D}}^{s'',s'''} \subseteq \hat{\mathcal{D}}^{s,s'}\right)\right\} \tag{6.24}$$

Recall from Section 6.4 that a valid path includes all fixed edges (s'', s''') (i.e., $t \in \mathcal{T}_E^{i'}$, where $\{i'\} = \hat{\mathcal{D}}_{\min}^{s'',s'''}$) and any conditional edges (s'', s''') for which s'' and s''' differ exclusively in possible realizations of endogenous parameters associated with the same sources as s and s' (i.e., $\hat{\mathcal{D}}^{s'',s'''} \subseteq \hat{\mathcal{D}}^{s,s'}$). The condition $Q_t^{s'',s'''} = True$, which is also generally required, is implicit in Equation (6.22) due to its presence in the definition of ${}^{E}\mathcal{E}_t^{\eta}$.

Additionally, set $\tilde{\mathcal{V}}_t$ consists of only the vertices needed to construct the edges in set $\tilde{\mathcal{E}}_t$:

$$\tilde{\mathcal{V}}_t := \{s\} \cup \{s'\} \cup \left\{s'' : s'' \in \mathcal{S}, \ \left(\exists\, s''' \in \mathcal{S} : (s'', s''') \in \tilde{\mathcal{E}}_t \text{ or } (s''', s'') \in \tilde{\mathcal{E}}_t\right)\right\} \tag{6.25}$$

Notice that Equation (6.25) implies that $\tilde{\mathcal{V}}_t \subseteq \mathcal{V}_t$.[3]

(2) Use a breadth-first search to determine whether there is a path from s to s' in graph $\tilde{\mathbb{G}}_t$. If the algorithm returns $True$, a valid path already exists; if it returns $False$, no such path exists.

(ii) If no such path exists, add edge (s, s') to $\mathbb{G}_t$:

$$\mathcal{E}_t := \mathcal{E}_t \cup \{(s, s')\} \tag{6.26}$$

Step 4b Repeat Step 3a, and all sub-steps, with ${}^{E}\mathcal{E}_t^{\eta}$ replaced by ${}^{C}\mathcal{E}_t^{\eta}$.

Step 5 Use the final set of edges for each time period, $\mathcal{E}_t$, to construct the complete set of endogenous scenario pairs, $\mathcal{SP}_N$:

$$\mathcal{SP}_N := \{(t, s, s') : t \in \mathcal{T}, \ (s, s') \in \mathcal{E}_t\} \tag{6.27}$$

Notice that Step 4 of the algorithm is easily parallelizable since each time period is considered separately. Also, note that because we are simply interested in connectivity when adding new edges to the graph, we deviate from the approach of Hooshmand Khaligh and MirHassani (2016b) and use a breadth-first search instead of Dijkstra's algorithm. This has the advantage of a simpler implementation and a time complexity of only $O(|\tilde{\mathcal{V}}_t| + |\tilde{\mathcal{E}}_t|)$ for each $t \in \mathcal{T}$, as opposed to $O(|\tilde{\mathcal{V}}_t|^2 + |\tilde{\mathcal{E}}_t|)$ for the classical form of Dijkstra's algorithm.

[3] In practice, it may be simpler to let $\tilde{\mathcal{V}}_t := \mathcal{V}_t$, since any vertices $s \in \mathcal{V}_t \setminus \tilde{\mathcal{V}}_t$ will not be connected to any other vertices in $\tilde{\mathbb{G}}_t$ and will automatically be ignored by the algorithm in the next step.

We further note that the endogenous scenario pairs are generated in an "upward" fashion since we start from an empty graph and progressively add new edges. Boland et al. (2016) states that it is also possible to take a "downward" approach, in which one would start from a complete graph and progressively remove unnecessary edges.

6.5.2 Order of Steps

It is important to emphasize that fixed edges must be added to the graph before conditional edges. This is due to the fact that fixed edges can be used in *any* path and can thus decrease the number of conditional edges that must be added to the graph.

It is also crucial that we consider steps 4a and 4b of the algorithm in order of increasing η (i.e., increasing cardinality of $|\hat{\mathcal{D}}^{s,s'}|$). This is formally stated in the following proposition.

Proposition 8. *In the proposed graph algorithm for scenario-pair generation, steps 4a and 4b must be executed in order of increasing η to avoid adding unnecessary edges to the graph.*

Proof. This can be shown in part by a simple counter example. First, consider the 3 vertices shown in the empty graph in Figure 6.5a, and assume that we can execute the steps in order of *decreasing* η. Because scenarios 2 and 3 differ in the possible realizations of both θ_1 and θ_3 (i.e., $\eta = 2$), we start with this potential edge in Figure 6.5b. No valid path exists between these vertices; thus, we add edge $(2, 3)$ to the graph.

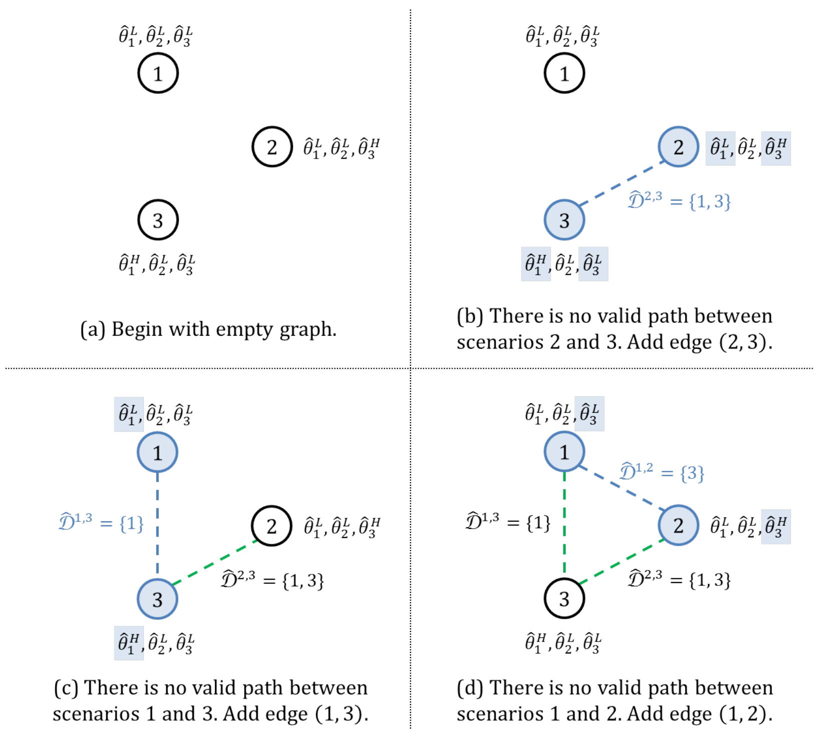


Figure 6.5: If we consider $\eta = 2$ before $\eta = 1$, the algorithm adds 3 edges to the graph.

We next consider scenarios 1 and 3 in Figure 6.5c, which differ only in the possible realization of θ_1 (i.e., $\eta = 1$). There is no valid path between these vertices either, so we add edge $(1, 3)$ to the graph.

Finally, in Figure 6.5d, we consider scenarios 1 and 2, which differ only in the possible realization of θ_3 (i.e., $\eta = 1$). It may appear that we have an existing path $(2,3,1)$; however, notice that $\hat{\mathcal{D}}^{2,3} \not\subseteq \hat{\mathcal{D}}^{1,2}$ and $\hat{\mathcal{D}}^{1,3} \not\subseteq \hat{\mathcal{D}}^{1,2}$. In other words, edges $(2,3)$ and $(1,3)$ will both disappear if θ_1 is realized, leaving no path between scenarios 1 and 2. It follows that we must add edge $(1,2)$ to the graph. Overall, this procedure provides us with 3 edges.

Now consider Figure 6.6. Here, we instead start with edges $(1,3)$ and $(1,2)$, for which $\eta = 1$, and we find that edge $(2,3)$ is not required. This leaves us with only 2 edges. Therefore, in this case, it is clear that we cannot evaluate the algorithm in order of decreasing η without adding unnecessary edges to the graph.

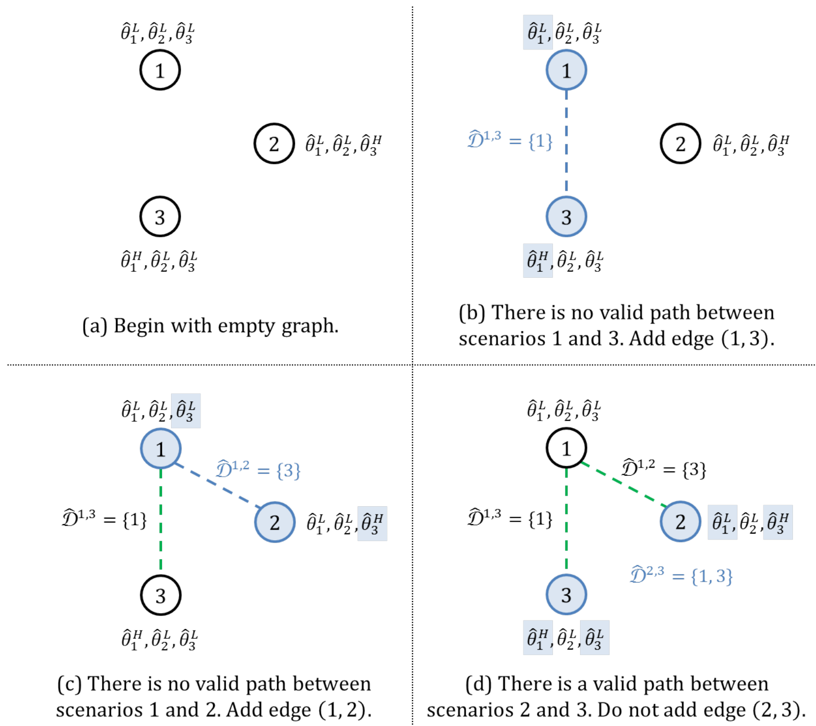


Figure 6.6: If we instead consider $\eta = 1$ before $\eta = 2$, the algorithm adds only 2 edges to the graph.

This is true in general. Recall that to have a valid path between scenarios s and s', each edge (s'', s''') in the path must satisfy $\hat{\mathcal{D}}^{s'',s'''} \subseteq \hat{\mathcal{D}}^{s,s'}$. But if we start with $\eta = I$ and work backward to $\eta = 1$, we will have $|\hat{\mathcal{D}}^{s'',s'''}| \geq |\hat{\mathcal{D}}^{s,s'}|$, which satisfies this condition only in the case that the cardinalities are equal. This means that once we exhaust all edges (s, s') for which $\eta = I$ and proceed to those for $\eta = I - 1$, none of the existing edges (s'', s''') in the graph can be used in a valid path. The same is true when proceeding from $\eta = I - 1$ to $\eta = I - 2$, and so forth.

If we instead start with $\eta = 1$ and proceed forward to $\eta = I$, we will have $|\hat{\mathcal{D}}^{s'',s'''}| \leq |\hat{\mathcal{D}}^{s,s'}|$. This means that the cardinalities for the edges in the graph are as small as possible, and these edges can thus potentially be used in any valid path when we proceed from $\eta = 1$ to $\eta = 2$, etc. It is therefore necessary to consider steps 4a and 4b of the algorithm in order of increasing η to avoid adding unnecessary edges to the graph. $\square$

Conceptually, the overall idea here is that we must: (1) attempt to satisfy all scenario-linking requirements by first connecting scenarios that differ only in the possible realizations of endogenous parameters associated with a *single source* of uncertainty, and then (2) gradually relax this restriction until all required links are generated.

6.6 Conclusions

In this chapter, we have presented an algorithm for non-anticipativity constraint generation in multistage stochastic programs with both endogenous and exogenous uncertainties. This approach produces the minimum number of NACs through a hybrid strategy in which the first-period and exogenous scenario pairs are generated by set definitions based on existing reduction properties, and the endogenous scenario pairs are generated by a new graph-theory algorithm based on recent work by Boland et al. (2016) and Hooshmand Khaligh and MirHassani (2016b). We have imposed no restrictions on the structure of the underlying scenario set. The results of Chapter 5 suggest that, in this case, the elimination of redundant NACs may also lead to a drastic reduction in problem size and solution times. For large instances, such techniques may be a necessary step for model generation, since memory limitations can easily rule out the consideration of fullspace, or even partially reduced, stochastic programs.

Acknowledgments

We gratefully acknowledge financial support from the John E. Swearingen Graduate Fellowship and the Center for Advanced Process Decision-making at Carnegie Mellon University.

Chapter 7

Stochastic Programming for Non-Experts

7.1 Introduction

Whether in the business world or a personal setting, it is often necessary to make the "best" decisions to satisfy some primary objective, subject to many competing requirements. For example, a company may wish to minimize operating costs while also ensuring that its production meets strict quality standards and customer demand. These types of word problems can generally be cast as *mathematical optimization* problems that have the following form (Williams, 2013):

$$\begin{aligned} \text{min} \quad & \textit{Operating costs} \\ \text{s.t.} \quad & \textit{Quality standards} \\ & \textit{Customer demand} \end{aligned}$$

Here, "min" indicates that we are attempting to minimize the operating costs (our *objective function*), and "s.t." indicates the start of the list of restrictions, or *constraints*, that our decisions are subject to. In practice, *Operating costs*, *Quality standards*, and *Customer demand* would each refer to equations that consist of different variables, although such details are outside the scope of this chapter.

To those without a background in mathematics, it is probably natural to question *why* we are introducing foreign concepts like objective functions and constraints. The simple answer to this question is that such formulations naturally lend themselves to standard solution methods that can automatically determine the best, or *optimal*, decisions *for us*. In other words, if we (or a colleague) can correctly formulate the mathematical optimization problem, we may be able to completely eliminate the guesswork from our decision making. This is especially useful in cases with thousands of constraints, where it would be nearly impossible for a human being to manually arrive at the optimal solution. This idea is emphasized at length in Sashihara (2011), which provides a nontechnical look at optimization and its value to businesses. As the basics of optimization are widely available in the literature, we will skip a more in-depth review of the subject and proceed with the current discussion.

7.1.1 How Do We Model Uncertainty in Optimization Problems?

Now, what happens if some of the problem data is not known with complete certainty? For example, what if a company has to plan a production schedule without knowing exactly what the customer demand will be? It is tempting to simply use our best guess for the uncertain values in such a case. Unfortunately, this approach can be very risky, since a wrong guess can lead to costly, unforeseen losses — a situation we will explore further in Section 7.2.

The good news is that it is possible to appropriately model cases like this. The somewhat bad news is that such problems are not well defined, and multiple approaches exist to hedge against uncertainty. In fact, there are so many options that Powell (2014) fittingly refers to this as "the jungle of stochastic optimization."

We can slightly narrow down our choices, however, depending on the problem type. For *planning and scheduling* under uncertainty, at least in our opinion, there are 4 primary approaches: stochastic programming, robust optimization, chance-constrained optimization, and dynamic programming. There are also 2 popular extensions that blur the lines between these approaches: risk-averse stochastic programming (which makes stochastic programming look a bit more like robust optimization), and adjustable robust optimization (which makes robust optimization look a bit more like stochastic programming). This is a total of 6 frameworks. To the non-expert reader, however, we acknowledge that this is a total of 6 unfamiliar names. The open question is: "Which approach should we use, and when?"

We propose three qualitative guidelines to answer this question:

1. Is *probability data* available? (For example, is there information available that describes how likely it is for the customer demand to take high or low values?)

2. Can *corrective action* be taken after decisions are implemented? (In other words, can we react to new information?)

3. Is the level of *feasibility* critical? (Specifically, does the solution still "work" if the input data changes? How about in very unlikely circumstances?)

We use these guidelines in a decision chart in Figure 7.1 to indicate when it is appropriate to use each approach. Notice that as one of the 8 possibilities in the figure, if probability data is available, corrective action is essential, and the level of feasibility is not a significant concern, we can use stochastic programming, dynamic programming, or adjustable robust optimization. In cases like this, where there is more than one reasonable option, the final choice often comes down to whichever approach provides a more "manageable" problem, as well as the user's personal preference. It is important to note that regardless of which option is used in such cases, the results will likely be much more reliable than if we had instead ignored the uncertainty altogether.

We will focus on stochastic programming for the remainder of this chapter, as it provides a very general framework for optimization under uncertainty.

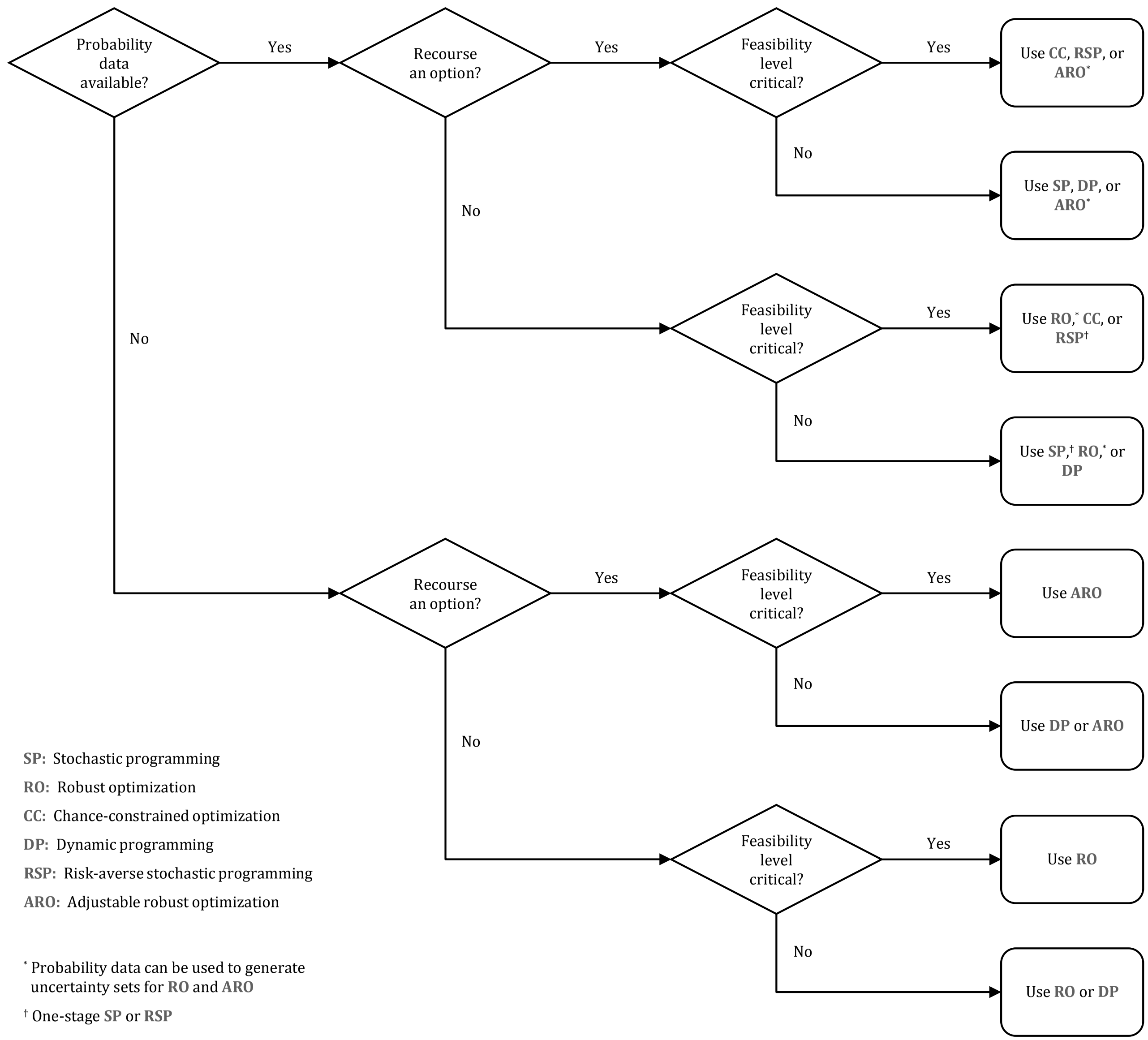


Figure 7.1: Qualitative guidelines for model selection.

7.1.2 Selecting an Appropriate Example

So far, we have briefly discussed optimization under uncertainty in very general terms. This may be difficult to conceptualize for many readers. It would be helpful, of course, to consider a concrete example.

In selecting an appropriate example, there are a couple of subtle points to keep in mind. First, who is the intended audience? As chemical engineers, it may be convenient for us to talk about process networks or oilfields. An electrical engineer, however, may not care about either of those things and might prefer to read about power grids. Or perhaps the reader is a mathematician (Figure 7.2) who has little appreciation for any type of engineering. In any case, it is important to select an example that *anyone* can relate to.

Figure 7.2: A mathematician who cannot fathom why anyone would have trouble understanding stochastic programming.

A second point is that classic examples in the literature, such as the farmer problem or the news vendor problem (Birge and Louveaux, 2011), identify optimal *quantities* of materials for use in the situation at hand; specifically, the amount of farm land to allocate to different types of crops or the number of newspapers to purchase from a distributor. In order to solve these problems, some knowledge of linear optimization is required. While this is a simple task for an expert, someone without a mathematical background would almost certainly have a difficult time coming to the conclusion that we need to plant exactly 250 acres of sugar beets.

The motivating example presented in the next section concerns an everyday situation that should be fairly easy to follow, regardless of background. It also includes only “yes” or “no” decisions, which we believe is more intuitive since the optimal solution can easily be obtained by checking each possible combination of decisions. We will frame all further discussions in the context of this example.

7.2 The Motivating Example

Meet Quinn (Figure 7.3). She is looking to buy a new car, and there is a limited-time sale going on at the local dealership. She has narrowed down her choice to 3 different cars, as shown in Figure 7.4. Each vehicle has a different purchase price and depreciates at a different rate. Specifically, after a 5 year period, she will be able to resell the cheapest car for only $3,000 (a $7,000 loss), the midgrade car for $10,000 (a $5,000 loss), and the most expensive car for $17,000 (a $3,000 loss). Clearly, the more expensive the car is, the better it holds its value.

Figure 7.3: Meet Quinn.

