Benjamin S. Narum

Problem-based scenario generation in stochastic programming with binary distributions

Case study in Air Traffic Flow Management

Master's thesis in Industrial Economics and Technology Management
Supervisor: Stein W. Wallace and Jamie Fairbrother
July 2020

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Economics and Management
Dept. of Industrial Economics and Technology Management

NTNU
Norwegian University of Science and Technology

Benjamin S. Narum

Problem-based scenario generation in stochastic programming with binary distributions

Case study in Air Traffic Flow Management

Master's thesis in Industrial Economics and Technology Management
Supervisor: Stein W. Wallace and Jamie Fairbrother
July 2020

Norwegian University of Science and Technology
Faculty of Economics and Management
Dept. of Industrial Economics and Technology Management

NTNU
Norwegian University of
Science and Technology

Abstract

This thesis considers problem-based scenario generation for stochastic problems with binary distributions. Stability, that is, the sensitivity of a stochastic program to the scenario set has been studied from both mathematical and empirical perspectives to develop a new problem-based scenario generation method. An output-distribution centric view of scenario generation and stochastic programming is introduced, contrasting recent literature on the field, and is used directly for scenario generation and as a tool to analyse stability and 'presence of uncertainty' in stochastic models. Scenario generation is considered more deeply for a stochastic model in Air Traffic Flow Management with a binary distribution to highlight the value of this contribution.

Summary

Scenario generation is about selecting which outcomes of the future are worth considering when solving a stochastic optimization problem, and to remove redundancies in the full representation of the stochastic phenomenon to be able to solve a decision problem.

This thesis finds that analysing a collection of output-distributions resulting from a restricted and relevant set of first-stage decisions is sufficient to find the problem structure which makes the formulation unstable and that these can be compensated against by constructing appropriate scenario sets based on empirically analysing such a collection of output-distributions.

These insights are applied to make a new scenario generation method for the particular case of binary distributions. This problem type is exceptionally well suited for problem-based scenario generation due to the high impact of changes in the stochastic variables, and the lack of alternative scenario generation methods for such problems makes this a valuable contribution. Three different clustering methods suited for binary domains are suggested and compared to guide how to choose one based on the specific problem.

The motivation for developing better scenario generation procedures is to solve large-scale, often combinatorial, stochastic models which otherwise cannot be solved. The proposed scenario generation method is therefore applied to a large-scale, combinatorial two-stage model in Air Traffic Flow Management specified by a binary distribution function. The model integrates strategic decision making across entire air traffic networks and integration between the strategic and tactical planning stages. It was shown that the suggested scenario generation procedure gave better and more reliable solutions than all other alternatives considered. Thus, the proposed scenario generation procedure creates a more accurate and reliable representation of the output-distribution to solve the given problem.

Preface

This master's thesis concludes my five year integrated Master of Science (M.Sc) program in 'Industrial Economics and Technology Management' at the Norwegian University of Science and Technology. A project report on the same theme, 'Problem-based Scenario Generation in Stochastic Linear Programming' (Narum, 2019), also preceded the thesis.

Thank you very much, Stein W. Wallace and Jamie Fairbrother, for excellent supervision on this thesis. Stein, thank you for helping me see what is important and keeping track of the high-level meaning of the work, I have learned much from insightful discussions. Jamie, thank you for great discussions, enjoyable meetings and essential feedback. Our discussions to clarify and explain complicated lines of reasoning have been crucial for this thesis.

I want to thank Alexandre Jacquillat and Kai Wang at the MIT Sloan School of Management for sharing their paper 'A Stochastic Integer Programming Approach to Air Traffic Scheduling and Operations' (Wang & Jacquillat, 2020), accompanying datasets and for answering questions. The model formulation provided great value as a case study problem for the work in this thesis.

Benjamin S. Narum

Trondheim 2020

Contents

1 Introduction **1**

2 Background and theory **5**
2.1 Problem-based scenario generation . . . 7
2.2 Stability . . . 8
2.3 Discussion . . . 10

3 Literature review **12**
3.1 Scenario generation by sample first-stage decisions . . . 12
3.2 Binary distributions . . . 13
3.3 Clustering methods for scenario generation . . . 13
3.4 Scenario reduction . . . 14
3.5 Problem-based scenario generation . . . 16

4 Integrated Model of Scheduling and Operations in Airport Networks **17**
4.1 Model description . . . 17
4.2 Mathematical formulation . . . 18
4.2.1 First-stage formulation . . . 22
4.2.2 Second-stage formulation . . . 22
4.3 Solution procedure . . . 24
4.3.1 Decomposition formulation . . . 24
4.4 Scenario generation in (Wang & Jacquillat, 2020) . . . 24
4.5 Problem characteristics . . . 25

5 Scenario generation **26**
5.1 Departing from accurate representations of the input-distribution . . . 27
5.2 Proximity of outcomes . . . 28
5.3 Collections of output-distributions . . . 30
5.3.1 Minimum transportation distance from collections of output-distributions . 31
5.3.2 Comparison with the Fortet-Mourier probability metric . . . 32
5.4 Clustering variables on collections of output-distributions . . . 33
5.5 Clustering method . . . 35
5.5.1 K-median . . . 36
5.5.2 K-means . . . 36
5.5.3 Binary point centroid clustering . . . 37
5.6 Recourse deviation . . . 38

6 Numerical experiments **39**
6.1 Experimental set-up . . . 39
6.1.1 Parameter configurations . . . 39
6.1.2 Implementation details . . . 40
6.2 Stability benchmarks . . . 41
6.2.1 Sample Average Approximation . . . 41