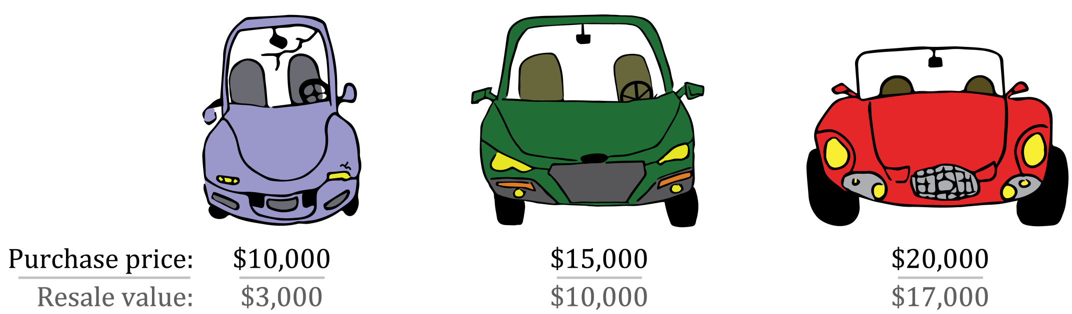


Figure 7.4: Quinn's 3 car choices.

Quinn's objective is to minimize her total cost of ownership based on purchase price and resale value. The agreement is that she will sign the paperwork now and then use her bonus from work to pay for the full price of the car upon delivery. Since she assumes that she will receive a $20,000 bonus, she considers this an easy decision: go with the most expensive car, which will lead to the lowest overall cost over 5 years. She also loves the color red, so it should be a win-win.

Unfortunately, Quinn's boss, a greedy hedge-fund manager (Figure 7.5), has other plans. He has his eye on a second yacht and is cutting bonuses this year to help pay for it. Quinn's bonus gets cut by 25% and she receives only $15,000.

Due to the new circumstances, Quinn is forced to settle for the midgrade car. She also discovers that the fine print in her new-car agreement had specified a 10% change fee in the event of any

Figure 7.5: Quinn's boss perusing a yacht catalog.

modifications to the order. She begrudgingly hands over 10% of the original order — a $2,000 penalty (see Figure 7.6) — and drives off the lot with a sick feeling in her stomach.

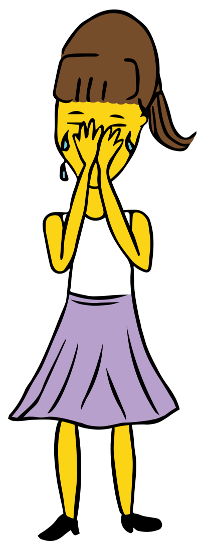


Figure 7.6: Quinn must pay a $2,000 penalty for changing the terms of her new-car agreement.

So how can we avoid situations like this?

7.3 The Basics of Stochastic Programming

Stochastic programming is a framework for optimization under uncertainty, generally applicable in cases where probability data is available and corrective action, or *recourse*, is essential. Feasibility is not always a significant concern here, due to our ability to take recourse or to re-solve the problem at a later point in time.

Interestingly enough, this approach was independently introduced by Beale (1955) and Dantzig (1955) in the same year more than 6 decades ago. It has been used extensively in academia for about the last 25 years, and has been applied to problems in a wide assortment of areas (Wallace and

Ziemba, 2005): production planning and scheduling, supply chain optimization, network resource utilization, electricity generation, lake eutrophication management, climate change mitigation, and groundwater pollution control, just to name a few.

As we proceed to discuss the basics of stochastic programming, please keep in mind that we will focus on *concepts*. In the words of King and Wallace (2012):

> "We believe that there is a more serious need for [an article] explaining the underlying 'whys' than the technically deeper 'hows.' But also, we try to avoid drowning the conceptual difficulties in technical details. We believe that whenever a student or user has the basic questions and difficulties in place, she can always walk down the alleys of detail reflecting her needs and abilities."

To allow the reader to fully appreciate how stochastic programming works, we will begin with a discussion on simulation.

7.3.1 Simulation

Consider the case shown in Figure 7.7, where we separately evaluate each of the possible scenarios for Quinn's car purchase. This amounts to 3 simulation problems: one where she receives a $10,000 bonus, one where she receives a $15,000 bonus, and another where she receives a $20,000 bonus. Probability information is available based on office gossip. In particular, it is 30% likely that Quinn and her colleagues will receive the lowest or highest bonus, and it is 40% likely that they will end up right in the middle at $15,000.

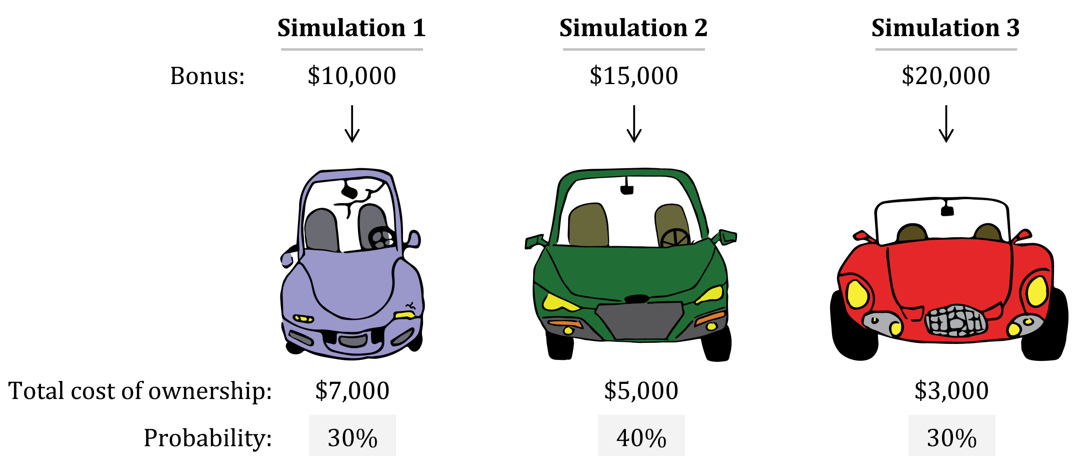


Figure 7.7: A simulation approach.

The solution to each of the simulation problems is fairly straightforward since the optimal solution is always for Quinn to purchase the most expensive car that she can afford. If the first scenario occurs, she will purchase the cheapest car, leaving her with a total cost of ownership of $7,000 after 5 years. If the second scenario occurs, she will purchase the midgrade car, corresponding to a total cost of ownership of $5,000 after 5 years. Likewise for the third scenario, she will purchase the most expensive car for a total cost of $3,000.

Because we cannot know which scenario will occur before it actually happens, the best we can do is calculate the total *expected* cost. This simply means that we will average the costs from each scenario based on their probabilities (30%, 40%, and 30%, respectively). In other words:

$$0.30(\$7,000) + 0.40(\$5,000) + 0.30(\$3,000) = \$5,000$$

Thus, on average, Quinn's total cost of ownership is $5,000 based on simulation.

7.3.2 2-Stage Stochastic Programming

Let us now make Figure 7.7 slightly more abstract and consider two points in time, A and B. Point A occurs *before* we know what the value of the bonus will be, and point B occurs *after*. This is shown in Figure 7.8. Based on this representation, it should be clear that in the simulation approach, the car purchasing decision occurs at point B. We may therefore refer to simulation as a "wait-and-see" strategy because the decision is purposely delayed until after the uncertainty is resolved.

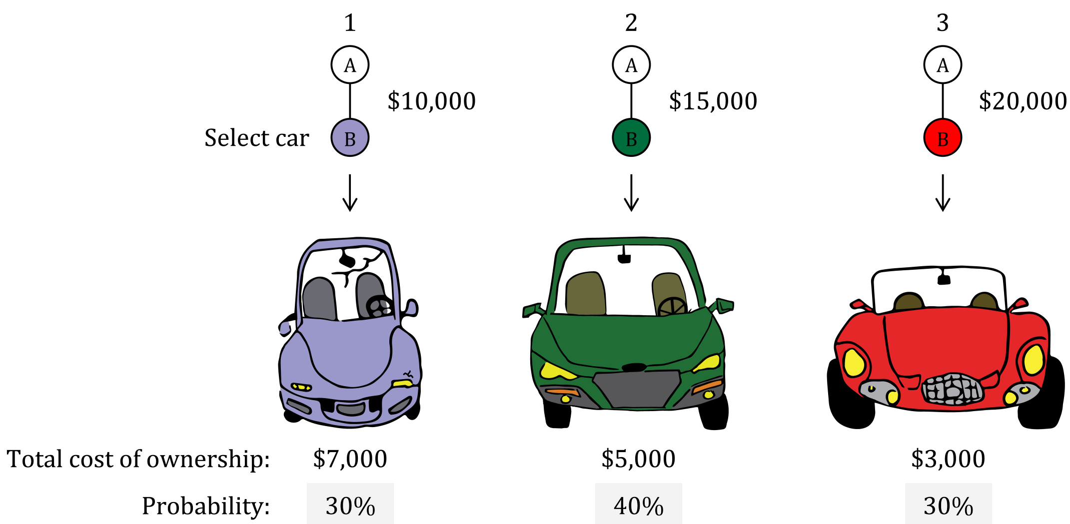


Figure 7.8: Simulation can be viewed as a "wait-and-see" strategy.

But there is an obvious problem with this approach: the idea that the decision can wait is often completely unrealistic, and in this case, Quinn would miss the sale at the dealership. In actuality, she must purchase *a single car* at point A, before complete information is available. The simulation approach does not provide any guidance for making such a decision. In fact, we can see in Figure 7.8 that it merely indicates the purchase of a different car at point B in each of the 3 cases, which by itself is not particularly helpful.

The main takeaway here is that simulation only provides us with a partial view of the full decision-making process. Stochastic programming directly addresses these shortcomings, as shown in Figures 7.9 and 7.10. We will discuss the solution shortly, but we first address the primary differences from simulation.

Specifically, in Figure 7.9, the car purchasing decision now occurs at point A. An additional change is that we have introduced horizontal lines that connect point A in each of the three scenarios. These lines, known as *non-anticipativity constraints*, ensure that Quinn makes a single decision at that time, regardless of scenario, without anticipating any one particular outcome for her bonus. In other words, her purchasing decision explicitly accounts for the fact that her bonus may be \$10,000, \$15,000, or \$20,000, instead of separately planning for each possibility like in the simulation approach in Figure 7.8. (A technically-inclined reader may notice that the simulation problem is in fact a *relaxation* of the stochastic programming problem.)

To use the proper terminology, we note that stochastic programming researchers refer to point A as the "first stage" or "stage 1." First-stage decisions are also commonly referred to as "here-and-now" decisions in what is perhaps a more straightforward characterization.

If we formulate and solve the 2-stage stochastic programming problem, the optimal solution indicates that Quinn should take a conservative approach and initially agree to purchase the cheapest car, as shown in Figure 7.9. This option has the smallest change fee, which allows her the most flexibility at point B once the true value of her bonus has been realized (Figure 7.10). Specifically, if her bonus ends up being better than the worst case, she can pay the 10% penalty (\$1,000) to upgrade to a better option. Note that point B, where these recourse decisions are made, is properly known as the "second stage" or "stage 2." Second-stage decisions are also commonly referred to as "wait-and-see" decisions.

If the first scenario occurs, Quinn will stay with the cheapest car, leaving her with a total cost of ownership of \$7,000 after 5 years (i.e., \$7,000 + \$0 penalty). If the second scenario occurs, she will switch to the midgrade car, corresponding to a total cost of ownership of \$6,000 after 5 years (i.e., \$5,000 + \$1,000 penalty). Similarly for the third scenario, she will switch to the most expensive car for a total cost of \$4,000 (i.e., \$3,000 + \$1,000 penalty). We encourage the reader to confirm that this solution is in fact cheaper than starting out with a more expensive car at stage 1 and then potentially downgrading at stage 2.

Note that although the simulation approach also provides "wait-and-see" decisions (which happen to be identical in this case), they are not in the context of any "here-and-now" decisions and may be suboptimal or even infeasible in other problems.

The total expected cost is:

$$0.30(\$7,000) + 0.40(\$6,000) + 0.30(\$4,000) = \$5,700$$

Thus, on average, Quinn's total cost of ownership is \$5,700 based on 2-stage stochastic programming. We will consider this solution in further detail in the next section.

As a final point, we note that, in general, we are not restricted to only 2 decision stages. A problem with more than 2 stages is known as a *multistage* stochastic programming problem.

7.4 Interpreting the Results

The solution of the stochastic programming problem instructs Quinn on what to do *now*, before all relevant information is available, and what to do *later*, after more information has been revealed. But what exactly does the total expected cost tell us? Clearly, these values are different for the simulation and 2-stage stochastic programming approaches (\$5,000 and \$5,700, respectively), and the simulation approach actually has a more appealing value. We address this source of confusion

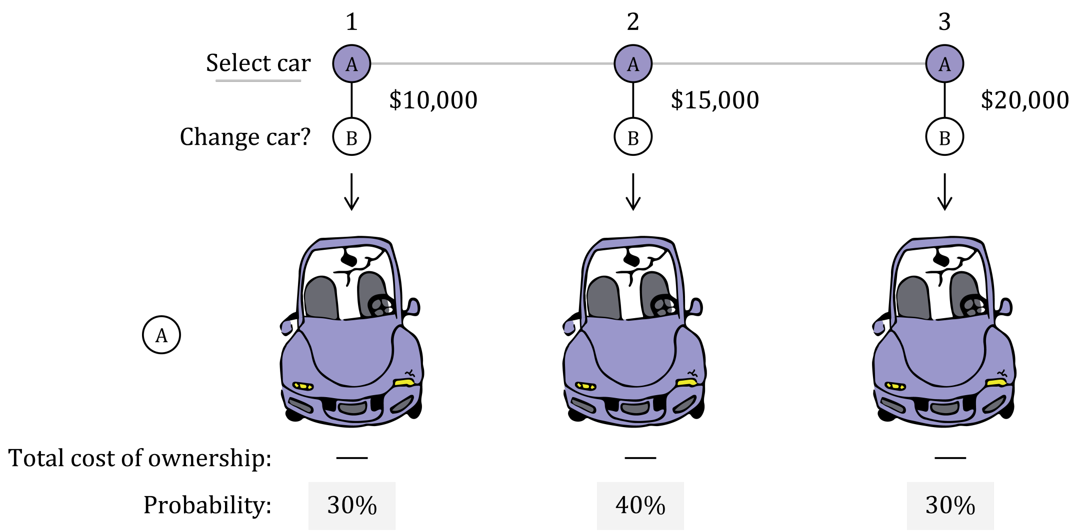


Figure 7.9: First-stage (here-and-now) decisions in stochastic programming.

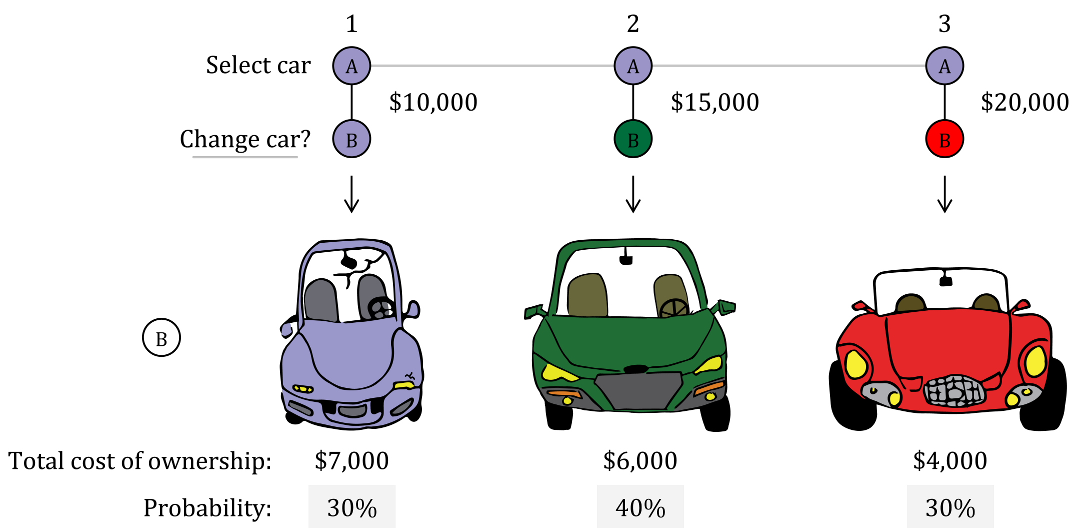


Figure 7.10: Second-stage (recourse) decisions in stochastic programming.

in the following two sections.

Before proceeding, we note that there are two primary approaches for evaluating the benefits of a stochastic programming solution: (1) calculating the *value of the stochastic solution*, and (2) using simulation. Both approaches rely on a comparison to a version of the problem where uncertainty is ignored, which is known as a *deterministic* problem. The specific deterministic problem used in these cases is one in which expected values are assumed for all uncertain quantities. This generally allows for the fairest comparison, since intentionally using the best or worst values can unfairly skew the results in favor of stochastic programming. In Quinn's case, the deterministic problem of choice should then be simulation problem 2 in Figure 7.8, with an expected bonus of \$15,000 (i.e., 0.30(\$10,000) + 0.40(\$15,000) + 0.30(\$20,000) = \$15,000), rather than the best case of \$20,000 previously assumed in Section 7.2 in the absence of any probability data. We first address the value of the stochastic solution in this context. Note that for the convenience of any interested readers, we provide the deterministic and stochastic programming models in Appendix D.

7.4.1 The Value of the Stochastic Solution

The value of the stochastic solution (Birge and Louveaux, 2011), or VSS for short, is the most common metric for evaluating the benefits of stochastic programming over deterministic optimization. The general idea here originates in the fact that we simply cannot directly compare the results of the two approaches. The reason for this can be seen by comparing simulation problem 2 in Figure 7.8 to all three scenarios in Figures 7.9 and 7.10. The deterministic problem has no "here-and-now" decisions and anticipates one particular outcome for Quinn's bonus. As previously described in Section 7.3.2, these are two fairly different frameworks. The challenge here, unfortunately, is that many people try to compare them anyway.

Consider the total cost of ownership of \$5,000 for the deterministic problem and the total expected cost of ownership of \$5,700 for the stochastic programming problem. The difference is \$700, with the stochastic programming problem actually costing *more*. This can lead to enormous uphill battles for researchers trying to convince business managers to adopt stochastic programming, since the raw numbers look *bad*. The numbers can look so bad, in fact, that Figure 7.11 may be the most likely outcome of such conversations.

The VSS "corrects" the solution of the deterministic problem so that it can be properly compared to the stochastic solution. It does this by first implementing the deterministic decisions as "here-and-now" decisions in the 2-stage stochastic framework, and then re-solving the corresponding stochastic programming problem. Recall that Quinn purchases the midgrade car in the deterministic problem (as previously indicated in simulation problem 2 in Figure 7.8). Thus, in this context, Quinn would first purchase the midgrade car in Figure 7.9 instead of the cheapest car, and re-solving the corresponding stochastic programming problem would yield the optimal recourse decisions, which happen to be the same as those shown in Figure 7.10. It should be no surprise that this is not a great purchasing strategy. Scenario 1 in Figure 7.10 would have a total cost of ownership of \$8,500 after 5 years (i.e., \$7,000 + \$1,500 penalty), for scenario 2 this would be \$5,000 (i.e., \$5,000 + \$0 penalty), and for scenario 3 it would be \$4,500 (i.e., \$3,000 + \$1,500 penalty). The total expected cost of ownership would then be \$5,900 — \$200 *more* than the stochastic programming solution. In other words, the deterministic approach would actually lead to

Figure 7.11: Convincing business managers to adopt stochastic programming can be an uphill battle.

suboptimal decisions, and the use of stochastic programming would lead to a savings of \$200 (i.e., VSS = \$200). The original difference of \$700 is thus meaningless and incredibly misleading.

To address the earlier point of the simulation approach providing a "more appealing value" for the total expected cost of ownership, the explanation follows directly from the previous discussion. This value is obtained by taking the probability-weighted average of three separate deterministic problems, and each of those problems underestimates the total cost. Directly comparing this result to the stochastic programming solution is ill-advised, since it gives the completely false impression that simulation performs best.

We summarize the primary differences between stochastic programming and a deterministic approach in Figure 7.12. The overall idea is that deterministic problems are tailored toward *one specific outcome*, so solutions will look better at first glance. These solutions are often suboptimal in practice, however, because they are based on less information and are thus less flexible in the event of an unexpected outcome. Stochastic programming provides a solution that is far less risky, and the corresponding decisions are best, on average, for *all possible outcomes*.

In the next section, we explore the use of simulation to convey this information in a slightly different manner.

7.4.2 Using Simulation to Measure the Effectiveness of Stochastic Solutions

In some cases, it may be beneficial to illustrate the effectiveness of a stochastic solution by framing the results as a competition between a stochastic decision maker and a deterministic decision maker (You et al., 2009). Such techniques are fairly well known in the literature (see, for instance, Shapiro and Philpott, 2007).

Here, we will assume that we have Stochastic Quinn and Deterministic Quinn. Stochastic Quinn uses stochastic programming to solve the car-buying problem, and she has available to her the solution shown in Figures 7.9 and 7.10. This solution accounts for the fact that her bonus may be \$10,000, \$15,000, or \$20,000. Deterministic Quinn uses a deterministic approach, and has at her disposal the solution corresponding to simulation problem 2 in Figure 7.8. This solution considers only the possibility that her bonus will be \$15,000.

To begin, the two Quinns must implement their first-stage ("here-and-now") purchasing decision. This means that Stochastic Quinn will agree to buy the cheapest car, and Deterministic Quinn will agree to buy the midgrade car. We then evaluate how their decisions perform in each of three possible realities: one in which the bonus is \$10,000, one in which the bonus is \$15,000, and one in which the bonus is \$20,000. The total cost of ownership for each case is shown in Figure 7.13. (Note that this is the same data presented in the previous section.)