6.2.2 Scenario Reduction . . . 42
6.2.3 Recourse deviation . . . 43
6.2.4 Discussion on stability . . . 43
6.3 Comparison of scenario sets . . . 45
6.3.1 Average corrected feature vectors . . . 46
6.3.2 Explanatory capabilities of output-distributions . . . 46

7 Discussion and conclusion **51**
7.1 Discussion . . . 51
7.1.1 Scenario generation using collections of output-distributions . . . 51
7.1.2 Modeling insights from output-distributions . . . 53
7.1.3 Stability driving tractability of scenario generation . . . 54
7.1.4 Generalized problem-based scenario generation . . . 56
7.2 Conclusion . . . 57
7.3 Future research . . . 57

List of Figures

2.1 Two-stage scenario tree. 6
2.2 Multistage scenario tree. 6
2.3 The output-distribution of (a) a well chosen first-stage decision, and (b) a bad first-stage decision. Taken from a facility location problem where the objective is to minimize the expected cost. Evaluated for 60 000 sampled outcomes. 7

4.1 Modelling structure of departure and arrival time with binary variables. The departure/arrival time is in period 5 since 5 is the last time index of binary value 1. 19
4.2 Capacity envelope at an airport. 20

5.1 Flowchart of how various quantities are obtained. 26
5.2 Illustration of different notions of proximity, see explanation in text. 35

6.1 Recourse distribution for problem instance K6. The plot shows the out-of-sample distribution of ten first-stage solutions from solving the problem with ten sampled scenarios, and the recourse deviation is the span between the minimum and maximum recourse value for each scenario. 44
6.2 Comparison of scenario sets in-sample (dots with annotated probabilities) and out-of-sample (line). 48
6.3 Comparison of scenario sets in the input domain for different generation methods. Black is bad weather, yellow is good weather. Probabilities for each scenario is annotated above. 49
6.4 Objective values plotted against occurrences of bad weather. First-stage decision from solving with a sampled scenario set with $\rho = 0.2$. 50

List of Tables

4.1 Overview of all variables and parameters. 21

6.1 All parameter configurations used in the numerical experiments. 39
6.2 Model constants and dataset key figures. 40
6.3 Stability of Sample Average Approximation. Mean and standard deviation of the evaluation of the in-sample and out-of-sample expected objective values among ten sampled scenario sets of respective sizes. "—" = memory issues. 42
6.4 Stability of scenario reduction. 43
6.5 Comparison of clustering scenario sets solved with $\rho = 0.2$. Objective evaluations for scenario generation procedure using each distance metric and clustering method for different set sizes. The suggested new scenario generation method using feature vector distances highlighted in green, scenario reduction as a benchmark highlighted in blue. "—" = memory issues. 45
6.6 Average corrected feature distance by k-median clustering for $\rho = 0.2$. Benchmark is scenario reduction, also highlighted in blue in Table 6.5. The difference from the benchmark is highlighted in green. 47

1 Introduction

There exists a vast collection of decision problems where uncertainty is an inherent property that is essential to consider, and neglecting the effects of uncertainty when modelling a problem can result in poor decisions. The challenge in stochastic programming is that the stochastic version of the problem is significantly more demanding to solve, and the added value can be difficult to determine without implementing the model first.

The uncertainty addressed in this thesis an inherent property of the problems, and cannot be removed by collecting more information, or without rendering the results of the model useless. The decision could, for example, be time-sensitive and waiting for more detailed information by letting uncertainty reveal itself would make the decision unnecessary because time has already passed.

In our context, we specify the uncertainty by a distribution function, either based on empirical data or by knowing the analytical distribution. Modelling these decision problems can be difficult, and a natural question to ask is whether it is possible to collapse the probability distribution by representing it with a single value, say the expectation, and still get an accurate solution. The short answer is no. The real answer is that it all depends on the problem at hand. If you solve the deterministic version of the problem, defined by using a single outcome value instead of a distribution, you will not know by how much the solution is suboptimal in the (real) stochastic setting. The deterministic version can solve some problems adequately but which problems this holds for is a question of experience and research.

There are two ends of the spectrum for models considering uncertainty. On one side, there is the deterministic version which can give inadequate solutions and on the other, there is the stochastic version with a full description of the distribution which we cannot solve in most cases. We cannot use the full description of the distribution function either because it is continuous, and the solution method requires a discrete distribution, or because of an exponentially large amount of outcomes. Discretizing and simplifying the probability distribution is a compromise between the deterministic and exact stochastic problem formulations, and the quality of the discrete distribution determines the impact of this compromise.

We don't solve all problems by its stochastic version because formulating and solving stochastic models is often more laborious, and the deterministic version is more straightforward, if a deterministic counterpart exists. Additionally, stochastic programs have a higher overhead from solving for multiple scenarios, meaning that size and complexity of the problem will catch up with you more quickly if the problem is of large scale or high complexity. Thus, solving the model within a reasonable time isn't tractable. Modelling problems by considering uncertainty requires precise modelling assumptions to reduce size. It is said in the stochastic programming research community that they are still only solving toy-sized versions of the problems they want to solve (Sen, 2019, in conference).

The task of making a concise discrete representation of a probability distribution is called *scenario generation*. The resulting discrete distribution is called a *scenario set* which consists of a set of *outcomes* with corresponding *probabilities*. One outcome, together with its probability, is referred to as one *scenario*.

The quality of the scenario set is essential to get reliable solutions from a stochastic model. Because of this interdependence between the reliability of the final solution and the scenario set, scenario generation is part of the modelling itself. It cannot be considered preliminary data manipulation, and we must develop it alongside the solution procedure (King & Wallace, 2012).

This thesis will explore problem-based approaches of scenario generation to construct more concise scenario sets. Problem-based means that specific properties about the problem, and not only the distribution, is utilized to generate higher quality scenario sets.

Rationale

Scenario generation has historically been driven by the probability distribution of the physical phenomenon and how to convert that distribution into a statistically most accurate scenario set. This approach does not directly take in to account the fact that the optimization model transforms the scenario set into a completely different output-distribution, and that this output-distribution is what determines the final solution of the model. Distribution-based scenario generation will work as long as the number of scenarios is large enough, but if the problem is too difficult to solve, the high number of scenarios can make it intractable.

Recent advances in optimization have enabled us to solve large-scale problems and NP-hard combinatorial problems. The stochastic version of such problems is even larger. With a linear increase in the number of stochastic variables, the number of dimensions of the distribution grows linearly, which in turn causes an exponential growth in the number of possible outcomes. This exponential growth is what makes such models unsolvable. At the same time, there is no end to the number of applications within energy planning, transportation, logistics, finance and engineering that require solving large-scale problems.

Developing better scenario generation procedures is one of two approaches to make these problems tractable. The second approach is to develop more efficient solution techniques. It is important to note that these two approaches are complementary, and thus a contribution to scenario generation serve great value on top of potential improvements in solution procedures.

Problem-based scenario generation is about looking more closely at the specific problem, problem class or model formulation to generate scenarios. We can utilize this additional information about the problem to reduce the number of scenarios further than what is possible with distribution-based methods.

The literature on problem-based scenario generation has seen only recent advancements in the last couple of years and is therefore still forming. Next chapter clarifies problem-based scenario generation as a concept. Additionally, a specific problem which is of interest in itself is solved to showcase the use of problem-based methods for scenario generation. The case study problem is an Air Traffic Flow Management problem, which is a hard combinatorial large-scale problem with binary input distribution.

Binary distributions are of particular interest for decision problems because of the implications they represent. A binary stochastic variable could represent a network link failing, whether a customer is present or not, or significant qualitative differences in operating modes of a system. They are therefore more important than merely being a subset of general discrete distributions. In particular, scenario generation with binary distributions is an especially appropriate application of problem-based scenario generation since changes in a binary outcome vector represent significant shifts in behaviour within the modelled system, and it is therefore especially advantageous to encapsulate this behaviour when generating scenarios. Additionally, other scenario generation methodologies such as property matching (Høyland, Kaut & Wallace, 2003) and Sampling Average Approximation (SAA) (Shapiro, 2003) may become useless for these problems because of the binary domain. The literature on scenario generation for binary distributions has only a few papers, although many applications have formulations with binary distributions. This thesis seeks to expand the research literature on stochastic programming with binary distributions.

Problem description

This thesis aims to understand how problem structure in stochastic problems can be exploited to construct more concise scenario sets than would be possible by distribution-based methods. Finding problem structure is in itself a difficult task which is not well defined for most problem classes. Once the problem structure is found, scenario generation procedures should exploit it appropriately to give a more concise description of the uncertainty. Successfulness of such a procedure is determined by benchmarking the out-of-sample stability against other methods.

Contribution

This thesis finds that analysing collections of output-distributions resulting from a set of restricted relevant first-stage solutions can be used to find problem structures in stochastic programs, and shows that these structures can be exploited to generate scenario sets which give a more concise representation of the uncertainty than possible by distribution-based alternatives.

We propose a new computationally tractable problem-based scenario generation procedure which is agnostic to the particular kind of problem without the need for considerable tailoring. The method is developed with implementation in mind, and computational experiments on a large-scale combinatorial stochastic two-stage model in Air Traffic Flow Management (ATFM) specified by a binary distribution illustrates its successfulness.

Extra care is needed to make sure the method can be applied to problems with binary input-distributions, and we consider different three clustering methods for scenario generation. The most appropriate method comes down to a consideration of the specific problem and guidance for relevant considerations on the complexity of generating scenarios for binary distributions is given.

The newly introduced recourse deviation and analysis of collections of output-distributions give more nuanced information about instability and can serve as a proxy for evaluating the presence of uncertainty in stochastic problems. We exemplify how such insights can be found in particular on the case study problem.

The exposition of previous literature on problem-based scenario generation and the simultaneous consideration of mathematical and empirical stability theory for stochastic programming have not been done before in the literature and can also be considered a contribution in itself.

Outline

The thesis assumes a basic understanding of stochastic programming, which regards typical model formulations and solution procedures. Everything regarding scenario generation is explained or have references to relevant research literature. As a starting point, the reader is referred to either (Kall & Wallace, 1994) or (Birge & Louveaux, 2011) for theory on stochastic programming and solution procedures.

The structure of the thesis is as follows. Chapter 2 gives background and theory on problem-based scenario generation and the stability of stochastic programs. Chapter 3 is a literature review of what has been done previously with the techniques applied in this thesis, including stability and clustering methods for scenario generation. Chapter 4 introduces the case study problem in Air Traffic Flow Management (ATFM) for solving strategic planning of air traffic schedules by the integration of the entire airport network and between the strategic and tactical planning stages. Chapter 5 is an overview of how stability theory and clustering methods are combined in a new way to generate problem-based scenario sets for stochastic problems with binary distributions.

Chapter 6 contains numerical experiments, results and analyses of the ATFM problem. Lastly, Chapter 7 discusses and concludes the work.