We can see in the figure that Stochastic Quinn's second-stage (recourse/"wait-and-see") decisions coincide with those of the three scenarios in Figure 7.10. There are no surprises here. Deterministic Quinn, on the other hand, did not plan to have to change her original purchasing decision. If the bonus is \$15,000, as was the case in Section 7.2, she does very well — \$1,000 better than Stochastic Quinn. But if the bonus is \$10,000, she is unprepared and loses \$1,500. If the bonus is \$20,000, she loses \$500. The overall lesson here is that a deterministic approach is essentially a gamble: if Quinn correctly predicts the future, she will perform very well; if her prediction is off, she will generally lose out. Stochastic programming allows Quinn to prepare for more than just a

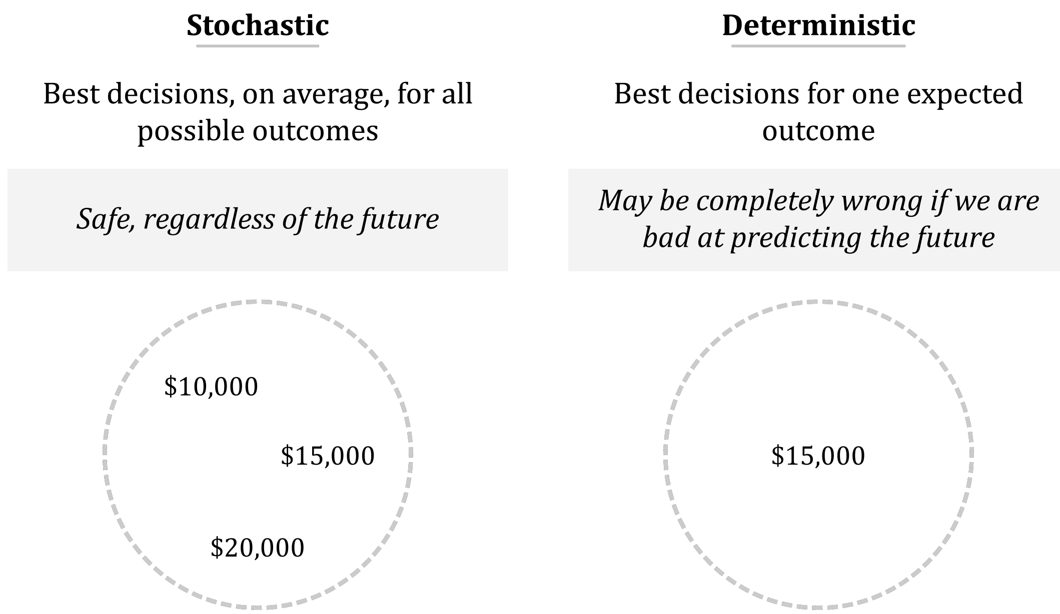


Figure 7.12: Primary differences between stochastic programming and a deterministic approach.

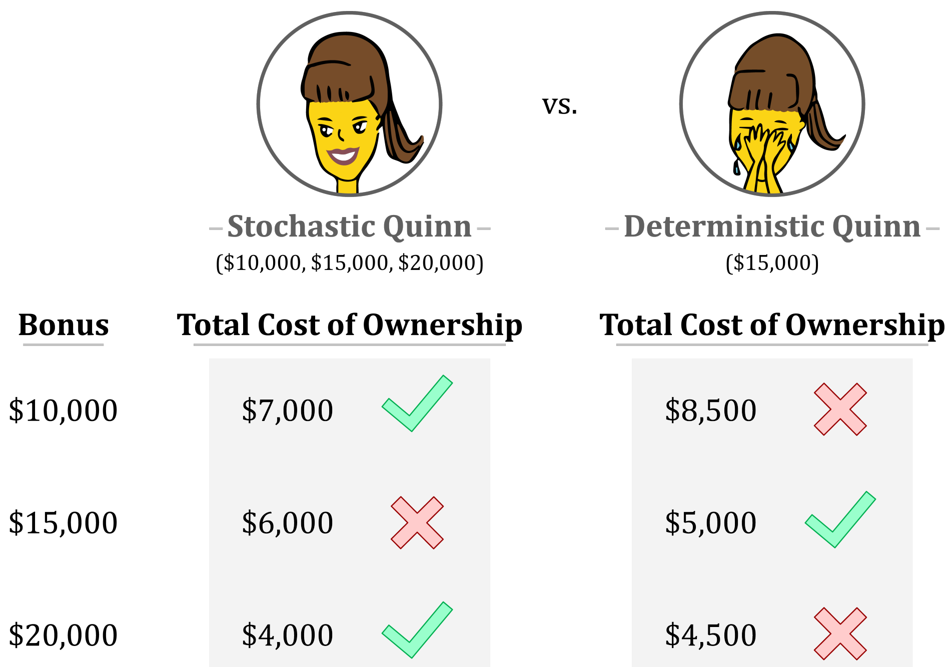


Figure 7.13: A competition between a stochastic decision maker and a deterministic decision maker.

single case.

7.5 Other Non-technical Resources for Stochastic Programming

Hopefully the reader now appreciates that, at least in concept, stochastic programming is fairly easy to understand. One final question, however, is: "If stochastic programming has been around for more than 60 years, why is there not already an explanation like this in the literature?"

The answer to this question is mostly based in opinion. However, the truth is that there is a very limited number of educational resources available to non-expert users. This problem is certainly not confined to stochastic programming (and is likely an issue in any highly-specialized research area), but it does appear that the "by mathematicians, for mathematicians" style of most publications in this area is at least somewhat problematic. Such sentiments can be traced back almost 25 years to the first textbook on stochastic programming, Kall and Wallace (1994):

> "Over the last few years, both of the authors, and also most others in the field of stochastic programming, have said that what we need more than anything just now is a basic textbook — a textbook that makes the area available not only to mathematicians, but also to students and other interested parties who cannot or will not try to approach the field via the journals."

Stochastic programming is primarily math based, and Kall and Wallace (1994) certainly lived up to those standards. This was the appropriate choice for the first textbook in the area (at least in our opinion).

The problem is that years later, with the technical foundation of stochastic programming firmly in place, some researchers still seem to resist literature that simplifies the explanation of stochastic programming, instead viewing such efforts as trivializing the field. In the words of King and Wallace (2012):

> "At one point several attempts to publish the text were made but failed, possibly because the text was not good enough, certainly because the editors (or their reviewers) did not appreciate that stochastic programming was more than mathematics and algorithms."

This textbook is unmatched as far as conceptual explanations are concerned, and we highly recommend King and Wallace (2012) to anyone interested in a gentle introduction to stochastic programming. Other popular "introductory" texts (which are largely math based) include Sen and Higle (1999), Higle (2005), Shapiro and Philpott (2007), and Birge and Louveaux (2011).

While stochastic programming problems can pose unique modeling and computational challenges, these technical details are generally handled by researchers who are capable of sorting out these issues. The major barrier remaining in adopting stochastic programming outside of academia is then *how to sell this technique* to someone without a math or engineering degree. In other words, the difficulty in interpretation of the results (Grossmann et al., 2016) can be the difference between a business manager seeing value in using stochastic programming and being utterly confused with why "what we have now" needs to be fixed in the first place. We have attempted to address this concern in the previous section.

With these considerations in mind, it is probably no surprise that there are very few (publicly-disclosed) real-world implementations of stochastic programming. The classic success story is the Russell-Yasuda Kasai model (Cariño et al., 1994), which is a stochastic-programming-based asset/liability management model for determining an optimal investment strategy for a Japanese insurance company. This strategy led to a $79 million increase in income over only a 2-year period. Other equally impressive successes may certainly exist; however, we suspect that few companies would be willing to disclose details about anything that gives them such a major competitive advantage.

We openly encourage others in the stochastic programming community to share their success stories, as well as to make their work more accessible to those outside the field. In the latter case, previous work addressing the analysis of optimal solutions (e.g., Greenberg, 1996) may be of particular interest.

7.6 Conclusions

Life is uncertain; it is often the case that we must make decisions before all of the relevant information is available to us. If we would like to make the *best* possible decisions in such cases, we generally look to one of the many techniques for optimization under uncertainty. We have presented three qualitative guidelines and a corresponding decision chart (Figure 7.1) to help users make this selection.

Stochastic programming, in particular, provides a very general framework for optimization under uncertainty, provided that probability data is available. Specifically, it aligns with the way that we commonly make decisions in real life: we must make a decision "here and now" before knowing exactly what will happen in the future, and we can then take recourse in the future once complete (or more complete) information is available to us. The scenario-based form of stochastic programming can be viewed as a rigorous extension of simulation.

A complicating factor in the interpretation of the results is that stochastic solutions tend to appear to be *worse* than their deterministic counterparts. This misleading comparison can be corrected with the concept of the value of the stochastic solution (VSS), or similarly explained away with the use of a simulation strategy that puts a stochastic decision maker up against a deterministic decision maker. The main takeaway here is that deterministic decisions are made with *one* expected outcome in mind, and can easily lead to unexpected losses due to the unpredictability of the future. Stochastic programming considerably lowers this risk by providing decisions that are best, on average, for *all* possible outcomes.

Unfortunately, there are currently very few resources for non-experts interested in learning the basics of stochastic programming, and this framework has found limited adoption outside of academia. It is our hope that this chapter will be the first step in bridging the gap between these users and the stochastic programming community.

Acknowledgments

We gratefully acknowledge financial support from the John E. Swearingen Graduate Fellowship and the Center for Advanced Process Decision-making at Carnegie Mellon University. We also thank Melissa Wertz and Holly Apap for the cartoons used throughout this chapter.

Chapter 8

Conclusions

We now critique all previous sections of this thesis in order to summarize the strengths and weaknesses, as well as overall contributions, for the presented work.

8.1 Review of Chapter 1

Chapter 1 provides a thorough and unifying review of all major publications in the area of stochastic programming under endogenous uncertainties (Type 1 and Type 2) and both endogenous and exogenous uncertainties. As discussed, these areas are still maturing and there is much room for future research (which we later consider in Section 8.9). Note that we purposely discuss only a few publications on purely-exogenous uncertainty as this area is fairly well known.

After this point, we focus on the basic concepts behind purely-exogenous uncertainty and purely-endogenous uncertainty, with a strong emphasis on the flow of the decision-making process and the structure of the corresponding scenario trees. Perhaps most notably, we formalize the idea of the decision-dependent endogenous scenario tree by introducing a superstructure representation that captures all possible outcomes of the tree. We use several figures to point out the differences between the two types of uncertainty in an effort to convey the information as clearly as possible. We emphasize that a review of this scope and level of accessibility does not exist anywhere else in the literature.

8.2 Review of Chapter 2

Here we introduce the basic definitions and notation. The notation is admittedly quite complex and may be somewhat difficult for even a seasoned researcher. We attempt to base all of our discussions on the underlying scenario trees introduced in Chapter 1 and include many brief examples to indicate how the set and parameter definitions may appear in practice.

The most notable contribution of this section is our "composite" scenario tree. Similar to the endogenous case, this is also a superstructure. Its novelty, however, rests on the fact that it accommodates both endogenous and exogenous realizations by duplicating the exogenous scenario tree for each possible combination of realizations of the endogenous parameters. The structure of this tree not only makes the ensuing discussions easier to follow (by providing the reader with something tangible to refer to), but also serves as the basis for several theoretical reduction properties presented in Chapter 4, which we review in Section 8.4.

We also briefly review scenario probabilities in this chapter, which follow along closely with the previously-mentioned definitions.

8.3 Review of Chapter 3

The notation and definitions introduced in Chapter 2 are then used to define our models in Chapter 3. Our first step in presenting the stochastic programming models is to introduce a simple deterministic planning model. We gently extend this case to account for exogenous uncertainty, which allows the reader to clearly see what simple changes are required to convert the model. We follow a similar procedure to illustrate how the deterministic model is extended to the purely endogenous case, and then finally to the case of both endogenous and exogenous uncertainties. The idea here is to follow a logical progression of increasing complexity so as to help the reader understand the basic concepts, instead of immediately presenting a monolithic model with no prior context.

The three stochastic programming models are based on the assumption that the underlying scenarios are the result of a Cartesian product over all possible combinations of realizations of the uncertain parameters. We do not relax this assumption until Chapter 6, which we review shortly in Section 8.6.

Due to the Cartesian-product assumption, some readers may perceive the models as too "academic," with little applicability in the real world. However, we feel that this is not an accurate assessment. First, general models can be derived from an "academic" special case, and this task is often much easier (or may only be possible) after gaining invaluable insights from studying such special cases. Our Cartesian-product-based models may be viewed from the same perspective, as they allow us to explain concepts with well-behaved math in the context of predictably-structured scenario trees before proceeding to a more general, and less intuitive, case. Furthermore, a second point is that Cartesian products are perfectly acceptable in the preliminary phases of a planning project, simply to gauge whether or not stochastic programming may be the right fit.

An additional weakness of these models is that they do not directly account for gradual resolution of uncertainty (see Section 8.9.3 for details on future work in this area).

8.4 Review of Chapter 4

This chapter is likely the most difficult section in the thesis. We suspect that it will be read in its entirety only by researchers interested in expanding upon this work. In fact, our general philosophy is that it is not *meant* to be carefully read by everyone working in this area. We will return to this point below.

In this chapter, we present theoretical reduction properties for first-period scenario pairs, exogenous scenario pairs, and finally endogenous scenario pairs. The motivation for this endeavor is that naively-generated stochastic programs can easily contain several thousand (or more) linking constraints. Aside from the fact that the presence of such constraints can make the model extremely difficult or even impossible to solve, it is often the case that many of these constraints are not even necessary. Specifically, they may be implied by other constraints in the model and can be safely removed. Doing this in a systematic way, however, is very challenging.

The discussions in this chapter are rooted in the structure of the underlying composite scenario tree as well as in rigorous mathematical proofs. Figures are provided for all reduction properties in an attempt to ground these abstract concepts in reality. By the end of the chapter, we show that we have eliminated *all* redundant scenario pairs, which is synonymous with removing all redundant non-anticipativity constraints. This is a major achievement, since our model (MSSP) is considerably more general than that of the purely-endogenous case for which Gupta and Grossmann (2011) previously eliminated all redundant NACs. This of course also lends itself to significant improvements in model size and solution times, as we discuss in the review of Chapter 5. Generally speaking, the reduction properties can enable us to study problems that would otherwise be impossible to even load into memory.

The reason why we previously stated that it is unnecessary for all researchers to fully understand these concepts is that they are defined in a very general way that can be applied to almost *any* stochastic programming problem. This means that, in principle, a user may supply a deterministic model with additional data characterizing the uncertainty, and we can, in turn, automatically generate the stochastic programming model. This capability can be made available in a framework such as PySP (Watson et al., 2012). The overall idea here is that researchers should be able to use our ideas for their own purposes instead of being expected to independently reinvent our formulations. We explore this idea further in Section 8.9.1 on future work.

The primary weakness of these properties is that three of them require that the scenario set be generated by Cartesian products over all endogenous realizations. As mentioned at the end of the previous section, we address this issue in Chapter 6, which we review in Section 8.6.

8.5 Review of Chapter 5

Given that our reduced multistage stochastic programming models are often still too large to solve directly with commercial MILP solvers, we propose two special solution methods for this class of problems. The first is a newly-developed heuristic that we refer to as sequential scenario decomposition; the second is Lagrangean decomposition.

The general idea behind our heuristic is that we sequentially solve purely-endogenous MILP subproblems that consist of only a small fraction of the total number of scenarios. After solving each subproblem, we: (1) extract the binary investment decisions from the solution, and (2) fix these values in scenarios of the original problem in order to satisfy the corresponding first-period and exogenous NACs. Once we have exhausted all subproblems, all binary investment decisions are fixed. We then solve the resulting model to obtain a feasible solution to the original problem.

For Lagrangean decomposition, we provide a brief review of the basics, primarily based on previous work by Goel and Grossmann (2006) and Gupta and Grossmann (2011, 2014a). Although we do explicitly show the decomposed form of our model, we do not offer any fundamentally new information here.

We test our reduction properties and solution methods on two (modified) example problems from the literature: the capacity expansion of a process network (Goel and Grossmann, 2006) and oilfield development planning (Gupta and Grossmann, 2014a). We begin with a very small instance of the process network problem as a motivating example. This allows us to provide a physical interpretation of the solution along with the optimal structure of the composite scenario tree.

We next consider a much larger instance of this problem with more than 176 million constraints and approximately 4.8 million binary variables in its fullspace form. This instance cannot be loaded into memory. After applying our reduction properties, we are able to reduce the number of constraints to about 838,000 (a 99.5% reduction) and the number of binary variables to about 61,000 (a 98.7% reduction). The problem can then be solved directly with CPLEX to a 0.99% optimality gap in about 1 hour. Our heuristic is able to find the same feasible solution in just 61 seconds. While Lagrangean decomposition provides us with acceptable bounds on the optimal solution in much less time than solving the reduced model directly, the consequence of relaxing all conditional endogenous NACs for this approach is apparent in the fact that we cannot reduce the optimality gap below 1.41%.

Finally, we solve a medium-sized instance of the oilfield development planning problem. This is a very challenging task, even with only 64 scenarios, as the instance contains approximately 304,000 constraints and 9,000 binary variables. Attempting to solve the reduced model directly (consisting of about 123,000 constraints and 8,900 binary variables) yields an optimality gap of about 50% after more than 11 hours of CPU time. The SSD heuristic, on the other hand, obtains a high-quality feasible solution in just 41 seconds. Lagrangean decomposition also performs very well in this case and, within 14 seconds, provides bounds that confirm that the heuristic solution is within 0.20% of the optimum.

8.6 Review of Chapter 6

As previously mentioned, the models presented in Chapter 3 assume that the set of scenarios has been generated by a Cartesian product over all sets of uncertain-parameter realizations. Here we relax this assumption in order to account for arbitrary scenario sets.

We begin by proposing a more general form of model (MSSP) and discuss how the reduction properties change in the absence of Cartesian products. The primary impact is that one fundamental property no longer applies. Because this is a necessary condition for two of our other reduction properties, we lose those as well. In order to still be able to remove redundant non-anticipativity constraints in such cases, we make use of recent work by Boland et al. (2016) and Hooshmand Khaligh and MirHassani (2016b) and introduce the concept of a non-anticipativity graph for endogenous scenario pairs. We then use this idea to propose a hybrid strategy for minimum-cardinality non-anticipativity constraint generation in which: (1) the first-period and exogenous scenario pairs are generated by set definitions based on existing reduction properties, and (2) the endogenous scenario pairs are generated by an extended form of the polynomial-time algorithm developed by Hooshmand Khaligh and MirHassani (2016b). Although this work remains theoretical and has not been implemented in practice, we believe that eliminating the leftover, redundant NACs that cannot be removed by our existing reduction properties can lead to a drastic reduction in problem size and solution times. Additionally, in very large problem instances, such techniques may be essential.

8.7 Review of Chapter 7

A significant problem that we have noticed in the field of stochastic programming is that there are very few educational resources available to non-expert users. This of course makes it considerably more difficult for researchers to promote these techniques among business managers and others outside the field. The rather frustrating point here, however, is that it does not have to be this way; the basic concepts behind stochastic programming are actually quite natural and can be explained without the use of complex mathematics. With some effort, we believe that it is possible to even make the discussion easy to understand.

We present the first attempt (of which we are aware) to convey the ideas of stochastic programming to a non-technical audience. We begin by discussing optimization and then optimization under uncertainty in very general terms, and we propose 3 qualitative guidelines for model selection. These guidelines are also very general and do not require any advanced knowledge of optimization or even mathematics.

Due to our belief that stochastic programming provides a more general framework for optimization under uncertainty than the other modeling options, we adopt a narrow view and focus on only this one approach. Our conceptual explanation revolves around the relationship to simulation, as well as the use of simulation to illustrate the advantages of stochastic programming over a deterministic approach. We frame most of the chapter in the context of a motivating example related to a car purchase, which almost anyone can relate to.

The discussions in this chapter are specifically tailored to 2-stage stochastic programs with exogenous uncertainty, and the figures include cartoons (which, admittedly, may not be appealing to some readers). As for the first point, we may prepare similar types of papers in the future to explain multistage stochastic programming and endogenous uncertainty. To the latter point, we strongly believe that visualization is critical for a clear, conceptual explanation.

It is difficult to predict the long-term impact of this work as it is still a new area of research. Of course, we hope that it will be well received and perhaps even influence companies and other researchers to begin using stochastic programming.

8.8 Contributions of Thesis

In summary, the following are the major contributions that have emerged from the research work for this thesis:

1. A thorough and accessible review of exogenous uncertainty and endogenous uncertainty in multistage stochastic programming.

2. A superstructure form for endogenous scenario trees.

3. A "composite" scenario tree for both endogenous and exogenous realizations.

4. General multistage stochastic programming models with endogenous and exogenous uncertainties that can easily be generated from a deterministic model.

5. New theoretical reduction properties to eliminate all redundant non-anticipativity constraints in multistage stochastic programs with endogenous and exogenous uncertainties.

6. A "sequential scenario decomposition" heuristic, which provides near-optimal solutions in very little CPU time.

7. Orders-of-magnitude reduction in the number of variables and constraints, as well as in solution times, through the use of our reduction properties and heuristic, as demonstrated in process network and oilfield development planning problems.

8. A graph-theory approach for generating the minimum number of non-anticipativity constraints in problems with arbitrary scenario sets.

9. The first stochastic programming paper designed specifically for the qualitative interpretation of results by non-expert users.

8.9 Future Work

Given that the study of stochastic programming under endogenous and exogenous uncertainties is still in its very early stages, the possibilities for future work appear to be almost endless. It is hard to predict where the field may be in 20 years. With the advent of quantum computing, for example, it may be possible to one day instantaneously solve problems 10 times the size of the ones we struggle with today.

We propose several directions for future work. These include both extensions of chapters in this thesis as well as other areas that we did not have a chance to touch upon. We again emphasize that this list likely does not even begin to scratch the surface of all future possibilities.

8.9.1 Software Implementation

As previously stated in Section 8.4, it is possible to automatically generate the multistage stochastic programming models presented in this thesis given a deterministic model and data on the uncertainty. Similar capabilities for exogenous uncertainty already exist in software such as PySP (Watson et al., 2012), JuMP, GAMS (through EMP or DECIS), AIMMS (Roelofs and Bisschop, 2017), and FortSP (Zverovich et al., 2014). (Note that the last 3 options are proprietary software.) Although our work has been implemented in a fairly general sense in GAMS, we believe that one of the most important future steps is to build on top of the powerful open-source platform already offered by PySP. This will allow other users to easily experiment with our models and solution methods without spending months of valuable research time re-implementing all of our work. If we reduce the number of barriers here, we may also reach a wider audience. It is important to keep in mind that given the complexity of our theoretical reduction properties, and even the graph-theory algorithm, it is otherwise unlikely that very many researchers will make use of our work.

It is also worth noting that, as far as we are aware, current software implementations require user-supplied *scenario* data. It is possible to further automate this process by automatically generating scenario trees from *raw* data supplied by the user. Works of interest in this area include Calfa et al. (2014) for exogenous scenario-tree generation and Tarhan (2009) for endogenous scenario-tree generation.

8.9.2 Risk Aversion

An unfortunate side effect of naively optimizing for expected values is that the overall weighted average may be acceptable, but some of the individual components may not be. For example, in the case of minimizing total expected cost, a standard stochastic programming formulation can provide an optimal, and even excellent, objective function value while still containing some scenarios with very high costs. *On average*, the solution may be fine; but what happens if one of those undesirable scenarios becomes reality?

In many real-world applications, such *risk-neutral* approaches may carry far too much risk, even if the high-cost scenarios happen to have low probability. It is often desirable to instead compromise on the quality of the objective function value in exchange for a cost distribution with more desirable properties. This can be accomplished by including *risk measures* directly in the stochastic programming formulation. Some of the commonly used risk measures are variance, shortfall probability, expected shortage, value-at-risk, and conditional value-at-risk (see Oliveira et al., 2013, and the references therein). An excellent review of risk measures can also be found in Rockafellar (2007).

It is likely that such considerations will be necessary before our models can be used in any real-world setting. We are unaware of any works that explore risk measures in the context of multistage stochastic programming under exogenous *and* Type 2 endogenous uncertainties; however, risk management has been explored in Colvin and Maravelias (2011) for the case of Type 2 endogenous uncertainty, as well as in works by Escudero et al. for purely exogenous uncertainty and even exogenous and Type 1 endogenous uncertainties (see, for instance, Escudero et al., 2017, and Escudero et al., 2016a, respectively).

8.9.3 Gradual Resolution of Uncertainty

Throughout this thesis, we have assumed that endogenous uncertainties are resolved instantaneously, either right after an investment decision is made or after a pre-specified lead time. This is not always an accurate characterization, however. In oilfield development planning, for instance, production from a particular field may last several decades. It is likely unrealistic to assume that the recoverable oil volume becomes known as soon as production begins, or all at once after producing from the field for a few years. Instead, the uncertain recoverable oil volume gradually becomes more and more certain over time, and new decisions can be made as new information becomes available.

Gradual resolution has been considered in stochastic programs with endogenous uncertainty in Tarhan and Grossmann (2008), Tarhan et al. (2009), Solak et al. (2010), and Tarhan et al. (2013). Extending our work to accommodate gradual resolution will allow our models to be used for a more general class of problems. The downside of this extension, however, is that it will further complicate the structure of the scenario tree and make the model more complex (due to the need for additional disjunctive constraints to indicate which phase of the uncertainty resolution we are in at any given time).

8.9.4 Extension of Heuristic

Although the sequential scenario decomposition heuristic is capable of providing both upper and lower bounds, it is not an exact solution method that can be used in an iterative procedure to converge to the optimal solution. We believe that it is possible to extend this approach such that we can in fact obtain an exact, iterative algorithm.

The first modification for such an approach is to “fix” variables through explicit equality constraints rather than treating them as constants. Furthermore, we must consider *all* severed linking constraints, not only those for binary investment decisions. This step ensures that we have dual variables for all of the respective NACs. We can then execute the SSD algorithm iteratively in forward and backward passes, where we use these multipliers to generate cuts in each backward pass. The cuts accumulate as the iterations proceed. This general idea is based on recent work by Zou et al. (2016) and Lara et al. (2017), and is a variation of nested Benders decomposition/stochastic dual dynamic programming.

We note that this framework entails decomposing the original multistage stochastic program into purely-endogenous subproblems, and a separate decomposition algorithm is required to solve them. One possible avenue here is the endogenous scenario grouping Lagrangean decomposition approach proposed by Gupta and Grossmann (2014a). It may also be worthwhile to explore a new solution method altogether, since Lagrangean decomposition is often the default choice, and its suitability for large-scale problems of this class is questionable. Specifically, as the number of dualized constraints increases, we can easily end up with an unmanageable number of multipliers. (This particular criticism has been voiced by Mercier and Van Hentenryck, 2011.) Other options include revisiting the branch-and-cut strategy proposed by Colvin and Maravelias (2010), although we caution that this particular approach would be intended for the purely-endogenous subproblems only, and may not work well for the original problem due to the fact that there will be a very large number of exogenous NACs that likely cannot be relaxed. Clearly, the development of effective solution methods is still an open area of research.

An additional consideration for extending the heuristic is that we may adopt more rigorous criteria for selecting scenarios from the exogenous scenario groups in each time period. Specifically, in addition to the requirements specified in Section 4.3 (see the discussion following the statement of Property 5), it may be beneficial to select scenarios with the highest probability. It is also worth noting that it may be necessary to implement feasibility cuts to ensure that fixing binary decisions for time period t does not lead to infeasible subproblems in later time periods.

8.9.5 Other Possible Directions

We also propose the following possible directions, which are less developed but may hold future value. We include relevant references wherever possible.

- A compact representation (or implicit formulation) for first-period and exogenous NACs within scenario groups in Lagrangean decomposition (see Escudero et al., 2016b).

- Multistage stochastic programming under exogenous, Type 1 endogenous, and Type 2 endogenous uncertainties. (Note that using the terminology of Hellemo, 2016, this can be equivalently expressed as: "Multistage stochastic programming under exogenous and Type 3 endogenous uncertainties.")

- Strategically eliminating redundant non-anticipativity constraints to obtain the best possible structure. (Since scenario pairs are generally non-unique, we often have a choice as to which non-anticipativity constraints to remove. Some choices may lead to problem structures that are more favorable to MILP solvers. In some cases, it may even be preferable to leave some redundant constraints in the model. For example, in Goel and Grossmann, 2006, the authors recommend the use of redundant indistinguishability constraints in Lagrangean decomposition in order to obtain tighter lower bounds.)

- Exploiting matrix structure in multistage stochastic programs based on the location of uncertain parameters in the model (i.e., uncertainty in the objective function, constraint coefficients, or right-hand side).

- Endogenous uncertainty with continuous distributions. (In the words of Vayanos et al., 2011: "To the best of our knowledge, all existing algorithms rely on the assumption that the uncertain parameters follow a discrete distribution.")

- Time-varying endogenous uncertainty. (This addresses the case where the realization of an endogenous parameter changes over time. For example, due to shifting geological conditions, the recovery from a given oilfield may depend on when we decide to drill. If we drill next year, the recovery may be 5,000 bbl/day, 2,000 bbl/day, or 1,000 bbl/day. If we instead wait 5 years, the possible recovery values may have decreased to 2,500 bbl/day, 1,000 bbl/day, or 500 bbl/day, respectively. This can be interpreted as a problem with Type 1 and Type 2 endogenous uncertainties, since the decision to *not* invest changes the underlying probability distribution such that the 3 original realizations have been replaced by 3 less-desirable possibilities. Similar situations can arise in any case where the value of the endogenous information diminishes over time. In other words, in such cases, there is an added incentive to make investments early in the planning horizon. The example described here can be modeled easily with stochastic programming, since we can simply multiply the respective endogenous parameter by a time-dependent scaling factor directly in the model.)

- Resolving endogenous uncertainties with continuous decisions rather than binary investment decisions. (As stated by Woodruff, 2003: "Although extension to real variables is possible, all work to date has focused on integer decisions that [affect] discovery timing." This extension can be accomplished fairly easily in our models through a custom uncertainty-resolution rule. Specifically, rather than checking the value of a binary decision variable, we can instead check the value of a separate binary variable that will only transition from 0 to 1 if a continuous variable has reached a certain threshold.)

Bibliography

S. Ahmed. *Strategic Planning under Uncertainty: Stochastic Integer Programming Approaches*. PhD thesis, University of Illinois at Urbana-Champaign, 2000.

R. M. Apap and I. E. Grossmann. A Graph-theory Approach for Efficient Non-anticipativity-constraint Reduction in Multistage Stochastic Programming Problems Involving Endogenous and Exogenous Uncertainties. In *AIChE Annual Meeting*, Salt Lake City, UT, 2015. https://www.aiche.org/conferences/aiche-annual-meeting/2015/proceeding/paper/282e-graph-theory-approach-efficient-non-anticipativity-constraint-reduction-multistage-stoc

E. M. L. Beale. On Minimizing a Convex Function Subject to Linear Inequalities. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 17(2): 173–184, 1955.

A. Ben-Tal, A. Goryashko, E. Guslitzer, and A. Nemirovski. Adjustable Robust Solutions of Uncertain Linear Programs. *Mathematical Programming*, 99(2, Ser. B): 351–376, 2004.

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

J. R. Birge. Stochastic Programming Computation and Applications. *INFORMS Journal on Computing*, 9(2): 111–133, 1997.

J. R. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer Science+Business Media, New York, NY, 2nd edition, 2011.

N. Boland, I. Dumitrescu, and G. Froyland. A Multistage Stochastic Programming Approach to Open Pit Mine Production Scheduling with Uncertain Geology. 2008. http://www.optimization-online.org/DB_FILE/2008/10/2123.pdf.

N. Boland, I. Dumitrescu, G. Froyland, and T. Kalinowski. Minimum Cardinality Non-anticipativity Constraint Sets for Multistage Stochastic Programming. *Mathematical Programming*, 157(1, Ser. B): 69–93, 2016.

M. E. Bruni, P. Beraldi, and D. Conforti. A Stochastic Programming Approach for Operating Theatre Scheduling under Uncertainty. *IMA Journal of Management Mathematics*, 26(1): 99–119, 2015.

B. A. Calfa, A. Agarwal, I. E. Grossmann, and J. M. Wassick. Data-driven Multi-stage Scenario Tree Generation via Statistical Property and Distribution Matching. *Computers & Chemical Engineering*, 68: 7–23, 2014.

B. A. Calfa, I. E. Grossmann, A. Agarwal, S. J. Bury, and J. M. Wassick. Data-driven Individual and Joint Chance-constrained Optimization via Kernel Smoothing. *Computers & Chemical Engineering*, 78: 51–69, 2015.

C. C. Carøe and R. Schultz. Dual Decomposition in Stochastic Integer Programming. *Operations Research Letters*, 24(1–2): 37–45, 1999.

D. R. Cariño, T. Kent, D. H. Myers, C. Stacy, M. Sylvanus, A. L. Turner, K. Watanabe, and W. T. Ziemba. The Russell-Yasuda Kasai Model: An Asset/Liability Model for a Japanese Insurance Company Using Multistage Stochastic Programming. *Interfaces*, 24(1): 29–49, 1994.

J. Choi, M. J. Realff, and J. H. Lee. Dynamic Programming in a Heuristically Confined State Space: A Stochastic Resource-constrained Project Scheduling Application. *Computers & Chemical Engineering*, 28(6–7): 1039–1058, 2004.

B. Christian and S. Cremaschi. Heuristic Solution Approaches to the Pharmaceutical R&D Pipeline Management Problem. *Computers & Chemical Engineering*, 74: 34–47, 2015.

B. Christian and S. Cremaschi. A Graph Theory Approach to Non-anticipativity Constraint Generation in Multistage Stochastic Programs with Incomplete Scenario Sets. In *AIChE Annual Meeting*, San Francisco, CA, 2016. https://www.aiche.org/conferences/aiche-annual-meeting/2016/proceeding/paper/682e-graph-theory-approach-non-anticipativity-constraint-generation-multistage-stochastic-pr

B. Christian and S. Cremaschi. Variants to a Knapsack Decomposition Heuristic for Solving R&D Pipeline Management Problems. *Computers & Chemical Engineering*, 96: 18–32, 2017.

M. Colvin and C. T. Maravelias. A Stochastic Programming Approach for Clinical Trial Planning in New Drug Development. *Computers & Chemical Engineering*, 32(11): 2626–2642, 2008.

M. Colvin and C. T. Maravelias. Scheduling of Testing Tasks and Resource Planning in New Product Development using Stochastic Programming. *Computers & Chemical Engineering*, 33 (5): 964–976, 2009.

M. Colvin and C. T. Maravelias. Modeling Methods and a Branch and Cut Algorithm for Pharmaceutical Clinical Trial Planning using Stochastic Programming. *European Journal of Operational Research*, 203(1): 205–215, 2010.

M. Colvin and C. T. Maravelias. R&D Pipeline Management: Task Interdependencies and Risk Management. *European Journal of Operational Research*, 215(3): 616–628, 2011.

D. B. Dantzig. Linear Programming under Uncertainty. *Management Science*, 1(3–4): 197–206, 1955.

J. Dupačová. Optimization under Exogenous and Endogenous Uncertainty. In *Proceedings of the 24th International Conference on Mathematical Methods in Economics*, pages 131–136, University of West Bohemia, Pilsen, Czech Republic, 2006.

L. F. Escudero, M. A. Garín, M. Merino, and G. Pérez. The Value of the Stochastic Solution in Multistage Problems. *TOP*, 15(1): 48–64, 2007.

L. F. Escudero, M. A. Garín, M. Merino, and G. Pérez. BFC-MSMIP: An Exact Branch-and-fix Coordination Approach for Solving Multistage Stochastic Mixed 0-1 Problems. *TOP*, 17(1): 96–122, 2009.

L. F. Escudero, M. A. Garín, M. Merino, and G. Pérez. On Multistage Mixed 0-1 Optimization under a Mixture of Exogenous and Endogenous Uncertainty in a Risk Averse Environment. In *XIII International Conference on Stochastic Programming*, Bergamo, Italy, 2013a. http://dinamico2.unibg.it/icsp2013/doc/ms/4%20ICSP_escudero.pdf.

L. F. Escudero, M. A. Garín, G. Pérez, and A. Unzueta. Scenario Cluster Decomposition of the Lagrangian Dual in Two-stage Stochastic Mixed 0-1 Optimization. *Computers & Operations Research*, 40(1): 362–377, 2013b.

L. F. Escudero, M. A. Garín, J. F. Monge, and A. Unzueta. On Preparedness Resource Allocation Planning for Natural Disaster Relief by Multistage Stochastic Mixed 0-1 Bilinear Optimization Based on Endogenous Uncertainty and Time Consistent Risk Averse Management. 2016a. Submitted for publication.

L. F. Escudero, M. A. Garín, and A. Unzueta. Cluster Lagrangean Decomposition in Multistage Stochastic Optimization. *Computers & Operations Research*, 67: 48–62, 2016b.

L. F. Escudero, M. A. Garín, and A. Unzueta. Scenario Cluster Lagrangean Decomposition for Risk Averse in Multistage Stochastic Optimization. *Computers & Operations Research*, 85: 154–171, 2017.

M. L. Fisher. An Applications Oriented Guide to Lagrangian Relaxation. *Interfaces*, 15(2): 10–21, 1985.

B. Flach. *Stochastic Programming with Endogenous Uncertainty: An Application in Humanitarian Logistics*. PhD thesis, Pontifical Catholic University of Rio de Janeiro, 2010.

V. Goel and I. E. Grossmann. A Stochastic Programming Approach to Planning of Offshore Gas Field Developments under Uncertainty in Reserves. *Computers & Chemical Engineering*, 28(8): 1409–1429, 2004.

V. Goel and I. E. Grossmann. A Class of Stochastic Programs with Decision Dependent Uncertainty. *Mathematical Programming*, 108(2–3, Ser. B): 355–394, 2006.

V. Goel, I. E. Grossmann, A. S. El-Bakry, and E. L. Mulkay. A Novel Branch and Bound Algorithm for Optimal Development of Gas Fields under Uncertainty in Reserves. *Computers & Chemical Engineering*, 30(6–7): 1076–1092, 2006.

H. J. Greenberg. The ANALYZE Rulebase for Supporting LP Analysis. *Annals of Operations Research*, 65(1): 91–126, 1996.

I. E. Grossmann, R. M. Apap, B. A. Calfa, P. García-Herreros, and Q. Zhang. Recent Advances in Mathematical Programming Techniques for the Optimization of Process Systems under Uncertainty. *Computers & Chemical Engineering*, 91: 3–14, 2016.

M. Guignard. Lagrangean Relaxation. *TOP*, 11(2): 151–228, 2003.

V. Gupta and I. E. Grossmann. Solution Strategies for Multistage Stochastic Programming with Endogenous Uncertainties. *Computers & Chemical Engineering*, 35(11): 2235–2247, 2011.

V. Gupta and I. E. Grossmann. A New Decomposition Algorithm for Multistage Stochastic Programs with Endogenous Uncertainties. *Computers & Chemical Engineering*, 62: 62–79, 2014a.

V. Gupta and I. E. Grossmann. Multistage Stochastic Programming Approach for Offshore Oilfield Infrastructure Planning under Production Sharing Agreements and Endogenous Uncertainties. *Journal of Petroleum Science and Engineering*, 124: 180–197, 2014b.

H. Held and D. L. Woodruff. Heuristics for Multi-stage Interdiction of Stochastic Networks. *Journal of Heuristics*, 11(5): 483–500, 2005.

L. Hellemo. *Managing Uncertainty in Design and Operation of Natural Gas Infrastructure*. PhD thesis, Norwegian University of Science and Technology, 2016.

J. L. Higle. Stochastic Programming: Optimization When Uncertainty Matters. In *INFORMS Tutorials in Operations Research: Emerging Theory, Methods, and Applications*, chapter 2, pages 30–53. 2005.

F. Hooshmand Khaligh and S. A. MirHassani. A Mathematical Model for Vehicle Routing Problem under Endogenous Uncertainty. *International Journal of Production Research*, 54(2): 579–590, 2016a.

F. Hooshmand Khaligh and S. A. MirHassani. Efficient Constraint Reduction in Multistage Stochastic Programming Problems with Endogenous Uncertainty. *Optimization Methods and Software*, 31(2): 359–376, 2016b.

T. W. Jonsbråten. *Optimization Models for Petroleum Field Exploitation*. PhD thesis, Norwegian School of Economics and Business Administration, 1998.

T. W. Jonsbråten, R. J.-B. Wets, and D. L. Woodruff. A Class of Stochastic Programs with Decision Dependent Random Elements. *Annals of Operations Research*, 82: 83–106, 1998.

P. Kall and S. W. Wallace. *Stochastic Programming*. John Wiley & Sons, Chichester, 1994.

A. J. King and S. W. Wallace. *Modeling with Stochastic Programming*. Springer Science+Business Media, New York, NY, 2012.

N. Lappas and C. Gounaris. Multi-stage Adjustable Robust Optimization for Process Scheduling under Uncertainty. *AIChE Journal*, 62(5): 1646–1667, 2016.

C. L. Lara, D. S. Mallapragada, D. J. Papageorgiou, A. Venkatesh, and I. E. Grossmann. MILP Formulation and Nested Decomposition Algorithm for Planning of Electric Power Infrastructures. 2017. Manuscript in preparation.

M. Laumanns, S. Prestwich, and B. Kawas. Distribution Shaping and Scenario Bundling for Stochastic Programs with Endogenous Uncertainty. *Stochastic Programming E-Print Series*, 2014. http://edoc.hu-berlin.de/series/speps/2014-5/PDF/5.pdf.

P. Li, H. Arellano-Garcia, and G. Wozny. Chance Constrained Programming Approach to Process Optimization under Uncertainty. *Computers & Chemical Engineering*, 32(1–2): 25–45, 2008.

M. L. Liu and N. V. Sahinidis. Optimization in Process Planning under Uncertainty. *Industrial and Engineering Chemistry Research*, 35(11): 4154–4165, 1996.

X. Liu, S. Küçükyavuz, and J. Luedtke. Decomposition Algorithms for Two-stage Chance-constrained Programs. *Mathematical Programming*, 157(1, Ser. B): 219–243, 2016.

F. Maggioni, E. Allevi, and M. Bertocchi. Bounds in Multistage Linear Stochastic Programming. *Journal of Optimization Theory and Applications*, 163(1): 200–229, 2014.

L. Mercier and P. Van Hentenryck. An Anytime Multistep Anticipatory Algorithm for Online Stochastic Combinatorial Optimization. *Annals of Operations Research*, 184(1): 233–271, 2011.

F. Oliveira, V. Gupta, S. Hamacher, and I. E. Grossmann. A Lagrangean Decomposition Approach for Oil Supply Chain Investment Planning under Uncertainty with Risk Considerations. *Computers & Chemical Engineering*, 50: 184–195, 2013.

S. Peeta, F. S. Salman, D. Gunnec, and K. Viswanath. Pre-disaster Investment Decisions for Strengthening a Highway Network. *Computers & Operations Research*, 37(10): 1708–1719, 2010.

G. Ch. Pflug. On-line Optimization of Simulated Markovian Processes. *Mathematics of Operations Research*, 15(3): 381–395, 1990.

W. B. Powell. *Approximate Dynamic Programming: Solving the Curses of Dimensionality.* John Wiley & Sons, Hoboken, NJ, 2nd edition, 2011.

W. B. Powell. Clearing the Jungle of Stochastic Optimization. In *INFORMS Tutorials in Operations Research: Bridging Data and Decisions*, chapter 4, pages 109–137. 2014.

R. Raman and I. E. Grossmann. Relation Between MILP Modelling and Logical Inference for Chemical Process Synthesis. *Computers & Chemical Engineering*, 15(2): 73–84, 1991.

R. T. Rockafellar. Coherent Approaches to Risk in Optimization under Uncertainty. In *INFORMS Tutorials in Operations Research: OR Tools and Applications: Glimpses of Future Technologies*, chapter 3, pages 38–61. 2007.

R. T. Rockafellar and R. J.-B. Wets. Scenarios and Policy Aggregation in Optimization under Uncertainty. *Mathematics of Operations Research*, 16(1): 119–147, 1991.

M. Roelofs and J. Bisschop. *AIMMS: The Language Reference*. AIMMS B.V., 2017. https://download.aimms.com/aimms/download/manuals/AIMMS3_LR.pdf.

A. Ruszczyński. Decomposition Methods in Stochastic Programming. *Mathematical Programming*, 79(1–3): 333–353, 1997.

N. V. Sahinidis. Optimization under Uncertainty: State-of-the-art and Opportunities. *Computers & Chemical Engineering*, 28(6–7): 971–983, 2004.

S. Sashihara. *The Optimization Edge: Reinventing Decision Making to Maximize All Your Company's Assets*. McGraw-Hill Education, USA, 2011.

R. Schultz. Stochastic Programming with Integer Variables. *Mathematical Programming*, 97(1–2, Ser. B): 285–309, 2003.

S. Sen and J. L. Higle. An Introductory Tutorial on Stochastic Linear Programming Models. *Interfaces*, 29(2): 33–61, 1999.

A. Shapiro and A. Philpott. A Tutorial on Stochastic Programming. 2007. http://www2.isye.gatech.edu/people/faculty/Alex_Shapiro/TutorialSP.pdf.

S. Solak, J.-P. B. Clarke, E. L. Johnson, and E. R. Barnes. Optimization of R&D Project Portfolios under Endogenous Uncertainty. *European Journal of Operational Research*, 207(1): 420–433, 2010.

B. Tarhan. *Stochastic Programming Approaches for Decision-dependent Uncertainty and Gradual Uncertainty Resolution*. PhD thesis, Carnegie Mellon University, 2009.

B. Tarhan and I. E. Grossmann. A Multistage Stochastic Programming Approach with Strategies for Uncertainty Reduction in the Synthesis of Process Networks with Uncertain Yields. *Computers & Chemical Engineering*, 32(4–5): 766–788, 2008.

B. Tarhan, I. E. Grossmann, and V. Goel. Stochastic Programming Approach for the Planning of Offshore Oil or Gas Field Infrastructure under Decision-dependent Uncertainty. *Industrial and Engineering Chemistry Research*, 48(6): 3078–3097, 2009.