2 Background and theory

Decision making under stochastic uncertainty is about making decisions which are ‘well hedged’ against uncertain outcomes in the future. Two-stage stochastic problems are the prototypical exemplifications of this problem class, and three fundamental aspects define them:

(I) A decision to be determined before some stochastic uncertainty is revealed

(II) A distribution function for the stochastic uncertainty

(III) Evaluation of the cost of the decision once the uncertainty has been revealed

A two-stage problem can be illustrated by a scenario tree as shown in Figure 2.1, where the root node represents the *first-stage decision* (I), the branches represents the possible stochastic outcomes (II), also called the scenario set, and the leaf nodes represents the cost evaluation (III) of the first-stage decision for each possible outcome, also referred to as the *second-stages*. A second-stage may also involve determining a second-stage decision.

In application, this formulation is commonly used to model strategic and tactical decisions in a way also to incorporate operational considerations into the first-stage decision. The operations then depend both on the first-stage strategic or tactical decision and the stochastic uncertainty. In logistics, this may involve deciding transportation routes in advance without knowing the demand at different stations, or determining which facilities to invest in when their utility is uncertain.

Multistage problems are an extension of two-stage problems with multiple phases of stochastic uncertainty separated by consecutive decisions. An example from applications includes portfolio selection where the portfolio has to be rebalanced at certain time steps considering newly revealed information about the uncertainty at each stage. Conceptually, multistage problems can be seen as recursive two-stage problems because the cost evaluation of stage number two in a multistage problem involves solving another two-stage problem where the next level may or may not be another two-stage problem. This is illustrated in Figure 2.2.

In this thesis, we consider only the two-stage setting as multistage problems are often more complicated and require more sophisticated approximations. The essential ideas presented in the context of two-stage stochastic programs are also relevant for more general stochastic problems and meaningful for multistage problems. Expanding these ideas entirely to the multistage setting could be a pertinent consideration for further research.

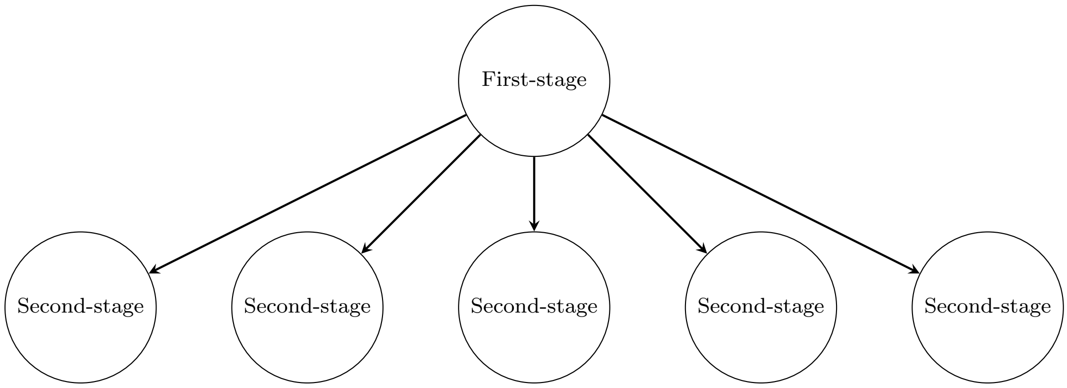


Figure 2.1: Two-stage scenario tree.

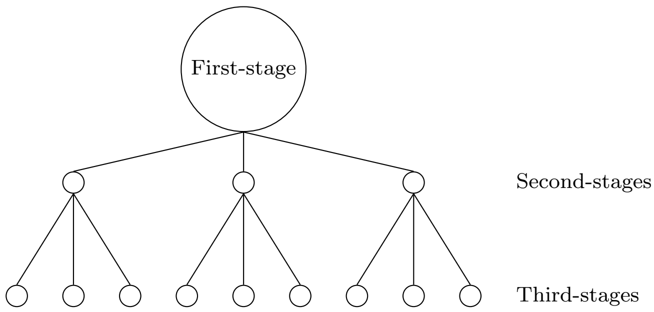
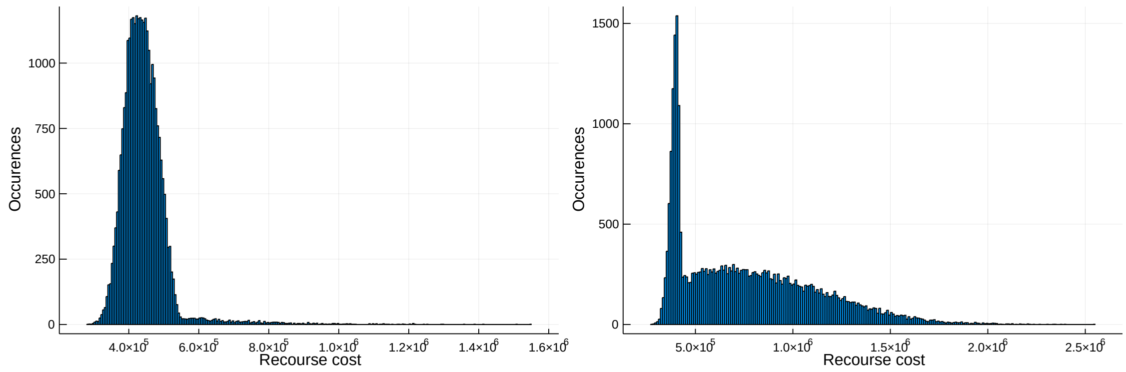


Figure 2.2: Multistage scenario tree.