B. Tarhan, I. E. Grossmann, and V. Goel. Computational Strategies for Non-convex Multistage MINLP Models with Decision-dependent Uncertainty and Gradual Uncertainty Resolution. *Annals of Operations Research*, 203(1): 141–166, 2013.

S. Terrazas-Moreno, I. E. Grossmann, J. M. Wassick, S. J. Bury, and N. Akiya. An Efficient Method for Optimal Design of Large-scale Integrated Chemical Production Sites with Endogenous Uncertainty. *Computers & Chemical Engineering*, 37: 89–103, 2012.

K. Tong, Y. Feng, and G. Rong. Planning under Demand and Yield Uncertainties in an Oil Supply Chain. *Industrial and Engineering Chemistry Research*, 51(2): 814–834, 2012.

F. Trespalacios and I. E. Grossmann. Review of Mixed-integer Nonlinear and Generalized Disjunctive Programming Methods. *Chemie Ingenieur Technik*, 86(7): 991–1012, 2014.

P. Vayanos, D. Kuhn, and B. Rustem. Decision Rules for Information Discovery in Multi-stage Stochastic Programming. In *Proceedings of the 2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 7368–7373, Orlando, FL, USA, 2011.

K. Viswanath, S. Peeta, and F. S. Salman. Investing in the Links of a Stochastic Network to Minimize Expected Shortest Path Length. Technical report, Purdue University, 2004. https://www.krannert.purdue.edu/programs/phd/working-papers-series/2004/1167.pdf.

S. W. Wallace and W. T. Ziemba. *Applications of Stochastic Programming*. Society for Industrial and Applied Mathematics and the Mathematical Programming Society, Philadelphia, PA, 2005.

J.-P. Watson, D. L. Woodruff, and W. E. Hart. PySP: Modeling and Solving Stochastic Programs in Python. *Mathematical Programming Computation*, 4(2): 109–149, 2012.

M. Webster, N. Santen, and P. Parpas. An Approximate Dynamic Programming Framework for Modeling Global Climate Policy under Decision-dependent Uncertainty. *Computational Management Science*, 9(3): 339–362, 2012.

H. P. Williams. *Model Building in Mathematical Programming*. John Wiley & Sons, West Sussex, England, 5th edition, 2013.

D. L. Woodruff. Stochastic Programming Models with Decision Dependent Probabilities. In *INFORMS Annual Meeting*, Atlanta, GA, USA, 2003. http://www.maximalsoftware.com/slides/Atl03Woodruff/stochslides.pdf.

F. You, J. M. Wassick, and I. E. Grossmann. Risk Management for a Global Supply Chain Planning under Uncertainty: Models and Algorithms. *AIChE Journal*, 55(4): 931–946, 2009.

Q. Zhang, R. M. Lima, and I. E. Grossmann. On the Relation between Flexibility Analysis and Robust Optimization for Linear Systems. *AIChE Journal*, 62(9): 3109–3123, 2016.

J. Zou, S. Ahmed, and X. A. Sun. Nested Decomposition of Multistage Stochastic Integer Programs with Binary State Variables. 2016. Submitted for publication.

V. Zverovich, C. A. Valle, F. Ellison, and G. Mitra. *FortSP: A Stochastic Programming Solver*. OptiRisk Systems, 2014. http://www.optirisk-systems.com/manuals/FortspManual.pdf.

Appendices

Appendix A Supplementary Material for Chapter 2

A.1 Correlated Parameters

In Chapter 2, we state that uncertain parameters $\xi_{j,t}$ and $\theta_{i,h}$ are assumed to be independent. Simple correlations between the parameters do not preclude the use of a Cartesian product, however. This is for the simple reason that if we have an exogenous-uncertain parameter ψ_t that is correlated with $\xi_{\hat{\jmath},t}$, we can use the known correlation to explicitly express ψ_t as a function of the other uncertain parameter; i.e., $\psi_t^s = f(\xi_{\hat{\jmath},t}^s)\ \forall\ s \in \mathcal{S}$. This means that relative to parameter $\xi_{\hat{\jmath},t}$, the value of ψ_t is known. Furthermore, by using a function to express the value of ψ_t for each scenario, we can exclude this parameter from the scenario-generation process. The same argument applies for an endogenous-uncertain parameter ϕ that is correlated with $\theta_{\hat{\imath},\hat{h}}$, as we can state $\phi^s = f(\theta_{\hat{\imath},\hat{h}}^s)\ \forall\ s \in \mathcal{S}$.

If, however, there are many uncertain parameters with complex correlations that the modeler may not even be aware of, then the Cartesian-product approach should not be used. Using the exogenous case as an example, recall that there is only *one* corresponding realization of ψ_t for each realization of $\xi_{\hat{\jmath},t}$ (since the two parameters are correlated). If we were to generate all possible combinations of realizations for $\xi_{\hat{\jmath},t}$ and ψ_t, we would end up with many scenarios that reflect impossible outcomes. In fact, even the realistic scenarios would be improperly weighted because the calculated probabilities would be incorrect. It is for these reasons that we restrict the scenario-generation process to independent parameters.

A.2 Alternate Composite Scenario Tree

When we generate the composite scenario tree by the Cartesian product $\mathcal{R}_X \times \mathcal{R}_N$, instead of $\mathcal{R}_N \times \mathcal{R}_X$, we produce the scenario tree shown in Figure A.1.

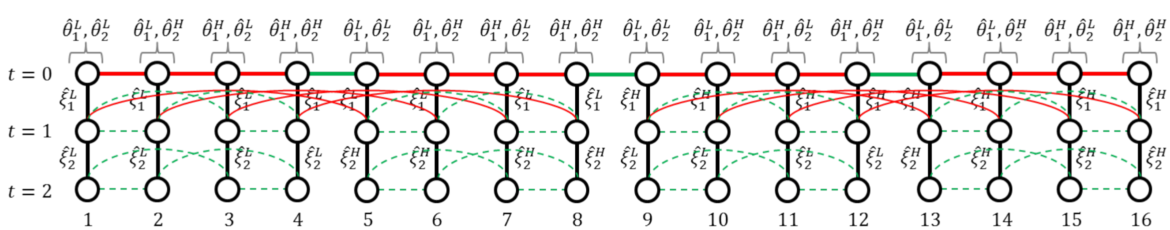


Figure A.1: Alternate composite scenario tree.

Notice that this tree lacks a unique structure that we can easily exploit.

A.3 Proof of 'Scenario Probabilities Must Sum to 1'

Starting with Equation (2.10) and taking the product of all elements in each tuple, then summing over all of these products, we can factor out like-terms to arrive at the following expression:

$$\begin{aligned}\sum_{s\in\mathcal{S}} p^s = \hat{\omega}_{1,1}^1 \Big[\hat{\omega}_{I,H_I}^1 \Big(\hat{\upsilon}_{1,1}^1 \Big[\hat{\upsilon}_{J,T}^1 + \cdots + \hat{\upsilon}_{J,T}^{R_{J,T}}\Big] + \cdots + \hat{\upsilon}_{1,1}^{R_{1,1}} \Big[\hat{\upsilon}_{J,T}^1 + \cdots + \hat{\upsilon}_{J,T}^{R_{J,T}}\Big]\Big) + \cdots + \\ \hat{\omega}_{I,H_I}^{M_{I,H_I}} \Big(\hat{\upsilon}_{1,1}^1 \Big[\hat{\upsilon}_{J,T}^1 + \cdots + \hat{\upsilon}_{J,T}^{R_{J,T}}\Big] + \cdots + \hat{\upsilon}_{1,1}^{R_{1,1}} \Big[\hat{\upsilon}_{J,T}^1 + \cdots + \hat{\upsilon}_{J,T}^{R_{J,T}}\Big]\Big)\Big] + \cdots + \\ \hat{\omega}_{1,1}^{M_{1,1}} \Big[\hat{\omega}_{I,H_I}^1 \Big(\hat{\upsilon}_{1,1}^1 \Big[\hat{\upsilon}_{J,T}^1 + \cdots + \hat{\upsilon}_{J,T}^{R_{J,T}}\Big] + \cdots + \hat{\upsilon}_{1,1}^{R_{1,1}} \Big[\hat{\upsilon}_{J,T}^1 + \cdots + \hat{\upsilon}_{J,T}^{R_{J,T}}\Big]\Big) + \cdots + \\ \hat{\omega}_{I,H_I}^{M_{I,H_I}} \Big(\hat{\upsilon}_{1,1}^1 \Big[\hat{\upsilon}_{J,T}^1 + \cdots + \hat{\upsilon}_{J,T}^{R_{J,T}}\Big] + \cdots + \hat{\upsilon}_{1,1}^{R_{1,1}} \Big[\hat{\upsilon}_{J,T}^1 + \cdots + \hat{\upsilon}_{J,T}^{R_{J,T}}\Big]\Big)\Big]\end{aligned} \tag{A.1}$$

For ease of exposition, we consider only the realization probabilities associated with the first and last endogenous parameters ($\theta_{1,1}$ and θ_{I,H_I}, respectively) and the first and last exogenous parameters ($\xi_{1,1}$ and $\xi_{J,T}$, respectively). Since $\sum_{r=1}^{R_{j,t}} \hat{\upsilon}_{j,t}^r = 1 \;\forall\, j \in \mathcal{J},\, t \in \mathcal{T}$, we have $\hat{\upsilon}_{J,T}^1 + \cdots + \hat{\upsilon}_{J,T}^{R_{J,T}} = 1$, which reduces this expression to:

$$\begin{aligned}\sum_{s\in\mathcal{S}} p^s = \hat{\omega}_{1,1}^1 \Big[\hat{\omega}_{I,H_I}^1 \Big(\hat{\upsilon}_{1,1}^1 + \cdots + \hat{\upsilon}_{1,1}^{R_{1,1}}\Big) + \cdots + \hat{\omega}_{I,H_I}^{M_{I,H_I}} \Big(\hat{\upsilon}_{1,1}^1 + \cdots + \hat{\upsilon}_{1,1}^{R_{1,1}}\Big)\Big] + \cdots + \\ \hat{\omega}_{1,1}^{M_{1,1}} \Big[\hat{\omega}_{I,H_I}^1 \Big(\hat{\upsilon}_{1,1}^1 + \cdots + \hat{\upsilon}_{1,1}^{R_{1,1}}\Big) + \cdots + \hat{\omega}_{I,H_I}^{M_{I,H_I}} \Big(\hat{\upsilon}_{1,1}^1 + \cdots + \hat{\upsilon}_{1,1}^{R_{1,1}}\Big)\Big]\end{aligned} \tag{A.2}$$

By the same reasoning, $\hat{\upsilon}_{1,1}^1 + \cdots + \hat{\upsilon}_{1,1}^{R_{1,1}} = 1$, which gives:

$$\sum_{s\in\mathcal{S}} p^s = \hat{\omega}_{1,1}^1 \Big[\hat{\omega}_{I,H_I}^1 + \cdots + \hat{\omega}_{I,H_I}^{M_{I,H_I}}\Big] + \cdots + \hat{\omega}_{1,1}^{M_{1,1}} \Big[\hat{\omega}_{I,H_I}^1 + \cdots + \hat{\omega}_{I,H_I}^{M_{I,H_I}}\Big] \tag{A.3}$$

Now, since $\sum_{m=1}^{M_{i,h}} \hat{\omega}_{i,h}^m = 1 \;\forall\, i \in \mathcal{I},\, h \in \mathcal{H}_i$, we have $\hat{\omega}_{I,H_I}^1 + \cdots + \hat{\omega}_{I,H_I}^{M_{I,H_I}} = 1$, which leaves:

$$\sum_{s\in\mathcal{S}} p^s = \hat{\omega}_{1,1}^1 + \cdots + \hat{\omega}_{1,1}^{M_{1,1}} = 1 \quad \square \tag{A.4}$$

Appendix B Supplementary Material for Chapter 4

B.1 Proof of Property 1

Consider two indistinguishable scenarios $\hat{s}, \hat{s}' \in \mathcal{S}$ in time period τ, where $\hat{s} < \hat{s}'$. For simplicity, consider only variables $y_\tau^{\hat{s}}$ and $y_\tau^{\hat{s}'}$. By Equation (4.1), we generate two scenario pairs: $(\hat{s}, \hat{s}')$ and $(\hat{s}', \hat{s})$. Scenario pair $(\hat{s}, \hat{s}')$ corresponds to non-anticipativity constraint $y_\tau^{\hat{s}} = y_\tau^{\hat{s}'}$. Scenario pair $(\hat{s}', \hat{s})$ corresponds to non-anticipativity constraint $y_\tau^{\hat{s}'} = y_\tau^{\hat{s}}$, which is the same equality constraint. By symmetry, we may replace the condition $s \neq s'$ in Equation (4.1) with $s < s'$. In this case, we only generate the first pair, and we avoid the second, redundant constraint. $\square$

B.2 Proof of Property 2a

Because *all* scenarios are indistinguishable at the beginning of the first time period, *adjacent* scenarios must also be indistinguishable at that time. Thus, we can enforce non-anticipativity between all scenarios by linking consecutive nodes; e.g., $y_1^1 = y_1^2,\ y_1^2 = y_1^3, \ldots, y_1^{S-1} = y_1^S$. $\square$

B.3 Proof of Proposition 1

We generate set $\mathcal{SP}_F$ by pairing off all S scenarios in consecutive order. This gives $S-1$ independent links (i.e., scenario pairs), which is the minimum number of links required to connect S elements. $\square$

B.4 Proof of Property 2b

As previously stated, exogenous NACs apply only between scenarios s and s' in the same subtree. Because each subtree represents an exogenous scenario tree, non-anticipativity constraints within that tree apply as if the uncertainty were purely exogenous. (Note that in the case of purely-exogenous uncertainty, the adjacent-scenario approach to non-anticipativity is well known (see, for example, Colvin and Maravelias, 2011). However, we provide the rest of the proof for completeness.)

Accordingly, consider an exogenous scenario tree in its standard form, as shown in Figure 1.2a. For each time period $t \in \mathcal{T},\ t < T$, each scenario passes through a node that is shared among one or more scenarios. All scenarios that pass through one such node at time t must be indexed consecutively, since they all refer to the same path up until this time (i.e., they have the same history). When we duplicate this node to give each scenario its own respective copy, we create consecutive, indistinguishable nodes that refer to the same state and must be linked together with non-anticipativity constraints (see Figure 1.2b). One natural approach to enforce non-anticipativity between these indistinguishable scenarios in time period t is to link them together in consecutive order. $\square$

B.5 Proof of Proposition 2

In each time period, excluding $t = T$, we partition the set of scenarios into exogenous scenario-group subsets ${}^X\mathcal{G}_t^k\ \forall\ k \in \mathcal{K}_t$. Since each scenario must be assigned to one group (scenario 1 to group 1, and all others by Equation (4.5)), the union of all such groups in each time period must give the complete set of scenarios; i.e.,

$$\bigcup_{k \in \mathcal{K}_t} {}^X\mathcal{G}_t^k = \mathcal{S} \qquad \forall\, t \in \mathcal{T} \setminus \{T\} \tag{B.1}$$

Thus, we are considering all scenarios in each time period where exogenous NACs apply.

We enforce non-anticipativity between consecutive scenarios in each of the exogenous scenario groups. By Equation (3.11), the corresponding scenario pairs for $t \in \mathcal{T} \setminus \{T\}$ must be in set $\mathcal{SP}_X$ because they are adjacent (i.e., $(s, s') \in \mathcal{A}$), in the same subtree (i.e., $Sub(s) = Sub(s')$), and indistinguishable (i.e., $Q_t^{s,s'} = True$). Since each group has different realizations for the exogenous parameters and/or different possible realizations for the endogenous parameters, no links between the groups are possible, and such pairs cannot be in set $\mathcal{SP}_X$. Thus, the pairs in each group are the *only* possible exogenous scenario pairs in each time period. It follows that the union of these sets of tuples must be equivalent to set $\mathcal{SP}_X$:

$$\bigcup_{t\in\mathcal{T}\setminus\{T\}}\left[\bigcup_{k\in\mathcal{K}_t}\left\{(t,s,s'):\ s,s'\in {}^X\mathcal{G}_t^k,\ \ (s,s')\in\mathcal{A}\right\}\right]=\mathcal{SP}_X \tag{B.2}$$

Now, consider the scenario pairs in each exogenous scenario group in time period $t \in \mathcal{T}\setminus\{T\}$. Because we link consecutive scenarios, this gives $|{}^X\mathcal{G}_t^k|-1$ scenario pairs in each group, which is the minimum number of links required to connect $|{}^X\mathcal{G}_t^k|$ elements. These pairs cannot be implied through the use of any endogenous scenario pairs, since we generate the exogenous pairs first. Thus, we have the minimum number of scenario pairs in each group. We have shown that these are the only possible exogenous scenario pairs in each time period, and the union of these sets of tuples is equivalent to set $\mathcal{SP}_X$. Hence, set $\mathcal{SP}_X$ contains the minimum number of exogenous scenario pairs. $\square$

B.6 Proof of Property 4

By Property 3, endogenous NACs are expressed between scenarios s and s' that differ in the possible realization of a single endogenous parameter $\theta_{i',h}$ and are identical in the realizations of all exogenous parameters in all time periods. Accordingly, for each $i'\in\mathcal{I}$ and $h\in\mathcal{H}_{i'}$, we seek to identify the minimum number of scenario pairs (s,s') that satisfy these conditions.

To this end, in an arbitrary time period $t=\tau$, we partition the set of scenarios into endogenous scenario-group subsets. These subsets are given by ${}^N\mathcal{G}_{i',h}^l$ and are indexed by $l\in\mathcal{L}_{i',h}$ for each $i'\in\mathcal{I}$ and $h\in\mathcal{H}_{i'}$. By the endogenous scenario-group algorithm, each scenario must be assigned to one such group for each $i'\in\mathcal{I}$ and $h\in\mathcal{H}_{i'}$. In other words, the union of all of these groups must give the complete set of scenarios; i.e.,

$$\bigcup_{l\in\mathcal{L}_{i',h}} {}^N\mathcal{G}_{i',h}^l=\mathcal{S} \qquad \forall\ i'\in\mathcal{I},\ \ h\in\mathcal{H}_{i'} \tag{B.3}$$

Thus, we are considering all scenarios in each case where endogenous NACs may apply.

We enforce non-anticipativity between consecutive scenarios in each of the endogenous scenario groups, as indicated by Equation (4.15). This gives $|{}^N\mathcal{G}_{i',h}^l|-1$ scenario pairs in each group, which is the minimum number of links required to connect $|{}^N\mathcal{G}_{i',h}^l|$ elements. Other connections between the scenarios are implied by transitivity.

Furthermore, by Property 3, it is sufficient to consider *only* the pairs formed in each endogenous scenario group. This is because there are no links between groups, other than those that already exist in another group. We can prove this by contradiction.

First, suppose that we have a link between two scenarios, $\hat{s}$ and $\hat{s}'$. By Property 3, these scenarios must differ only in the possible realization of a single endogenous parameter $\theta_{\hat{\imath},\hat{h}}$. Second, assume that this link cannot be formed by pairing two scenarios in the same endogenous scenario group. In other words, these scenarios belong to two separate groups corresponding to parameter $\theta_{\hat{\imath},\hat{h}}$, and by the endogenous scenario-group algorithm, we have $G_N(\hat{\imath},\hat{h},\hat{s})=\hat{l}$ and $G_N(\hat{\imath},\hat{h},\hat{s}')=\tilde{l}$. It follows that the respective groups from Equation (4.14) are ${}^N\mathcal{G}_{\hat{\imath},\hat{h}}^{\hat{l}}$ and ${}^N\mathcal{G}_{\hat{\imath},\hat{h}}^{\tilde{l}}$. Note that the scenarios in one group must differ from the scenarios in the other group in terms of the possible realization of *at least one* uncertain parameter other than $\theta_{\hat{\imath},\hat{h}}$. (If this were not the case, then ${}^{N}\mathcal{G}^{\hat{l}}_{\hat{\imath},\hat{h}}$ and ${}^{N}\mathcal{G}^{\tilde{l}}_{\hat{\imath},\hat{h}}$ would be a single group.) Since $\hat{s}$ and $\hat{s}'$ differ in the possible realization of $\theta_{\hat{\imath},\hat{h}}$ *and* belong to two separate groups, these scenarios must differ in the possible realizations of *at least two* uncertain parameters. This violates Property 3. Thus, the original assumption is false and any endogenous scenario pair must be formed between two scenarios in the same endogenous scenario group.

At this point, we have shown that if we consider only scenario pairs (s, s') for which s and s' are consecutive scenarios in an endogenous scenario group: (1) we are able to link all scenarios in each group; and (2) from this linking, we are able to produce all endogenous scenario pairs generated by Property 3 (either explicitly, or implicitly through the use of some explicitly-generated pairs). Note that since the endogenous scenario groups are defined in terms of Property 3, Property 4 must also be *at least* as restrictive as Property 3. In other words, $\mathcal{SP}_{N^4} \subseteq \mathcal{SP}_{N^3}$, and this approach cannot produce any additional scenario pairs that cannot be obtained from Property 3. It follows that Property 4 is a sufficient condition for endogenous scenario-pair generation. $\square$

B.7 Proof of Proposition 3

This is the case described in Gupta and Grossmann (2011). We approach this proof from a different angle and continue from the proof of Property 4.

So far, we have shown that Property 4 gives the minimum number of pairs among the scenarios in each endogenous scenario group, and that it is sufficient to consider only these pairs. Recall that by Equation (4.15), we generate one such set of pairs for each group. The complete set of endogenous scenario pairs is given by the union of these sets, as defined in Equation (4.16). To prove that this resulting set contains the minimum number of pairs, it is only necessary for us to show that the pairs in each group cannot be implied by any other pairs.