The decision problem under stochastic uncertainty is the task of *selecting the first-stage decision that yields the best distribution of costs*, also referred to as the *output-distribution*. To find the output-distribution is known as the *distribution problem* (Wets, 1996). Unfortunately, the second-stage cost evaluation is often computationally difficult to determine and the preferred approach to solve the distribution problem is to sample different outcomes and evaluate for those.

The response of the distribution from changing the first-stage decision is therefore in general challenging to determine ex ante. Figure 2.3 shows the output-distribution evaluated at two different first-stage decisions for a facility location problem where the objective is to minimize the cost of delivering goods to customers. We see that the distribution in Figure 2.3b has a very irregular shape characteristic and that the one in Figure 2.3a is qualitatively very different. These variations in the output-distribution emphasize what a challenge it is to approximate these kinds of distributions. Keep in mind that each data point in the histogram is the result of solving an optimization problem with the stochastic outcome and first-stage decision as parameters.

To distinguish an output-distribution as better or worse, we need to consider a metric of utility on the distribution. The most common one would be the expected value, followed by tail-risk measures and others.



(a) **(b)**

Figure 2.3: The output-distribution of (a) a well chosen first-stage decision, and (b) a bad first-stage decision. Taken from a facility location problem where the objective is to minimize the expected cost. Evaluated for 60 000 sampled outcomes.

2.1 Problem-based scenario generation

Let us consider the two-stage decision problem with expected value as a utility metric. This is expressed by the objective[1]

$$f_P(x) = \mathbb{E}_P[f(x)(\xi)] = \int_\Xi f(x)(\xi)P(d\xi) \tag{2.1}$$

where x is the first-stage decision constrained to some feasible set X, and P is the probability measure of the underlying probability space $(\Omega, \mathcal{F}, P)$. For the problems in our context, we can decompose the objective into a deterministic cost $g(x)$ and a stochastic cost $Q(x)(\xi)$ known as the *recourse function*. The recourse function is interpreted as the future cost of a decision in a specific outcome, and the advantage of modelling with stochastic programs is that we take this cost directly into account. Thus, the output-distribution is

$$f(x)(\xi) = g(x) + Q(x)(\xi). \tag{2.2}$$

Evaluating $Q(x)(\xi)$ typically involves solving an optimization problem in itself and the integral $Q_P(x) = \mathbb{E}_P[Q(x)(\xi)]$, referred to as the *expected recourse*, cannot be evaluated if the distribution is continuous or has too many outcomes. We rely on discretizing or simplifying the distribution function by redefining its probability measure P, resulting in a set of discrete outcomes ξ_s with associated probabilities p_s defined over a finite index set $s \in \mathcal{S}$. The collection of outcomes together with its probabilities makes a discrete distribution and is referred to as a *scenario set*. The scenario set is defined by its probability measure $\mathcal{T}$. Thus, we make the approximation

$$f_P(x) \approx f_{\mathcal{T}}(x), \tag{2.3}$$

and solve $\text{argmin}_{x\in X} f_{\mathcal{T}}(x)$ instead of $\text{argmin}_{x\in X} f_P(x)$ to get the solution to the decision problem.

The scenario set's size is closely related to both the stability and the solution time of a stochastic program. The approximation error typically shrinks with more scenarios, but this also involves more evaluations of $Q(x)(\xi)$ which can be computationally very time-consuming. The procedure of creating the scenario set is called *scenario generation*. The challenge with scenario generation

[1]The notation $f(x)(\xi)$ means that $f(x)(\cdot) : \Xi \to \mathbb{R}$ is a function which takes the argument ξ, but the function $f(x)(\cdot)$ also changes with different first-stage decisions x. Introduction of new notation was done to avoid unnecessary confusion when notation from mathematical and empirical stability theory was combined.

lies in making appropriate estimates of $\mathbb{E}_P[Q(x)(\xi)]$ which also holds for a relatively large set of relevant x.

Let x be fixed and consider the stochastic variable $R_x(\omega) = f(x)(\xi(\omega))$ which we refer to as the *output-distribution* considering different first-stage decisions x, and $\xi(\omega)$ is referred to as the *input-distribution*. The input-distribution can take familiar forms like a multivariate normal distribution or a multivariate Bernoulli distribution, while the output-distribution can have peculiar and unfamiliar forms based on the nature of the problem.

Now, changing the distribution to a scenario set $\mathcal{T}$ will alter the properties of both the input-distribution and the output-distribution. Problem-based scenario generation is at its core about exploiting properties of $Q(x)(\xi)$ to estimate $\mathbb{E}_P[R_x(\omega)]$ as accurately as possible by the scenario set while still keeping the size of the scenario set as small as possible. Note that generating a scenario set which more accurately approximates the input distribution often causes more accurate output-distributions as well. However, it's not the case that the best possible concise description of the input distribution results in the best possible concise representation of the output-distribution.

The stability of stochastic problems with perturbations of its underlying distribution is a crucial tool to determine how to appropriately compensate the approximations of the input-distribution according to a specific problem's characteristics.

2.2 Stability

The reliability of the solution from a stochastic program is the reason scenario generation matters. If a perturbed scenario set results in significantly different results, then the generation procedure is unreliable, and we cannot trust that the result of the solution procedure is the solution to the model we formulated. In practice, we perturb a scenario set by adding slightly more or slightly fewer scenarios or, if the generation procedure is non-deterministic, generate the scenario set multiple times.