First, in the general case, we cannot guarantee that any uncertain parameters will be realized at the same time, since we have assumed that each parameter is associated with a different source. Second, we cannot guarantee that any of the parameters will be unrealized in certain time periods, either, since we have also assumed that there are no initial 'equality' periods. And third, we have assumed that there is no exogenous uncertainty, so there are no exogenous NACs. Thus, all of the endogenous NACs must be applied conditionally, and we cannot use any one to imply any of the others. This case can be seen clearly in Figure 1.5. It follows that, under these strict assumptions, the complete set of endogenous scenario pairs generated by Property 4, $\mathcal{SP}_{N^4}$, contains the minimum number of pairs. $\square$

B.8 Proof of Property 5

Consider exogenous scenario group $\hat{k}$ in time period $t = \tau$, where $\tau < T$, which corresponds to a set of scenarios given by ${}^{X}\mathcal{G}^{\hat{k}}_{\tau}$. These scenarios are adjacent, so we may express this set in the general form ${}^{X}\mathcal{G}^{\hat{k}}_{\tau} = \{s : s = n, n+1, \ldots, N\}$. It follows that the exogenous non-anticipativity constraints between the scenarios in group ${}^{X}\mathcal{G}^{\hat{k}}_{\tau}$ are:

$$y^{n}_{\tau} = y^{n+1}_{\tau}, \;\; y^{n+1}_{\tau} = y^{n+2}_{\tau}, \ldots, \;\; y^{N-1}_{\tau} = y^{N}_{\tau} \tag{B.4}$$

where, for simplicity, we consider only variables y^{s}_{τ}. Note that these scenarios are in the same subtree. Let ${}^{X}\mathcal{G}_{\tau}^{\tilde{k}}$ be the corresponding group of scenarios in a different subtree; i.e., all scenarios in the same position in ${}^{X}\mathcal{G}_{\tau}^{\hat{k}}$ and ${}^{X}\mathcal{G}_{\tau}^{\tilde{k}}$ are identical in the realizations of all exogenous parameters but differ in the possible realization of at least one endogenous parameter. Without loss of generality, we assume that the respective scenarios differ in the possible realization of exactly one endogenous parameter. We may express this set in the general form ${}^{X}\mathcal{G}_{\tau}^{\tilde{k}} = \{s : \ s = n^*, n^*+1, \ldots, N^*\}$, where $Pos(n) = Pos(n^*), \ Pos(n+1) = Pos(n^*+1), \ldots, Pos(N) = Pos(N^*)$. Exogenous NACs between the scenarios in group ${}^{X}\mathcal{G}_{\tau}^{\tilde{k}}$ can be written in the same form as for ${}^{X}\mathcal{G}_{\tau}^{\hat{k}}$:

$$y_{\tau}^{n^*} = y_{\tau}^{n^*+1}, \ \ y_{\tau}^{n^*+1} = y_{\tau}^{n^*+2}, \ldots, \ \ y_{\tau}^{N^*-1} = y_{\tau}^{N^*} \tag{B.5}$$

Endogenous non-anticipativity constraints apply only between scenarios s and s' in different subtrees, and by Property 3, it is sufficient to consider only s and s' that are identical in all exogenous realizations (i.e., $Pos(s) = Pos(s')$, where $s < s'$). Thus, the endogenous NACs between scenarios $s \in {}^{X}\mathcal{G}_{\tau}^{\hat{k}}$ and $s' \in {}^{X}\mathcal{G}_{\tau}^{\tilde{k}}$ are then:

$$y_{\tau}^{n} = y_{\tau}^{n^*}, \ \ y_{\tau}^{n+1} = y_{\tau}^{n^*+1}, \ldots, \ \ y_{\tau}^{N} = y_{\tau}^{N^*} \tag{B.6}$$

provided that the scenarios are indistinguishable (i.e., $\tau \in \mathcal{T}_{E}^{i'}$, or $\tau \in \mathcal{T}_{C}^{i'}$ and $Z_{\tau}^{s,s'} = True$, where $\{i'\} = \hat{\mathcal{D}}^{s,s'}$). Recall that s and s' differ in the possible realization of the same endogenous parameter, and only this one parameter, so the corresponding NACs between these scenarios are all active at the same time or are all ignored at the same time. Furthermore, if $Z_{\tau}^{s,s'} = False$, these non-anticipativity constraints do not apply, so it is only necessary to consider the case where these constraints are active.

Since $y_{\tau}^{n} = y_{\tau}^{n+1}$ by Equation (B.4), and $y_{\tau}^{n^*} = y_{\tau}^{n^*+1}$ by Equation (B.5), it follows that the first endogenous constraint in Equation (B.6), $y_{\tau}^{n} = y_{\tau}^{n^*}$, can be restated as $y_{\tau}^{n+1} = y_{\tau}^{n^*+1}$. Notice that this is the second endogenous constraint in Equation (B.6). This procedure can be continued to produce all of the remaining endogenous constraints in Equation (B.6). This shows that the exogenous NACs for two groups, along with *one* endogenous NAC linking one scenario from each group, imply all of the other endogenous NACs linking the two groups. Thus, only one endogenous NAC between the groups is sufficient. $\square$

B.9 Proof of Proposition 4

Starting from Proposition 3, notice that we have relaxed only one assumption; namely, that the problem is purely endogenous. To prove that we have the minimum number of endogenous scenario pairs, it is merely necessary for us to show that after the introduction of exogenous uncertainty, and the application of Property 5, the pairs in each endogenous scenario group cannot be implied by any other pairs.

First, recall that when both endogenous and exogenous uncertain parameters are present in the model, some of the endogenous scenario pairs can be implied by exogenous pairs. All such redundant pairs are eliminated by Property 5.

We may then rely on the remaining arguments in the proof of Proposition 3 to conclude that all endogenous NACs must be applied conditionally, and we cannot use any one to imply any of the others. It follows that, under the stated assumptions, the complete set of endogenous scenario pairs generated by Property 4 and Property 5, $\mathcal{SP}_{N^5}$, contains the minimum number of pairs. $\square$

B.10 Unique Scenarios Algorithm

For convenience in the algorithm, we partition the set of sources $\mathcal{I}$ into ordered sets $\mathcal{I}_E^t$ and $\mathcal{I}_C^t$ based on the given time period (not to be confused with set $\overline{\mathcal{I}}_t^s$, which was introduced for illustrative purposes in Chapter 1). Specifically, in the initial 'equality' time periods $t \in \mathcal{T}_E^{i'}$, the endogenous uncertainty cannot yet be resolved for sources $i' \in \mathcal{I}_E^t$, where $\mathcal{I}_E^t := \{i' : i' \in \mathcal{I},\ t \in \mathcal{T}_E^{i'}\}\ \forall\, t \in \mathcal{T}$. These sets of sources are associated with fixed endogenous NACs (since all uncertain parameters associated with these sources are guaranteed to be unresolved at time t, and thus all scenarios that differ only in the possible realizations of any of these parameters must be indistinguishable at that time). Note that we will always have $\mathcal{I}_E^T := \emptyset$, as the initial 'equality' periods should never span the entire time horizon. We do not restrict the definition of $\mathcal{I}_E^t$ to $t \in \mathcal{T} \setminus \{T\}$, however, as this would require us to treat $t < T$ and $t = T$ as two separate cases in the algorithm.

In the remaining 'conditional' time periods $t \in \mathcal{T}_C^{i'}$, the endogenous uncertainty may be resolved for sources $i' \in \mathcal{I}_C^t$, where $\mathcal{I}_C^t := \{i' : i' \in \mathcal{I},\ t \in \mathcal{T}_C^{i'}\}\ \forall\, t \in \mathcal{T}$. These sets of sources are associated with conditional endogenous NACs (since the uncertain parameters associated with these sources are no longer guaranteed to be unresolved at time t, and thus we can only say that the scenarios that differ in the possible realizations of any of these parameters *may* be indistinguishable at that time).

We now present the unique scenarios algorithm, in which we define sets $\mathcal{U}_t^{i',h}$ corresponding to each endogenous parameter $\theta_{i',h}$, for all $i' \in \mathcal{I}$ and $h \in \mathcal{H}_{i'}$, in each time period $t \in \mathcal{T}$.

Unique Scenarios Algorithm

Step 1 For each time period $t \in \mathcal{T}$:

Step 1a First, consider the sources that are associated with fixed endogenous NACs in this time period. In other words, for each source $i' \in \mathcal{I}_E^t$ (where the sources are considered in ascending numerical order):

(i) If i' is the first source in this set (i.e., $i' = \min_{\hat{i}'}(\hat{i}' \in \mathcal{I}_E^t)$), initialize the set of unique scenarios to that obtained from Property 5 (e.g., Equation (4.17)) in order to take advantage of the reductions associated with exogenous scenario grouping:

$$\mathcal{U}_t^{i',1} := \tilde{\mathcal{U}}_t \qquad i' = \min_{\hat{i}'}\Big(\hat{i}' \in \mathcal{I}_E^t\Big) \tag{B.7}$$

(ii) If there is more than one endogenous parameter associated with source i', define the corresponding set of unique scenarios that can be considered for each of these parameters, as indicated in Equation (B.8):

$$\mathcal{U}_t^{i',h+1} := \mathcal{U}_t^{i',h} \cap \left[\bigcup_{l \in \mathcal{L}_{i',h}} \left\{\min_{\hat{s}}\left(\hat{s} \in {}^N\mathcal{G}_{i',h}^l\right)\right\}\right] \qquad h = 1, \ldots, H_{i'} - 1 \tag{B.8}$$

(iii) If $i' < \max_{\hat{i}'}(\hat{i}' \in \mathcal{I}_E^t)$, there is at least one additional source to consider. Accordingly, define the set of unique scenarios for the first endogenous parameter of the *next* source, i'', as indicated in Equation (B.9):

$$\mathcal{U}_t^{i'',1} := \mathcal{U}_t^{i',H_{i'}} \cap \left[\bigcup_{l \in \mathcal{L}_{i',H_{i'}}} \left\{ \min_{\hat{s}} \left(\hat{s} \in {}^N\mathcal{G}^l_{i',H_{i'}} \right) \right\} \right] \qquad i'' = \min_{\hat{\imath}''} \Big(\hat{\imath}'' \in \mathcal{I}_E^t, \;\; \hat{\imath}'' > i' \Big) \tag{B.9}$$

(iv) If $i' = \max_{\hat{\imath}'}(\hat{\imath}' \in \mathcal{I}_E^t)$, this is the last source. Store the current set of unique scenarios in a separate, temporary set, UniqueSet, but in the same manner as Equation (B.9):

$$\text{UniqueSet} := \mathcal{U}_t^{i',H_{i'}} \cap \left[\bigcup_{l \in \mathcal{L}_{i',H_{i'}}} \left\{ \min_{\hat{s}} \left(\hat{s} \in {}^N\mathcal{G}^l_{i',H_{i'}} \right) \right\} \right] \tag{B.10}$$

Step 1b Next, consider the sources that are associated with conditional endogenous NACs in this time period. In other words, for each source $i' \in \mathcal{I}_C^t$ (where the sources are considered in ascending numerical order):

(i) If i' is the first source in this set (i.e., $i' = \min_{\hat{\imath}'}(\hat{\imath}' \in \mathcal{I}_C^t)$), initialize the set of unique scenarios based on the following two conditions:

(1) If $\mathcal{I}_E^t \neq \emptyset$, then t is an initial 'equality' time period for at least one of the sources. Accordingly, initialize the set of unique scenarios to the last-known value, stored in Equation (B.10), in order to take advantage of the reductions associated with the fixed endogenous NACs:

$$\mathcal{U}_t^{i',1} := \text{UniqueSet} \qquad i' = \min_{\hat{\imath}'} \Big(\hat{\imath}' \in \mathcal{I}_C^t \Big) \tag{B.11}$$

(2) If, however, $\mathcal{I}_E^t = \emptyset$, then there are no fixed endogenous NACs in time period t. Similar to sub-step (i) of Step 1a, initialize the set of unique scenarios to that obtained from Property 5:

$$\mathcal{U}_t^{i',1} := \tilde{\mathcal{U}}_t \qquad i' = \min_{\hat{\imath}'} \Big(\hat{\imath}' \in \mathcal{I}_C^t \Big) \tag{B.12}$$

In the case where $t = T$, recall that $\tilde{\mathcal{U}}_T := \mathcal{S}$.

(ii) Execute sub-step (ii) of Step 1a.

(iii) If $i' < \max_{\hat{\imath}'}(\hat{\imath}' \in \mathcal{I}_C^t)$, define the set of unique scenarios for the first endogenous parameter of the next source, i'', as indicated in Equation (B.13):

$$\mathcal{U}_t^{i'',1} := \mathcal{U}_t^{i',1} \qquad i'' = \min_{\hat{\imath}''} \Big(\hat{\imath}'' \in \mathcal{I}_C^t, \;\; \hat{\imath}'' > i' \Big) \tag{B.13}$$

Notice that Step 1a will automatically be skipped for $t = T$, since $\mathcal{I}_E^T = \emptyset$.

Because this algorithm is fairly complex, we next discuss some of the respective expressions in further detail.

Equation (B.8) addresses the case in which we have multiple endogenous parameters associated with some of the sources of uncertainty. Therefore, this expression updates the set of unique scenarios in time period t when advancing from one endogenous parameter h to the next, $h + 1$, for a given source i'. The assumption here is that *all* endogenous parameters associated with the same source must be realized at the same time. This is because an investment *in the source itself* determines the time at which the associated technical information can be realized. For example, once we drill an oilfield and begin producing from it, we assume that we can determine both the size and initial deliverability of the reserves.

Accordingly, for any scenarios s and s' that differ only in the possible realization of an endogenous parameter associated with source i', the corresponding non-anticipativity constraints will all apply at the same time or will all be ignored at the same time. This was previously shown in the discussion surrounding Figure 4.8. Because it is then sufficient to consider only the case where these constraints are active, when we proceed from the first endogenous parameter of source i' to the second (i.e., $h = 1$ to $h = 2$), we begin to pair off scenarios between the groups of $\theta_{i',1}$ (we prove this point in Appendix B.11), and the scenarios in each of those respective groups must be indistinguishable. It is therefore unnecessary to have more than one link between any such groups. The reasoning here is justified in Appendix B.11.

Thus, in time period t, when advancing from endogenous parameter $\theta_{i',h}$ to the next parameter of the same source, $\theta_{i',h+1}$, we require *at most* only a single 'representative' scenario from each endogenous scenario group corresponding to $\theta_{i',h}$; i.e., $\bigcup_{l \in \mathcal{L}_{i',h}} \left\{ \min_{\hat{s}} \left(\hat{s} \in {}^{N}\mathcal{G}^{l}_{i',h} \right) \right\}$. Notice the similarity of this case to our treatment of the exogenous scenario groups in Property 5 (see Equation (4.17)). We say "*at most*" since some of these representative scenarios may be non-unique based on Property 5 and/or the consideration of other endogenous parameters *before this point* in the unique scenarios algorithm. We successively remove non-unique scenarios in time period t by taking the intersection of the current set of unique scenarios, $\mathcal{U}_t^{i',h}$, and the set of representative scenarios, $\bigcup_{l \in \mathcal{L}_{i',h}} \left\{ \min_{\hat{s}} \left(\hat{s} \in {}^{N}\mathcal{G}^{l}_{i',h} \right) \right\}$. The result, as shown in Equation (B.8), is an *updated* set of unique scenarios that can be considered for the next parameter in the algorithm (i.e., $\theta_{i',h+1}$). It is important to note that this case is evaluated in the same way in all time periods and appears in sub-step (ii) of Step 1a and Step 1b of the algorithm.

Equation (B.9) addresses the case in which there are endogenous parameters that cannot be realized in some of the initial time periods. Rather than defining an updated set of unique scenarios in time period t for each endogenous parameter of the *same* source i', as in Equation (B.8), this expression considers the case of advancing from the last endogenous parameter, $H_{i'}$, of source i', to the first endogenous parameter of the next source, i''. The reasoning here is that if the uncertainty in source i' cannot yet be revealed as of time period t (i.e., $i' \in \mathcal{I}_E^t$), then we will have equality constraints corresponding to all scenario pairs (s, s') for which s and s' differ in the possible realization of a parameter associated with i'. This implies that there will then be redundant constraints associated with the parameters of the *next* source, i'', as illustrated in Figure 4.7. We can thus perform further reduction via the same strategy used in Equation (B.8) for the case of multiple parameters associated with the same source. Notice, in particular, that the form of Equation (B.9) is nearly identical to Equation (B.8).

One subtle difference here is that the sources in set $\mathcal{I}_E^t$ may be nonconsecutively indexed, which means that we cannot use index $i' + 1$ to access the next element in the set (as we do with $h+1$ to access the next parameter in Equation (B.8)). Instead, we use a strategy similar to that previously introduced in Equation (4.15) and state $i'' = \min_{\hat{\imath}''}(\hat{\imath}'' \in \mathcal{I}_E^t,\ \hat{\imath}'' > i')$. This expression simply allows us to advance from one source, i', to the next-lowest-indexed source, i'', in an ordered manner.

Equation (B.10) is a special case of Equation (B.9) that is evaluated only for the last source in set $\mathcal{I}_E^t$. The corresponding set of unique scenarios is stored in a temporary set, UniqueSet, which may be used for initialization in sub-step (i) of Step 1b.

In Step 1b, notice that time period t is not an initial 'equality' period for source i'. Here we consider sources $i' \in \mathcal{I}_C^t$, and we can no longer guarantee that we will have equality constraints corresponding to the scenario pairs (s, s') for which s and s' differ in the possible realization of a parameter associated with i'. It follows that since all of these constraints are conditional, the only reduction that we can perform is for multiple parameters associated with the same source i' (via sub-step (ii)); we cannot make any further assumptions to eliminate constraints corresponding to the next source, i''. We thus use Equation (B.13) in which the definition of the set of unique scenarios is unchanged from one source to the next. We state $\mathcal{U}_t^{i'',1} := \mathcal{U}_t^{i',1}$ in this equation, rather than $\mathcal{U}_t^{i'',1} := \mathcal{U}_t^{i',H_{i'}}$, because the reduction from sub-step (ii) cannot be carried over to the next source as it does in Step 1a. Also, note that the sources in set $\mathcal{I}_C^t$ may be nonconsecutively indexed, so like the treatment of set $\mathcal{I}_E^t$ in Equation (B.9), we use $i'' = \min_{\hat{\imath}''}(\hat{\imath}'' \in \mathcal{I}_C^t,\ \hat{\imath}'' > i')$ to access the next-lowest-indexed source.

As a brief example of how this algorithm is applied, consider Figure 4.7 and assume that only these 4 scenarios are under consideration. We start at $t = 1$. Since there is a single endogenous parameter associated with each of the two sources, we will drop the h index to simplify the notation.

It is clear that we are starting with unique scenarios $\tilde{\mathcal{U}}_1 = \{1, 5, 9, 13\}$ from Property 5. It is also clear that we have endogenous scenario groups ${}^N\mathcal{G}_1^1 = \{1, 9\}$ and ${}^N\mathcal{G}_1^2 = \{5, 13\}$ corresponding to θ_1, and ${}^N\mathcal{G}_2^1 = \{1, 5\}$ and ${}^N\mathcal{G}_2^2 = \{9, 13\}$ corresponding to θ_2. Notice that $t = 1$ is an initial 'equality' period only for θ_2 (i.e., $\mathcal{T}_E^1 = \emptyset$ and $\mathcal{T}_E^2 = \{1\}$), so $\mathcal{I}_E^1 = \{2\}$ and $\mathcal{I}_C^1 = \{1\}$.

We start with Step 1a for the sources associated with fixed endogenous NACs in the first time period and must consider $i' \in \{2\}$. We first initialize the corresponding set of unique scenarios (i.e., $\mathcal{U}_1^2$) in sub-step (i), as indicated in Equation (B.7): $\mathcal{U}_1^2 := \tilde{\mathcal{U}}_1 = \{1, 5, 9, 13\}$.

Notice that because there is only one endogenous parameter associated with source 2, we skip sub-step (ii). Since $i' = \max_{\hat{\imath}'}(\hat{\imath}' \in \{2\}) = 2$, we also skip sub-step (iii) and proceed to (iv). This yields the following, by Equation (B.10):

$$
\begin{aligned}
\text{UniqueSet} := \mathcal{U}_1^2 \cap &\left[\bigcup_{l \in \mathcal{L}_2} \left\{ \min_{\hat{s}} \left(\hat{s} \in {}^N\mathcal{G}_2^l \right) \right\} \right] \\
&= \{1, 5, 9, 13\} \cap \left[\left\{ \min_{\hat{s}} \left(\hat{s} \in \{1, 5\} \right) \right\} \cup \left\{ \min_{\hat{s}} \left(\hat{s} \in \{9, 13\} \right) \right\} \right]
\end{aligned}
$$

which simplifies to UniqueSet $:= \{1, 5, 9, 13\} \cap \{1, 9\} = \{1, 9\}$.

We then continue to Step 1b for the sources associated with conditional endogenous NACs in the first time period. Here, we must consider $i' \in \{1\}$. We first initialize the set of unique scenarios to the last-known value, given by the temporary set UniqueSet, in sub-step (i), condition (1). Specifically, by Equation (B.11), $\mathcal{U}_1^1 := \{1, 9\}$.

Because there is only one endogenous parameter associated with source 1, we skip sub-step (ii). We also skip sub-step (iii) since $i' = \max_{\hat{\imath}'}(\hat{\imath}' \in \{1\}) = 1$. In practice, we would then continue to $t = 2$. To summarize, $\mathcal{U}_1^1 := \{1, 9\}$ and $\mathcal{U}_1^2 := \{1, 5, 9, 13\}$.

Equation (3.22) requires us to form pairs among consecutive scenarios in sets ${}^N\mathcal{G}_1^1 \cap \mathcal{U}_1^1$, ${}^N\mathcal{G}_1^2 \cap \mathcal{U}_1^1$, ${}^N\mathcal{G}_2^l \cap \mathcal{U}_1^2$, and ${}^N\mathcal{G}_2^2 \cap \mathcal{U}_1^2$ for $t = 1$. These sets are given by $\{1, 9\} \cap \{1, 9\}$, $\{5, 13\} \cap \{1, 9\}$, $\{1, 5\} \cap \{1, 5, 9, 13\}$, and $\{9, 13\} \cap \{1, 5, 9, 13\}$, respectively, which reduce to $\{1, 9\}$, $\emptyset$, $\{1, 5\}$, and $\{9, 13\}$, respectively. Pairing off consecutive scenarios in these sets yields pairs $(1, 9)$, $(1, 5)$, and $(9, 13)$, as shown in Figure 4.7. Notice that none of the remaining pairs can be implied by any of the others.

It is worth noting that repeating this procedure for a case such as Figure 4.8 will lead to slightly different scenario pairs than pictured. This is due to the order in which we consider the endogenous parameters. Specifically, by considering pairs among scenarios that differ in the possible realization of $\theta_{\hat{\imath},2}$ first, and then those for $\theta_{\hat{\imath},1}$, we obtain scenario pairs $(\hat{s}, \hat{s}')$, $(\hat{s}, \hat{s}'')$, and $(\hat{s}'', \hat{s}''')$ in Figure 4.8. By the unique scenarios algorithm, however, we consider the parameters in numerical order (i.e., $\theta_{\hat{\imath},1}$, followed by $\theta_{\hat{\imath},2}$) and instead obtain $(\hat{s}, \hat{s}')$, $(\hat{s}, \hat{s}'')$, and $(\hat{s}', \hat{s}''')$. Although the first set may appear to be more "natural" based on the appearance of Figure 4.8, both sets are equally valid since only 3 pairs are required to link the 4 scenarios.