It has been observed repeatedly in practice that solutions to stochastic programs obtained by using reasonable approximations of the distribution are robust to reasonable perturbations of that distribution (Römisch & Wets, 2007b). The Fortet-Mourier probability semi-metric (2.4) is one theoretical tool used to explore this observation.

Let $\mathcal{P}(\Xi)$ be the set of all Borel probability measures on Ξ. The epi-distance of the objective function $f(x)(\xi)$ between the probability measures $P, Q \in \mathcal{P}(\Xi)$ can then be bounded from above by the probability semi-metric

$$d_{\mathcal{F}_\rho}(P,Q) = \sup\left\{\left|\int_\Xi f(x)(\xi)P(d\xi) - \int_\Xi f(x)(\xi)Q(d\xi)\right| \ : \ f(x) \in \mathcal{F}_\rho\right\} \tag{2.4}$$

where $\mathcal{F}_\rho = \{f(x)(\cdot) : \Xi \to \mathbb{R} \quad \text{s.t.} \quad x \in X \cap \rho\mathbb{B}\}$ is a class of measurable functions from Ξ to $\mathbb{R}$. The set $\mathcal{F}_\rho$ is interpreted as all possible output-distributions for each feasible first-stage decision $x \in X$ also within the ball $\rho\mathbb{B}$. The ball which is centred in the origin could potentially be relatively large.[2]

Thus $d_{\mathcal{F}}(Q,P)$ quantifies the absolute difference between the objective value of the optimization problem for two distributions with probability measures P, Q. The supremum means that the bound is evaluated at the first-stage decision that makes for the largest difference between the objective values $f_P(x), f_Q(x)$.

Römisch and Wets (2007b) show formally that stochastic programs are Lipschitz continuous with respect to the Fortet-Mourier metric for reasonable perturbations of the distribution, which

[2]We need ρ to be sufficiently large so that the solution set $S(P) = \operatorname{argmin}_{x\in X}\{f_P(x)\}$ is contained in the ball $S(P) \subset \rho\mathbb{B}$ and that the optimal objective value $v(P) = \min_{x\in X}\{f_P(x)\}$ fulfils $v(P) \geq -\rho$, based on the perturbation theory in (Rockafellar & Wets, 2009, Section 7J).

supports the empirical observation that stochastic programs are reasonably robust under perturbations of the underlying distribution.

There are three kinds of distributions to be considered in this context. First, there is the 'real' distribution, meaning the underlying unobtainable distribution for the physical phenomenon we are observing. Second, there is the observed distribution which is our most accurate description of the physical phenomenon, expressed by an analytical or empirical distribution function. Lastly, there is the scenario set used to solve the stochastic program, which is a simplification and approximation of the observed distribution.

Römisch and Wets (2007b) argue that their result means using the observed distribution instead of the real distribution will still provide reliable results. This result can also be applied to explain why approximating the observed distribution by a scenario set often also gives reliable results. However, at such a significant approximation the reliability can no longer be taken for granted.

From a different point of view in the scenario generation literature, Kaut and Wallace (2007) points out that a scenario generation method should be evaluated by its performance in practical problems rather than on its currently provable theoretical properties as there is an evident gap between them. As already pointed out, the reliability of stochastic programs is intuitively present in many problems. However, as the problems we want to solve grow larger, and we start pushing the size of the scenario sets to the bare minimum, we need criteria to evaluate if this inherent reliability is still sufficient. Kaut and Wallace (2007) introduces some practical evaluation criteria to determine if the solution from a given model is stable. The evaluation criteria are based on a scenario generation procedure which generates a collection of scenario sets $\mathcal{T}_1,\ \mathcal{T}_2, \dots$ which are perturbations with respect to each other. The scenario sets are then used to find corresponding solutions $x_1^*,\ x_2^*, \dots$ of the stochastic programming model. Finally, these solutions can be evaluated by the following criteria:

- *In-sample stability* determines if the solutions are stable when evaluated by the scenario set it was solved by. The condition for in-sample stability is

$$f_{\mathcal{T}_i}(x_i^*) \approx f_{\mathcal{T}_j}(x_j^*), \quad \forall i, j. \tag{2.5}$$

- *Out-of-sample stability* determines if the solutions are stable when evaluated by the whole observed distribution. The condition is

$$f_P(x_i^*) \approx f_P(x_j^*), \quad \forall i, j. \tag{2.6}$$

- *Bias* indicates if the generated scenario set accurately represents the expected value of the observed distribution, and is evaluated by

$$f_{\mathcal{T}_i}(x_i^*) \approx f_P(x_i^*), \quad \forall i. \tag{2.7}$$

Note that there is a distinction between evaluating the objective function in-sample and out-of-sample and the use of these evaluations for the different stability properties (2.5)–(2.7).

King and Wallace (2012) explains that if in-sample stability is not present, you may not have understood your problem properly. It need not be present for good scenario generation procedures, but it means that something is going on in the model which is odd and you may not know why it still gives good results or whether it will remain stable for similar problem instances.

Out-of-sample stability tells how the solution performs in the actual criteria we want to solve for. Out-of-sample stability is the only criteria we need to fulfil since this is the evaluation of the 'cost in reality'. The problem with using out-of-sample evaluation is that is can be very computationally intensive to determine, however, not intractable.

If a scenario generation procedure does not attain out-of-sample stability, but preserves in-sample stability, then the scenario generation creates a stability which is not really there. This is bad. The other way around, if out-of-sample stability is present, but not in-sample stability, then the evaluation in-sample does not give you much information but can still cause the solution algorithm to produce reasonable solutions when evaluated out-of-sample.