B.11 Proof of Property 6

Consider the endogenous scenario groups $l = 1, 2, \ldots, |\mathcal{L}_{\hat{\imath},\hat{h}}|$ corresponding to endogenous parameter $\theta_{\hat{\imath},\hat{h}}$. The scenarios in each of these respective group differ *only* in the possible realization of $\theta_{\hat{\imath},\hat{h}}$. Given that there are $M_{\hat{\imath},\hat{h}}$ (or $|\Theta_{\hat{\imath},\hat{h}}|$) possible realizations for $\theta_{\hat{\imath},\hat{h}}$, and in each respective group, each scenario must have a different possible realization for this endogenous parameter, it follows that there can only be $M_{\hat{\imath},\hat{h}}$ scenarios in each of these groups (i.e., one for each possible realization of $\theta_{\hat{\imath},\hat{h}}$).

Furthermore, the lowest-indexed scenario in each of these groups must have the lowest realization for $\theta_{\hat{\imath},\hat{h}}$. This is simply a consequence of the ordering on the set of realizations $\Theta_{\hat{\imath},\hat{h}}$ (i.e., $\hat{\theta}^1_{\hat{\imath},\hat{h}} < \hat{\theta}^2_{\hat{\imath},\hat{h}} < \cdots < \hat{\theta}^{M_{\hat{\imath},\hat{h}}}_{\hat{\imath},\hat{h}}$) and the lexicographical ordering on the Cartesian products used in the scenario-generation process (see Chapter 2). Specifically, as can be seen in Equation (2.5) and even more clearly in Figure 2.1c, we must exhaust all possible combinations of realizations for the uncertain parameters that occur *after* $\theta_{\hat{\imath},\hat{h}}$ in the Cartesian product before the realization of $\theta_{\hat{\imath},\hat{h}}$ can be incremented to the next possible value. This means that a scenario $\hat{s}$ defined with a low realization for $\theta_{\hat{\imath},\hat{h}}$ will come before a scenario $\hat{s}'$ with a high realization for $\theta_{\hat{\imath},\hat{h}}$ and all of the same possible realizations for the other uncertain parameters. Accordingly, in an endogenous scenario group, it follows that the lowest-indexed scenario must have the lowest realization for $\theta_{\hat{\imath},\hat{h}}$, the next-lowest-indexed scenario must have the next-lowest realization for $\theta_{\hat{\imath},\hat{h}}$, and so forth, until we reach the highest-indexed scenario in the group, which must have the highest realization for $\theta_{\hat{\imath},\hat{h}}$. For example, if $\theta_{\hat{\imath},\hat{h}}$ were defined with 3 possible realizations (*low* (L), *medium* (M), or *high* (H)), there would be 3 scenarios in each of the corresponding (ordered) endogenous scenario groups, with realizations of the following form: $(\ldots, \hat{\theta}^L_{\hat{\imath},\hat{h}}, \ldots), (\ldots, \hat{\theta}^M_{\hat{\imath},\hat{h}}, \ldots), (\ldots, \hat{\theta}^H_{\hat{\imath},\hat{h}}, \ldots)$. Note that this is the case depicted in Figure 4.4 for Property 4.

Consider two scenarios $\hat{s}$ and $\hat{s}'$ from one arbitrary group $\hat{l}$ corresponding to $\theta_{\hat{\imath},\hat{h}}$ (i.e., ${}^N\mathcal{G}^{\hat{l}}_{\hat{\imath},\hat{h}}$).

Recall that this means that $\hat{s}$ and $\hat{s}'$ must have the same possible realizations for all uncertain parameters except $\theta_{\hat{\imath},\hat{h}}$. Because *all* scenarios must be placed in an endogenous scenario group corresponding to each endogenous parameter (by the endogenous scenario-group algorithm), both of these scenarios will also be placed into groups for a different endogenous parameter, $\theta_{\tilde{\imath},\tilde{h}}$. The two scenarios cannot be placed in the same endogenous scenario group in this case, however, since they have different possible realizations for $\theta_{\hat{\imath},\hat{h}}$ and thus would differ in the possible realizations of both $\theta_{\hat{\imath},\hat{h}}$ *and* $\theta_{\tilde{\imath},\tilde{h}}$ (i.e., 2 parameters). This would violate Property 3. It follows, then, that two scenarios in the same endogenous scenario group cannot appear together in *any* other endogenous scenario group, for *any* endogenous parameter. This is a fairly obvious conclusion since $\hat{s}$ and $\hat{s}'$ differ in the possible realization of only one endogenous parameter, and in any arbitrary time period, we would expect scenario pair $(\hat{s}, \hat{s}')$ to appear only once.

The endogenous scenario groups corresponding to $\theta_{\hat{\imath},\hat{h}}$ will have the following form: ${}^{N}\mathcal{G}^{\hat{l}}_{\hat{\imath},\hat{h}} := \{s : s = \alpha_1, \alpha_2, \ldots, \alpha_{M_{\hat{\imath},\hat{h}}}\}$, ${}^{N}\mathcal{G}^{\hat{l}'}_{\hat{\imath},\hat{h}} := \{s : s = \beta_1, \beta_2, \ldots, \beta_{M_{\hat{\imath},\hat{h}}}\}$, ${}^{N}\mathcal{G}^{\hat{l}''}_{\hat{\imath},\hat{h}} := \{s : s = \eta_1, \eta_2, \ldots, \eta_{M_{\hat{\imath},\hat{h}}}\}$, etc. By Property 4, we pair off consecutive scenarios in each of these groups. Note that these scenarios may be nonconsecutively indexed, and this necessitates the use of a different naming convention than used previously in the proof of Property 5. The associated endogenous NACs for an arbitrary time period $t = \tau$ are then:

$$y_\tau^{\alpha_1} = y_\tau^{\alpha_2}, \ldots, \; y_\tau^{\alpha_{M_{\hat{\imath},\hat{h}}-1}} = y_\tau^{\alpha_{M_{\hat{\imath},\hat{h}}}} \tag{B.14}$$

$$y_\tau^{\beta_1} = y_\tau^{\beta_2}, \ldots, \; y_\tau^{\beta_{M_{\hat{\imath},\hat{h}}-1}} = y_\tau^{\beta_{M_{\hat{\imath},\hat{h}}}} \tag{B.15}$$

$$y_\tau^{\eta_1} = y_\tau^{\eta_2}, \ldots, \; y_\tau^{\eta_{M_{\hat{\imath},\hat{h}}-1}} = y_\tau^{\eta_{M_{\hat{\imath},\hat{h}}}} \tag{B.16}$$

provided that $\theta_{\hat{\imath},\hat{h}}$ has not yet been realized (i.e., the scenarios are indistinguishable). If $\theta_{\hat{\imath},\hat{h}}$ *has* been realized, then the scenarios are distinguishable and the NACs do not apply, so it is only necessary for us to consider the former case where these constraints are active.

At this point, recall that for any endogenous parameter, every scenario in $\mathcal{S}$ can be accounted for as a member of one of the endogenous scenario groups corresponding to that parameter. Further recall that none of the scenarios in those respective groups can appear together in any other group. This means that *all* other endogenous scenario groups can be produced, respectively, by selecting one scenario from different groups defined for $\theta_{\hat{\imath},\hat{h}}$. Accordingly, we will continue to use the same naming convention in the definitions of other endogenous scenario groups in this proof (i.e., we will use α_m, β_m, η_m, etc. for scenarios, where $m = 1, 2, \ldots, M_{\hat{\imath},\hat{h}}$).

Now, consider two scenarios $\hat{s}$ and $\hat{s}''$ from two separate groups $\hat{l}$ and $\hat{l}'$ corresponding to $\theta_{\hat{\imath},\hat{h}}$ (i.e., ${}^{N}\mathcal{G}^{\hat{l}}_{\hat{\imath},\hat{h}}$ and ${}^{N}\mathcal{G}^{\hat{l}'}_{\hat{\imath},\hat{h}}$). In this case, the scenarios *may* differ in the possible realization of $\theta_{\hat{\imath},\hat{h}}$ (depending on their respective positions in the two groups), and *must* differ in the possible realization of at least one endogenous parameter other than $\theta_{\hat{\imath},\hat{h}}$ (since they belong to two separate groups corresponding to $\theta_{\hat{\imath},\hat{h}}$). Notice that the only way for scenarios $\hat{s}$ and $\hat{s}''$ to have the same possible realization for $\theta_{\hat{\imath},\hat{h}}$ is if they have the same *position* in both groups (not to be confused with parameter $Pos(s)$). For example, if they both have the lowest realization for $\theta_{\hat{\imath},\hat{h}}$, they must be the lowest-indexed scenarios in their respective groups; if they both have the highest realization for $\theta_{\hat{\imath},\hat{h}}$, they must be the highest-indexed scenarios in their respective groups. If, then, $\hat{s}$ and $\hat{s}''$ have the same possible realization for $\theta_{\hat{\imath},\hat{h}}$ (i.e., the same position in both of their groups) and differ only in the possible realization of endogenous parameter $\theta_{\tilde{\imath},\tilde{h}}$, they will be placed in the same group corresponding to $\theta_{\tilde{\imath},\tilde{h}}$ by the endogenous scenario-group algorithm. (Note that if $\hat{s}$ and $\hat{s}''$ instead differ in other possible parameter realizations, they will be placed in different groups, and the discussion that follows would apply for the specific endogenous parameter for which this condition does apply.)

The endogenous scenario groups corresponding to $\theta_{\tilde{\imath},\tilde{h}}$ will then have the following form: ${}^{N}\mathcal{G}^{\tilde{l}}_{\tilde{\imath},\tilde{h}} := \{s:\ s=\alpha_1,\beta_1,\eta_1,\dots\}$, ${}^{N}\mathcal{G}^{\tilde{l}'}_{\tilde{\imath},\tilde{h}} := \{s:\ s=\alpha_2,\beta_2,\eta_2,\dots\}$, $\dots$, ${}^{N}\mathcal{G}^{\tilde{l}''}_{\tilde{\imath},\tilde{h}} := \{s:\ s=\alpha_{M_{\hat{\imath},\hat{h}}},\beta_{M_{\hat{\imath},\hat{h}}},\eta_{M_{\hat{\imath},\hat{h}}},\dots\}$, etc.

Notice that in ${}^{N}\mathcal{G}^{\tilde{l}}_{\tilde{\imath},\tilde{h}}$, the lowest-indexed scenario from group ${}^{N}\mathcal{G}^{\hat{l}}_{\hat{\imath},\hat{h}}$ has been grouped with the lowest-indexed scenarios from groups ${}^{N}\mathcal{G}^{\hat{l}'}_{\hat{\imath},\hat{h}}$ and ${}^{N}\mathcal{G}^{\hat{l}''}_{\hat{\imath},\hat{h}}$, the second-lowest-indexed scenarios have been grouped in ${}^{N}\mathcal{G}^{\tilde{l}'}_{\tilde{\imath},\tilde{h}}$, and so forth. In general, the remaining groups corresponding to $\theta_{\tilde{\imath},\tilde{h}}$ would be generated from all other groups corresponding to $\theta_{\hat{\imath},\hat{h}}$, in the same manner, and would consist of scenarios other than α_m, β_m, and η_m. Note that the same general approach also applies for the endogenous scenario groups of all other endogenous parameters, with the respective scenarios selected from different groups corresponding to $\theta_{\hat{\imath},\hat{h}}$. As before, the associated NACs for time period $t=\tau$ are:

$$y_\tau^{\alpha_1} = y_\tau^{\beta_1},\ \ y_\tau^{\beta_1} = y_\tau^{\eta_1},\dots \tag{B.17}$$
$$y_\tau^{\alpha_2} = y_\tau^{\beta_2},\ \ y_\tau^{\beta_2} = y_\tau^{\eta_2},\dots \tag{B.18}$$
$$y_\tau^{\alpha_{M_{\hat{\imath},\hat{h}}}} = y_\tau^{\beta_{M_{\hat{\imath},\hat{h}}}},\ \ y_\tau^{\beta_{M_{\hat{\imath},\hat{h}}}} = y_\tau^{\eta_{M_{\hat{\imath},\hat{h}}}},\dots \tag{B.19}$$

provided that $\theta_{\tilde{\imath},\tilde{h}}$ has not yet been realized.

Notice that because $y_\tau^{\alpha_1} = y_\tau^{\alpha_2}$ by Equation (B.14), and $y_\tau^{\beta_1} = y_\tau^{\beta_2}$ by Equation (B.15), we can rewrite the *first* endogenous constraint in Equation (B.17), $y_\tau^{\alpha_1} = y_\tau^{\beta_1}$, as $y_\tau^{\alpha_2} = y_\tau^{\beta_2}$. Notice that this is the *first* endogenous constraint in Equation (B.18).

Since $y_\tau^{\eta_1} = y_\tau^{\eta_2}$ by Equation (B.16), we can use this constraint with Equation (B.15) to rewrite the *second* endogenous constraint in Equation (B.17), $y_\tau^{\beta_1} = y_\tau^{\eta_1}$, as $y_\tau^{\beta_2} = y_\tau^{\eta_2}$. Notice that this is the *second* endogenous constraint in Equation (B.18).

Given any remaining scenarios that differ from α_2, β_2, and η_2 in the possible realization of only $\theta_{\tilde{\imath},\tilde{h}}$, this process can be continued to produce all of the remaining NACs corresponding to group ${}^{N}\mathcal{G}^{\tilde{l}'}_{\tilde{\imath},\tilde{h}}$ in Equation (B.18). In fact, if we consider only scenarios α_m, β_m, and η_m (where $m=1,2,\dots,M_{\hat{\imath},\hat{h}}$), it is not difficult to see that by using the first, second, third, etc. NACs from Equations (B.14)–(B.16), along with Equation (B.17), we can imply *all* of the NACs corresponding to the groups for $\theta_{\tilde{\imath},\tilde{h}}$ that involve these scenarios. This includes the NACs for ${}^{N}\mathcal{G}^{\tilde{l}''}_{\tilde{\imath},\tilde{h}}$, as shown in Equation (B.19).

To summarize the results in a general sense, first recall that we begin with the endogenous scenario groups corresponding to an arbitrary parameter $\theta_{\hat{\imath},\hat{h}}$. We will refer to these groups as our "base" groups. We assume that the corresponding NACs apply as equality constraints (i.e., fixed endogenous NACs).

Further recall that the groups for all other endogenous parameters can be produced by selecting scenarios from different base groups, where in each case, the scenarios have *the same position* in their respective base groups (e.g., the lowest indexed, the second-lowest indexed, etc.). However, for an arbitrary parameter $\theta_{\tilde{\imath},\tilde{h}}$, we have just shown that the groups produced from the *second-lowest-indexed* scenarios, the *third-lowest-indexed* scenarios, etc. result in redundant NACs. This is the case regardless of whether these constraints are conditional or fixed.

The reasoning here, in general, is that *all* NACs associated with an arbitrary endogenous parameter $\theta_{\tilde{\imath},\tilde{h}}$ can be implied by the base-group NACs and the NACs derived from pairing off the *lowest-indexed* scenarios from those base groups. (Note that our choice to use the lowest-indexed scenarios (rather than, for example, the highest) is arbitrary, and we have made this selection for convenience.)

It then follows that, for generating endogenous scenario pairs, it is sufficient to consider only scenarios s and s' that are in the *first position* of their respective base groups (i.e., the *lowest-indexed* scenarios from these groups), excluding all scenarios eliminated by Property 5. We refer to these scenarios as "unique." We may then proceed to another endogenous parameter for which the associated NACs apply as equality constraints, and consider new base groups, where we allow only the *new* lowest-indexed scenarios that were members of the previous set of unique scenarios. Because the introduction of new equality constraints may allow us to imply other existing endogenous constraints, we may be able to remove additional scenarios from the pairing process each time we update our set of unique scenarios. In time periods where we do not have fixed endogenous NACs, a similar strategy can still be used for the case of multiple parameters associated with the same source, although the set of unique scenarios can only be updated in the context of that particular source.

The strategy outlined here is the basis for the unique scenarios algorithm. □

B.12 Proof of Proposition 5

Starting from Proposition 4, notice that we have relaxed the two remaining assumptions; specifically, that there are no initial 'equality' periods and that there is only one endogenous parameter associated with each source of uncertainty. To prove that we have the minimum number of endogenous scenario pairs in this general case, it is only necessary for us to show that after relaxing the two assumptions, and then introducing Property 6, the pairs in each endogenous scenario group cannot be implied by any other pairs.

As previously discussed, there will be redundant scenario pairs in the model when we consider initial 'equality' periods and multiple endogenous parameters associated with some of the sources of uncertainty. These redundant pairs are eliminated by Property 6.

It follows that although there are both fixed endogenous NACs and conditional endogenous NACs, none of the remaining pairs can be used to imply any of the others. Thus, the complete set of endogenous scenario pairs, $\mathcal{SP}_N$, generated by Property 4, Property 5, and Property 6 contains the minimum number of pairs. □

B.13 Alternative Approach to Property 4

An alternative approach to Property 4 for the case where we have no initial 'equality' time periods and only one endogenous parameter associated with each source of uncertainty can be stated as follows:

For endogenous NACs, it is sufficient to consider only scenario pairs (s, s') for which s and s' are separated by a particular distance, $Dist(i', h)$, defined as:

$$Dist(i', h) := S_X \left(\prod_{\substack{h' \in \mathcal{H}_{i'} \\ h' > h}} M_{i',h'} \cdot \prod_{\substack{i'' \in \mathcal{I} \\ i'' > i'}} \prod_{h' \in \mathcal{H}_{i''}} M_{i'',h'} \right) \qquad \forall\, i' \in \mathcal{I},\;\; h \in \mathcal{H}_{i'} \tag{B.20}$$

where the indices i' and h are given by $\{(i', h)\} = \mathcal{D}^{s,s'}$ and indicate the specific endogenous parameter $\theta_{i',h}$ for which scenarios s and s' differ in possible realizations. The corresponding set of endogenous scenario pairs would then be defined as:

$$\Big\{(t, s, s') : \, t \in \mathcal{T},\;\; s, s' \in \mathcal{S},\;\; s < s',\;\; Pos(s) = Pos(s'),\;\; s' - s = Dist(i', h),\;\; \{(i', h)\} = \mathcal{D}^{s,s'}\Big\}$$

The general idea behind Equation (B.20) is that we can use a parameter's position in the scenario-generation Cartesian product to calculate how 'often' its realization will change (e.g., every 4 scenarios). This *distance* between scenarios is a direct result of our lexicographical ordering on the Cartesian product, which enforces that we must exhaust all possible combinations of realizations for the uncertain parameters that occur *after* $\theta_{i',h}$ in the Cartesian product before we can move on to the next possible realization for this parameter (see the beginning of Appendix B.11 for slightly more details). Accordingly, Equation (B.20) is expressed as the product of the number of possible realizations for all parameters that occur after $\theta_{i',h}$ in the Cartesian product.

This approach is inspired by an unpublished result from Gupta and Grossmann (2011) for the specific case of purely-endogenous multistage stochastic programs with no initial 'equality' time periods and only one endogenous parameter associated with each source of uncertainty. A similar conclusion regarding a "distance" between scenarios was also recently stated by Boland et al. (2016) for this class of problems. Note that for this specific case, the alternative approach — like Property 4 — provides the minimum number of endogenous scenario pairs. In fact, the resulting pairs are equivalent to those obtained by pairing off consecutive scenarios in each endogenous scenario group, and the use of these groups is entirely avoided. We cannot rely on this approach in general, however, as it does not extend well to the general case where these assumptions may not hold.

It is also worth noting that other methods exist for generating the minimum number of endogenous NACs in purely-endogenous MSSP problems with no initial 'equality' periods and only one endogenous parameter associated with each source. Specifically, the idea of using *arbitrary* scenario sets (rather than those generated by Cartesian products) was proposed by Boland et al. (2008) and has been the focus of two recent works, Boland et al. (2016) and Hooshmand Khaligh and MirHassani (2016b), as discussed in Chapter 6.

Appendix C Supplementary Material for Chapter 5

C.1 Reformulation of Penalty Terms in Lagrangean Relaxation

In its original form, model (MSSPS-LR) does not correspond to independent scenario subproblems since the objective function, Equation (5.11), contains penalty terms that involve both y_t^s and $y_t^{s'}$. For example:

$$\begin{aligned}\sum_{(s,s')\in\mathcal{SP}_F} {}^F\lambda_1^{s,s'}(y_1^s - y_1^{s'}) &= \sum_{(s,s')\in\mathcal{SP}_F}\left({}^F\lambda_1^{s,s'}y_1^s - {}^F\lambda_1^{s,s'}y_1^{s'}\right)\\ &= \sum_{(s,s')\in\mathcal{SP}_F} {}^F\lambda_1^{s,s'}y_1^s - \sum_{(s,s')\in\mathcal{SP}_F} {}^F\lambda_1^{s,s'}y_1^{s'}\end{aligned} \tag{C.1}$$

where, for illustrative purposes, we consider only the penalty terms associated with the first-period NACs. Because we have $s \in \mathcal{S}$ and $s' \in \mathcal{S}$, we can swap indices s and s' in any of the previous summations without changing the meaning of the respective summation. We use this simple observation to rewrite the last summation as follows:

$$\sum_{(s,s')\in\mathcal{SP}_F} {}^F\lambda_1^{s,s'}y_1^{s'} = \sum_{(s',s)\in\mathcal{SP}_F} {}^F\lambda_1^{s',s}y_1^{s} \tag{C.2}$$

Making this substitution, we have:

$$\begin{aligned}\sum_{(s,s')\in\mathcal{SP}_F} {}^F\lambda_1^{s,s'}(y_1^s - y_1^{s'}) &= \sum_{(s,s')\in\mathcal{SP}_F} {}^F\lambda_1^{s,s'}y_1^s - \sum_{(s',s)\in\mathcal{SP}_F} {}^F\lambda_1^{s',s}y_1^{s}\\ &= \sum_{s\in\mathcal{S}} y_1^s\left(\sum_{(s,s')\in\mathcal{SP}_F} {}^F\lambda_1^{s,s'} - \sum_{(s',s)\in\mathcal{SP}_F} {}^F\lambda_1^{s',s}\right)\end{aligned} \tag{C.3}$$

where index s is *fixed* in $(s,s') \in \mathcal{SP}_F$ and $(s',s) \in \mathcal{SP}_F$ based on the value of $s \in \mathcal{S}$ in the outer summation. Also note that $\sum_{(s,s')\in\mathcal{SP}_F} {}^F\lambda_1^{s,s'} - \sum_{(s',s)\in\mathcal{SP}_F} {}^F\lambda_1^{s',s}$ represents the difference between the sum of multipliers for all *outgoing* arcs from scenario s and the sum of multipliers for all *incoming* arcs to scenario s at the beginning of the first time period (where arcs refer to non-anticipativity constraints involving scenario s).