King and Wallace (2012) explains that the relation between in-sample and out-of-sample stability is not simple at all, but Prochazka and Wallace (2020) provides some additional important properties which should be fulfilled by a good scenario generation procedure:

- *Appropriate ordering of solutions* x, y by the relation

$$f_P(x) < f_P(y) \implies f_{\mathcal{T}}(x) < f_{\mathcal{T}}(y). \tag{2.8}$$

- *Avoidance of overconfident outliers*, where if the relation (2.8) is false then we require that

$$f_{\mathcal{T}}(x) \approx f_P(x), \tag{2.9}$$

  meaning that the significance of a violation of ordering is not too large.

Property (2.8) ensures that an optimization procedure which aims to find $\text{argmin}_{x \in X}\{f_{\mathcal{T}}(x)\}$ would converge to a solution within the actual solution set $S(P) = \text{argmin}_{x \in X}\{f_P(x)\}$. If this property is ensured for a scenario generation procedure, we would be done.

Property (2.9) is a weaker claim than property (2.8). If the implication (2.8) is false we may require property (2.9) instead. This ensures that a violation of (2.8) would not be too severe in terms of how wrong the solution using $\mathcal{T}$ would be.

Prochazka and Wallace (2020) also suggests that these properties are more important for better first-stage decisions because an optimization procedure will converge towards better solutions. If the representation of the output-distribution of a terrible first-stage solution is off, it may not matter because the search algorithm disregards it to move towards much better solutions. However, as the algorithm converges to an optimal solution, the difference between compared solutions is smaller, and it's more important that the evaluations are most accurate there.

2.3 Discussion

It cannot be emphasised enough that the first-stage decision is the only thing we aim to solve for and it's evaluated only by its out-of-sample objective value. All other quantities or decisions variables in the problem are tools to support finding the best first-stage decision. The second-stage evaluation and decision vectors are simply a means to evaluate how the system we model responds to a different first-stage decision.

Now, we argue that an output-distribution centric view of stochastic problems is essential because the decision problem under uncertainty is all about hedging the first-stage decision against some set of possible outcomes. Collapsing the output-distribution by evaluating in its utility metric will rid us of much information about the problem-specific characteristics of the uncertainty. If we can analyse the output-distribution to find clusters of outcomes where changes in the first-stage decision manifest themselves in approximately the same way, these can be re-represented as a single scenario with minimum loss of accuracy in the final result. The goal of scenario generation is indeed only to find out-of-sample outcomes which similarly manifest themselves in the output-distribution.

Utilising the fact that it is only the final first-stage decision that matters provides much freedom when formulating the scenario set. This is because scenario generation becomes an argument about where the solution procedure converges to, not how accurately the distribution is represented. Scenario generation based on problem insight and simple heuristics can often be helpful; if we can check that they work. That is why the tools of empirically testing stability are so essential for scenario generation. They allow us to validate the effectiveness and reliability of a scenario set, giving us the freedom to depart from statistically sound approximation and to explore how problem-specific corrections can provide better representations of uncertainty.

Dupačová, Gröwe-Kuska and Römisch (2003) suggested that the Fortet-Mourier metric could be used as a canonical metric to assist scenario generation. Note that the Fortet-Mourier metric itself cannot be minimised but attempts to bound it and minimise the bound have been done in (Dupačová et al., 2003), see details in the literature review.

Even if we could solve for the Fortet-Mourier metric, this thesis argues there is a gap between a most effective scenario set, meaning reliable and with minimum cardinality, and the one obtained from minimising the Fortet-Mourier metric. The discrepancy lies in the supremum over possible output-distributions in (2.4). It may be too conservative evaluating the deviation of the utility metric at the first-stage decision where it is largest, and a too conservative metric of stability could result in making unnecessarily large scenario sets. This thesis aims to explore how scenario generation can be improved to solve otherwise intractable problems by generating more concise scenario sets.

As pointed out by Prochazka and Wallace (2020), it is at the better first-stage decisions that the scenario approximation matters the most. In this thesis, we attempt to evaluate the stability of stochastic programs empirically at more appropriate first-stage decisions than is accounted for by the Fortet-Mourier metric but are also motivated by the deviation within the set of output-distributions of these first-stage decisions, analogous to the Fortet-Mourier metric, but less conservatively.

Comparing the mathematical theory on the stability of stochastic programs and empirical testing has not been done in the literature before, and this thesis builds on finding the synergies between these to develop better approximations and tools for analysing the stability of stochastic problems.

3 Literature review

This review highlights previous work on various methods of scenario generation in the literature on stochastic programming which has served as inspiration for this thesis. There are three such sources of inspiration; stability across first-stage decisions, scenario generation with binary distributions, and clustering methods for scenario generation. Finally, problem-based scenario generation, which was the onset for this thesis, is briefly reviewed by an overview of previous advances.

3.1 Scenario generation by sample first-stage decisions

Prochazka and Wallace (2020) considered appropriate relations between in-sample and out-of-sample stability and demonstrated them by implementing a heuristic to fit a scenario set to adhere to these properties. They constructed an appropriate objective function for the fitting algorithm, as a function of the in-sample and out-of-sample expected objective values, based on their postulates of what properties a good scenario tree should exhibit. What's especially interesting about their contribution is that they consider stability across different first-stage decisions. These were obtained by solving the stochastic program by a heuristic procedure multiple times.