We follow the same general procedure to reformulate the penalty terms corresponding to the exogenous and fixed endogenous NACs (while being mindful of the time index). The objective function can then be expressed as Equation (5.12), which allows us to easily decompose the problem into independent scenario subproblems that can be solved in parallel.

C.2 Heuristic for Lagrangean Decomposition

In Lagrangean decomposition, a heuristic is needed in order to determine an upper bound (i.e., feasible solution). For the heuristic implemented here, we accomplish this by selectively fixing decisions from the Lagrangean subproblems in the original problem and then solving the resulting model. This heuristic consists of three main parts: (1) fixing the integer first-stage decisions in all scenarios to satisfy the first-period non-anticipativity constraints; (2) fixing the integer decisions for all other time periods in *some* scenarios to satisfy the endogenous NACs; and (3) fixing the integer decisions for all other time periods in all remaining scenarios to satisfy the exogenous NACs. We next discuss each step in detail.

To satisfy the first-period NACs, we first identify from the solution of the Lagrangean subproblems the scenario with an unweighted total cost that is closest to the total expected cost (neglecting all penalty terms). We then extract the integer first-stage decisions from that scenario, and fix these decisions in *all* scenarios.

To satisfy the endogenous NACs, we first select one representative scenario from each exogenous scenario group in time periods $t \in \mathcal{T} \setminus \{T\}$. For convenience, we select the lowest-indexed scenario (i.e., the first scenario) from each group. For the final time period, we select all scenarios. We then begin at $t = 1$ and check for indistinguishability among these scenarios at the end of the first period, based on the first-stage decisions previously fixed and whether or not the uncertainty in the endogenous parameters can be resolved at that point in time. For each representative scenario, we use this information to define the set of all representative scenarios from which this scenario is indistinguishable (including itself).

We assign the same integer recourse decisions for the end of this period, and all integer here-and-now decisions for the beginning of the next time period, to all scenarios in each of these sets, respectively, based on the decisions of one scenario in the set. This scenario will be one for which decisions have already been fixed, or if decisions have not been fixed for any of the scenarios, it will be the scenario with an unweighted total cost closest to the total expected cost of all of the scenarios in the set (neglecting all penalty terms). Note that when calculating the total expected cost among these scenarios, we proportionally scale the respective scenario probabilities such that they sum to 1. This step is performed in a serial fashion.

After this procedure, any of the scenarios for which decisions have not yet been fixed can be deemed distinguishable from the other representative scenarios. Accordingly, we fix their respective integer recourse decisions for the end of this period, and their integer here-and-now decisions for the beginning of the next time period, based on the decisions from the corresponding Lagrangean subproblems without making any further modifications. We then proceed to the next time period and check for indistinguishability among the representative scenarios, taking into consideration the decisions fixed in the previous step, whether or not the scenarios were distinguishable at the end of the previous time period, and whether or not the uncertainty in the endogenous parameters can be resolved at this point in time. As before, we use this indistinguishability information to successively fix decisions. We repeat this process until we have reached the end of the final time period.

Similar to the sequential scenario decomposition heuristic, for each exogenous scenario group, we extract the integer decisions from *one* representative scenario and fix these decisions in all scenarios of that respective group in order to satisfy the corresponding exogenous NACs. This representative scenario will be the same one chosen during the previous step for the endogenous NACs (and thus we use the *fixed* integer decisions from this scenario rather than the original decisions from the Lagrangean subproblem).

At this point, all integer decisions are fixed, and we solve the resulting LP to obtain a valid upper bound. Note that this heuristic is executed in the first iteration only in cases where an initial upper bound has not been specified. In all subsequent iterations, the heuristic is evaluated only if the best *lower* bound has improved from the previous iteration by an appreciable amount (e.g., 0.1%). These restrictions are effective in reducing the overall running time of the algorithm, since it is often unnecessary (and computationally intensive) to update the upper bound in every iteration.

C.3 Number of Scenarios Considered in the SSD Heuristic

Due to the fact that no binary decisions have been fixed in the first subproblem of the SSD heuristic, there is always an 'offset' observed in the number of scenarios in the second subproblem. The number of scenarios in the two will not always be equal, however. For instance, in Example 1, if we were to consider 3 exogenous realizations in each time period, we would have 27 scenarios in the first subproblem and 54 in the second.

We further observe in this example that in each subsequent subproblem, the number of scenarios doubles. This is because we are past the offset observed in the second subproblem, and there are 2 realizations for the exogenous parameter in each time period. If there were 3 realizations for the exogenous parameter in each period, we would see the number of scenarios in each subproblem begin to triple at this point.

We next generalize these observations (for the particular class of problems considered in this thesis) in order to easily calculate the number of scenarios considered in the SSD heuristic.

First, we note that Equation (5.1) may be equivalently expressed as:

$$\mathcal{S}_{SSD}^{1} := \{s : \, s \in \tilde{\mathcal{U}}_1\} \tag{C.4}$$

for $\hat{t} = 1$, and:

$$\mathcal{S}_{SSD}^{\hat{t}} := \left\{ s : \, s \in \tilde{\mathcal{U}}_{\hat{t}} \setminus \bigcup_{\hat{\tau}=1}^{\hat{t}-1} \mathcal{S}_{SSD}^{\hat{\tau}} \right\} \qquad \forall\, \hat{t} \in \mathcal{T}, \;\; 1 < \hat{t} < T \tag{C.5}$$

for $\hat{t} = 2, \ldots, T-1$. Before proceeding, we can further simplify Equation (C.5) by reformulating the term $\bigcup_{\hat{\tau}=1}^{\hat{t}-1} \mathcal{S}_{SSD}^{\hat{\tau}}$. Notice that for $\hat{t} = 2$, we have $\bigcup_{\hat{\tau}=1}^{1} \mathcal{S}_{SSD}^{\hat{\tau}} = \tilde{\mathcal{U}}_1$; for $\hat{t} = 3$, we have $\bigcup_{\hat{\tau}=1}^{2} \mathcal{S}_{SSD}^{\hat{\tau}} = \tilde{\mathcal{U}}_1 \cup (\tilde{\mathcal{U}}_2 \setminus \tilde{\mathcal{U}}_1) = \tilde{\mathcal{U}}_2$; for $\hat{t} = 4$, we have $\bigcup_{\hat{\tau}=1}^{3} \mathcal{S}_{SSD}^{\hat{\tau}} = \tilde{\mathcal{U}}_1 \cup (\tilde{\mathcal{U}}_2 \setminus \tilde{\mathcal{U}}_1) \cup (\tilde{\mathcal{U}}_3 \setminus (\tilde{\mathcal{U}}_2 \cup \tilde{\mathcal{U}}_1)) = \tilde{\mathcal{U}}_1 \cup (\tilde{\mathcal{U}}_2 \setminus \tilde{\mathcal{U}}_1) \cup (\tilde{\mathcal{U}}_3 \setminus \tilde{\mathcal{U}}_2) = \tilde{\mathcal{U}}_3$; and so forth. The ability to simplify the expressions for $\hat{t} \geq 3$ relies on the fact that the set of unique scenarios in time period $\hat{t}$ includes all unique scenarios from the previous time periods (i.e., $\tilde{\mathcal{U}}_{\hat{\tau}} \subseteq \tilde{\mathcal{U}}_{\hat{t}} \;\; \forall\, \hat{\tau} < \hat{t}$), as can easily be seen by selecting the first scenario from each exogenous scenario group at $t = 1$ and $t = 2$ in Figure 5.1a. It follows that:

$$\bigcup_{\hat{\tau}=1}^{\hat{t}-1} \mathcal{S}_{SSD}^{\hat{\tau}} = \tilde{\mathcal{U}}_{\hat{t}-1} \qquad \forall\, \hat{t} \in \mathcal{T}, \;\; 1 < \hat{t} < T \tag{C.6}$$

We can then make this replacement in Equation (C.5) to obtain:

$$\mathcal{S}_{SSD}^{\hat{t}} := \left\{ s : \, s \in \tilde{\mathcal{U}}_{\hat{t}} \setminus \tilde{\mathcal{U}}_{\hat{t}-1} \right\} \qquad \forall\, \hat{t} \in \mathcal{T}, \;\; 1 < \hat{t} < T \tag{C.7}$$

By Equation (C.4), the number of scenarios considered in the first subproblem can be expressed as:

$$|\mathcal{S}_{SSD}^{1}| = |\mathcal{K}_1| \tag{C.8}$$

Similarly, by Equation (C.7), the number of scenarios considered in subsequent subproblems can be expressed as:

$$|\mathcal{S}_{SSD}^{\hat{t}}| = |\mathcal{K}_{\hat{t}}| - |\mathcal{K}_{\hat{t}-1}| \qquad \forall\, \hat{t} \in \mathcal{T},\ \ 1 < \hat{t} < T \tag{C.9}$$

The total number of scenarios considered by the SSD heuristic is then:

$$\begin{aligned} \sum_{\hat{t} \in \mathcal{T} \setminus \{T\}} |\mathcal{S}_{SSD}^{\hat{t}}| = |\mathcal{K}_1| + (|\mathcal{K}_2| - |\mathcal{K}_1|) + (|\mathcal{K}_3| - |\mathcal{K}_2|) + \cdots \\ + (|\mathcal{K}_{T-1}| - |\mathcal{K}_{T-2}|) \end{aligned} \tag{C.10}$$

which reduces to:

$$\sum_{\hat{t} \in \mathcal{T} \setminus \{T\}} |\mathcal{S}_{SSD}^{\hat{t}}| = |\mathcal{K}_{T-1}| \tag{C.11}$$

This is equivalent to the number of groups in the second-to-last time period. The explanation here is that we select one scenario from each exogenous scenario group, and the maximum number of groups for $\hat{t} \in \mathcal{T} \setminus \{T\}$ occurs at $\hat{t} = T - 1$.

Appendix D Quinn's Car Buying Problem

D.1 Deterministic Model

The deterministic model for Quinn's car-buying problem can be formulated as follows. We attempt to use straightforward nomenclature to keep the discussion as simple as possible, and for this reason, we purposely allow the notation to conflict with earlier chapters in this thesis. Here, Z is the objective function value, which is equal to the total cost of ownership over a 5-year period. Indices i and j are used to refer to the first, second, or third car, and p_i is the purchase price of car i, whereas d_i is the depreciation of car i after 5 years. Note that j is an alias of i, and p_i and d_i are input parameters. As far as decision variables are concerned, we have b_i, which is a binary variable that indicates whether or not to buy car i now, as well as y_j, which is another binary variable that indicates whether or not to switch to car j later.

$$\min_{b,y} Z = Cost \tag{D.1}$$

$$\text{s.t.} \quad Cost = \left(1 - \sum_{j=1}^{3} y_j\right) \sum_{i=1}^{3} b_i d_i + \sum_{j=1}^{3} y_j \left(d_j + Penalty\right) \tag{D.2}$$

$$\sum_{i=1}^{3} b_i = 1 \tag{D.3}$$

$$\sum_{j=1}^{3} y_j \leq 1 \tag{D.4}$$

$$b_i + y_i \leq 1 \qquad i = 1, 2, 3 \tag{D.5}$$

$$Bonus \geq \left(1 - \sum_{j=1}^{3} y_j\right) \sum_{i=1}^{3} b_i p_i \tag{D.6}$$

$$Bonus \geq \sum_{j=1}^{3} y_j p_j \tag{D.7}$$

$$Penalty = \sum_{i=1}^{3} b_i \cdot 0.10 p_i \tag{D.8}$$

$$b_i, y_j \in \{0, 1\} \qquad i, j = 1, 2, 3 \tag{D.9}$$

The objective function, Equation (D.1), attempts to minimize the total cost of ownership. This is subject to the restrictions described in constraints (D.2)–(D.9). We next describe each of these equations in detail.

Equation (D.2) defines the total cost of ownership, which is equal to the purchase price of the originally-chosen car *if* Quinn does not switch to a different car later on (i.e., $y_j = 0$ for all j). If Quinn *does* change to a different car (i.e., $y_j = 1$ for one j), the cost will instead be equal to the cost of this new car plus applicable penalties (defined in Equation (D.8) as 10% of the purchase price of the originally-selected car).

Equation (D.3) simply states that Quinn must purchase *exactly one* car, and Equation (D.4) states that she can switch to only one car, *at most*. Equation (D.5) is slightly more abstract and ensures that Quinn cannot switch to the same car that she originally agreed to purchase (in other words, b_i and y_i cannot simultaneously be equal to 1).

Equation (D.6) states that if Quinn has agreed to purchase car i, and she has not switched cars, her bonus amount must be greater than or equal to the corresponding purchase price. This constraint is ignored if she does in fact switch to a different car. In such a case, Equation (D.7) then specifies that her bonus amount must be greater than or equal to the purchase price of the newly-selected car.

Finally, Equation (D.9) defines b_i and y_j as binary variables.

D.2 Stochastic Programming Model

The stochastic programming model is a fairly straightforward extension of the deterministic model. Specifically, we replace the objective function with the total *expected* cost of ownership using the probability of each scenario, $Prob^s$, and we index all variables for each scenario s. (Note that superscripts here denote indices and *not* exponents.) The only other change to the model is the addition of the non-anticipativity constraints, shown in Equation (D.18). These constraints link the 3 scenarios such that the same car purchasing decision, b_i^s, must be made at the beginning of each scenario. Note that, here, decision variables b_i and y_j refer specifically to here-and-now and recourse decisions, respectively.

$$\min_{b,y} \tilde{Z} = \sum_{s=1}^{3} \left(Prob^s \cdot Cost^s\right) \tag{D.10}$$

$$\text{s.t.} \quad Cost^s = \left(1 - \sum_{j=1}^{3} y_j^s\right) \sum_{i=1}^{3} b_i^s d_i + \sum_{j=1}^{3} y_j^s \left(d_j + Penalty^s\right) \qquad s = 1,2,3 \tag{D.11}$$

$$\sum_{i=1}^{3} b_i^s = 1 \qquad s = 1,2,3 \tag{D.12}$$

$$\sum_{j=1}^{3} y_j^s \leq 1 \qquad s = 1,2,3 \tag{D.13}$$

$$b_i^s + y_i^s \leq 1 \qquad i = 1,2,3, \quad s = 1,2,3 \tag{D.14}$$

$$Bonus^s \geq \left(1 - \sum_{j=1}^{3} y_j^s\right) \sum_{i=1}^{3} b_i^s p_i \qquad s = 1,2,3 \tag{D.15}$$

$$Bonus^s \geq \sum_{j=1}^{3} y_j^s p_j \qquad s = 1,2,3 \tag{D.16}$$

$$Penalty^s = \sum_{i=1}^{3} b_i^s \cdot 0.10 p_i \qquad s = 1,2,3 \tag{D.17}$$

$$b_i^s = b_i^{s+1} \qquad i = 1,2,3, \quad s = 1,2 \tag{D.18}$$

$$b_i^s, y_j^s \in \{0,1\} \qquad i,j = 1,2,3, \quad s = 1,2,3 \tag{D.19}$$

D.3 Reformulation

One complicating aspect regarding the original model is that it is a mixed-integer nonlinear programming problem in its current form. This is due to bilinear terms in Equations (D.2) and (D.6). Although we can solve the problem with a global optimization solver such as BARON, it is possible to instead reformulate it as a mixed-integer *linear* programming problem due to the fact that the bilinear terms involve only binary variables.

To do so, we first let $Y = \sum_{j=1}^{3} y_j$. Note that Y, which is a sum of binary variables, is also binary because only *one* y_j can be active at any one time (see Equation (D.4)). We also remove Equation (D.8) and substitute it directly into the objective function. Equations (D.2) and (D.6) then become:

$$Cost = \sum_{i=1}^{3} b_i d_i - \sum_{i=1}^{3} b_i Y d_i + \sum_{j=1}^{3} y_j d_j + 0.10 \sum_{j=1}^{3} \left(\sum_{i=1}^{3} b_i y_j p_i\right) \tag{D.20}$$

$$Bonus \geq \sum_{i=1}^{3} b_i p_i - \sum_{i=1}^{3} b_i Y p_i \tag{D.21}$$

Notice that we now have bilinear terms $b_i Y$ and $b_i y_j$. Let $g_i = b_i Y$ and $h_{i,j} = b_i y_j$ replace these bilinear terms. Since b_i, Y, and y_j are all binary, we have $b_i \wedge Y \Leftrightarrow g_i$ and $b_i \wedge y_j \Leftrightarrow h_{i,j}$. Making these replacements and reformulating the logic expressions yields the following:

$$Cost = \sum_{i=1}^{3} b_i d_i - \sum_{i=1}^{3} g_i d_i + \sum_{j=1}^{3} y_j d_j + 0.10 \sum_{j=1}^{3} \left(\sum_{i=1}^{3} h_{i,j} p_i \right) \tag{D.22}$$

$$Bonus \geq \sum_{i=1}^{3} b_i p_i - \sum_{i=1}^{3} g_i p_i \tag{D.23}$$

$$g_i \geq b_i + Y - 1 \qquad i = 1, 2, 3 \tag{D.24}$$

$$g_i \leq b_i \qquad i = 1, 2, 3 \tag{D.25}$$

$$g_i \leq Y \qquad i = 1, 2, 3 \tag{D.26}$$

$$h_{i,j} \geq b_i + y_j - 1 \qquad i, j = 1, 2, 3 \tag{D.27}$$

$$h_{i,j} \leq b_i \qquad i, j = 1, 2, 3 \tag{D.28}$$

$$h_{i,j} \leq y_j \qquad i, j = 1, 2, 3 \tag{D.29}$$

$$0 \leq g_i \leq 1 \qquad i = 1, 2, 3 \tag{D.30}$$

$$0 \leq h_{i,j} \leq 1 \qquad i, j = 1, 2, 3 \tag{D.31}$$

At this point, we may return Y to its original form, $\sum_{j=1}^{3} y_j$. We also simplify Equations (D.22) and (D.23). (Note that, for clarity, we do not renumber these equations.) The reformulated deterministic model in its complete form is as follows:

$$\min_{b,y} Z = Cost \tag{D.1}$$

$$\text{s.t.} \quad Cost = \sum_{i=1}^{3} (b_i - g_i) d_i + \sum_{j=1}^{3} y_j d_j + 0.10 \sum_{j=1}^{3} \left(\sum_{i=1}^{3} h_{i,j} p_i \right) \tag{D.22}$$

$$\sum_{i=1}^{3} b_i = 1 \tag{D.3}$$

$$\sum_{j=1}^{3} y_j \leq 1 \tag{D.4}$$

$$b_i + y_i \leq 1 \qquad i = 1, 2, 3 \tag{D.5}$$

$$Bonus \geq \sum_{i=1}^{3} (b_i - g_i) p_i \tag{D.23}$$

$$Bonus \geq \sum_{j=1}^{3} y_j p_j \tag{D.7}$$

$$g_i \geq b_i + \sum_{j=1}^{3} y_j - 1 \qquad i = 1, 2, 3 \tag{D.24}$$

$$g_i \leq b_i \qquad i = 1, 2, 3 \tag{D.25}$$

$$g_i \leq \sum_{j=1}^{3} y_j \qquad i = 1,2,3 \tag{D.26}$$

$$h_{i,j} \geq b_i + y_j - 1 \qquad i,j = 1,2,3 \tag{D.27}$$

$$h_{i,j} \leq b_i \qquad i,j = 1,2,3 \tag{D.28}$$

$$h_{i,j} \leq y_j \qquad i,j = 1,2,3 \tag{D.29}$$

$$b_i, y_j \in \{0,1\} \qquad i,j = 1,2,3 \tag{D.9}$$

$$0 \leq g_i \leq 1 \qquad i = 1,2,3 \tag{D.30}$$

$$0 \leq h_{i,j} \leq 1 \qquad i,j = 1,2,3 \tag{D.31}$$

The corresponding stochastic programming formulation is then:

$$\min_{b,y} \tilde{Z} = \sum_{s=1}^{3} (Prob^s \cdot Cost^s) \tag{D.10}$$

$$\text{s.t.} \quad Cost^s = \sum_{i=1}^{3} (b_i^s - g_i^s)d_i + \sum_{j=1}^{3} y_j^s d_j + 0.10 \sum_{j=1}^{3} \left(\sum_{i=1}^{3} h_{i,j}^s p_i \right) \qquad s = 1,2,3 \tag{D.32}$$

$$\sum_{i=1}^{3} b_i^s = 1 \qquad s = 1,2,3 \tag{D.12}$$

$$\sum_{j=1}^{3} y_j^s \leq 1 \qquad s = 1,2,3 \tag{D.13}$$

$$b_i^s + y_i^s \leq 1 \qquad i = 1,2,3, \quad s = 1,2,3 \tag{D.14}$$

$$Bonus^s \geq \sum_{i=1}^{3} (b_i^s - g_i^s)p_i \qquad s = 1,2,3 \tag{D.33}$$

$$Bonus^s \geq \sum_{j=1}^{3} y_j^s p_j \qquad s = 1,2,3 \tag{D.16}$$

$$g_i^s \geq b_i^s + \sum_{j=1}^{3} y_j^s - 1 \qquad i = 1,2,3, \quad s = 1,2,3 \tag{D.34}$$

$$g_i^s \leq b_i^s \qquad i = 1,2,3, \quad s = 1,2,3 \tag{D.35}$$

$$g_i^s \leq \sum_{j=1}^{3} y_j^s \qquad i = 1,2,3, \quad s = 1,2,3 \tag{D.36}$$

$$h_{i,j}^s \geq b_i^s + y_j^s - 1 \qquad i,j = 1,2,3, \quad s = 1,2,3 \tag{D.37}$$

$$h_{i,j}^s \leq b_i^s \qquad i,j = 1,2,3, \quad s = 1,2,3 \tag{D.38}$$

$$h_{i,j}^s \leq y_j^s \qquad i,j = 1,2,3, \quad s = 1,2,3 \tag{D.39}$$

$$b_i^s = b_i^{s+1} \qquad i = 1,2,3, \quad s = 1,2 \tag{D.18}$$

$$b_i^s, y_j^s \in \{0,1\} \qquad i,j = 1,2,3, \quad s = 1,2,3 \tag{D.19}$$

$$0 \leq g_i^s \leq 1 \qquad i = 1,2,3, \quad s = 1,2,3 \tag{D.40}$$

$$0 \leq h^s_{i,j} \leq 1 \qquad i, j = 1, 2, 3, \;\; s = 1, 2, 3 \tag{D.41}$$