Their approach showed promising results, lowering set sizes from 50 sampled to 3 fitted scenarios, but it had some caveats based on the development time of various heuristics tailored for the specific problem to be solved. Prochazka and Wallace explicitly states that due to this overhead their procedure is only appropriate for applications where online solution times need to be reduced or where scenario generation is especially challenging.

The fitting algorithm in (Prochazka & Wallace, 2020) required many first-stage solutions, and afterwards, they had to be evaluated out-of-sample. Many out-of-sample evaluations may be reasonable on many occasions, but if the second-stage is especially time-consuming to solve it may take an unreasonable amount of time. For reference, the Air Traffic Flow Management problem considered in this thesis which has a MIP second-stage can take up to 320–1280 CPU hours to evaluate out-of-sample for a single first-stage decision. One aim of this thesis is therefore to reduce the required number of such first-stage solutions.

The contribution of this thesis builds on (Prochazka & Wallace, 2020) by also considering stability across different first-stage decisions. What makes it possible to use fewer first-stage decisions is that the whole output-distribution is used instead of only the expected recourse. In a way, this is using the same available information which must be obtained in both procedures but exploiting it better.

We also generate a set of solutions but instead of perturbing the solutions by the solution trajectories, i.e. letting the heuristic make for their distinctiveness, they are perturbed in the scenarios used to find them. This way, the perturbations in the resulting output-distributions has a relation to the stability of the model itself, not necessarily on the solution procedure to obtain them.

3.2 Binary distributions

The literature on scenario generation for binary distributions is very sparse. The only article found which addresses scenario generation in particular for these problems is (Prochazka & Wallace, 2018), although there exist application papers with scenario generation for binary distributions as part of the work. In was mentioned that the approach in (Prochazka & Wallace, 2020) may also be effective. Otherwise, scenario reduction (see Section 3.4) is a viable distribution-based approach.

The Air Traffic Flow Management problem in (Wang & Jacquillat, 2020), which we investigate closer in this thesis, is specified by a binary distribution to represent different operating modes at airports. Additionally, Ball, Colbourn and Provan (1995) discuss network reliability on a class of problems characterized by how failures affect the system. This problem class is naturally formulated as stochastic programs with binary distributions. They mention applications in communication, transportation, power networks and command and control systems.

Prochazka and Wallace (2018) looked at binary distributions specifically for problems where penalty costs may or may not appear as a result of the stochastic outcome. These are analogous to problems with tail risk measures, for which the continuous counterpart was considered in (Fairbrother, Turner & Wallace, 2018, 2019).

Prochazka and Wallace argue that for such problems, only some binary outcomes cause penalties for a given first-stage decision. They identified an ordering-relation between outcomes which sorts them by whether an outcome can cause a penalty or not, given that we know the penalty for one of the outcomes. This information is used to ease the out-of-sample evaluation significantly. The second-stage cost is then computed recursively, starting at the worst scenario, here defined by only 1s, and lowering the outcome variables to 0 one at a time. This recursive algorithm can be represented by a tree where the root node is the worst outcome, and each child node has one less 1 than the parent node. This tree can be searched recursively, and a node which doesn't cause a penalty informs us that every child node can be ignored.

There exist a limited number of problems with this structure; therefore, this thesis considers more general classes of problems. We use arguments on stability and relations among output-values instead of ordering relations between binary outcomes; thus, it holds for any problem with a binary distribution. Still, ordering relations by (Prochazka & Wallace, 2018) give exact answers, and still maybe very large outcome sets to evaluate over, while the stability approach is only approximate.

3.3 Clustering methods for scenario generation

Clustering is a well-known method for systematizing large sets of data points into clusters such that all elements within a cluster exhibits similar properties and at the same time exhibit different properties between clusters. Centroid-based clustering methods, in particular, select a centre-point and attempt to minimize the distance between the centre-point and all other points in the cluster. K-means is one such method where the centre-point is the average point in the cluster. The k-median method, on the other hand, selects one of the dataset points as a centre-point. The k-means problem can be solved approximately by an efficient greedy approximation, while k-means is a combinatorial NP-hard problem which for large datasets must be approximated, i.e. by forward and backward propagation (Heitsch & Römisch, 2003).

A scenario generation method based on clustering has three steps (Sun, Teng, Konstantelos & Strbac, 2018):

- Selection of clustering variables and distance metric
- Selection of clustering technique
- Selection of a representative scenario from each cluster

We apply this framework for the rest of this thesis.

Sun et al. (2018) tested a large variety of parameters for all of these steps on a Transmission Network Expansion Planning (TNEP) problem. Among the considered clustering methods where hierarchical-, centroid- and distribution-based clustering, and for the clustering variables they used the input-distribution, objectives from solving the second stage for a given first-stage decision and a combination of the two. Both the L_1 and L_2 distance metrics were tested. In both (Feng & Ryan, 2016; Sun et al., 2018) variations of qualitatively distinguishing scenarios based on problem characteristics were used to make the scenarios more effective or enhance the computational time to solve the clustering problem.

It can be concluded from the work that among the various tested parameters in (Sun et al., 2018), centroid-based clustering was effective in many contexts, while the effectiveness of the scenarios was more sensitive on the choice of clustering variables combined with how the qualitative distinctions from the problem were made. See (Sun et al., 2018, Section 5.3).

The approaches in both (Feng & Ryan, 2016; Sun et al., 2018) was entirely or partly based on clustering by problem-based clustering variables, meaning that in different ways, they solved the problem and evaluated the second-stage score for each out-of-sample scenario. A downside of the approaches in these papers is however that the given first-stage solution was held constant, meaning that they did not capture how the characteristics of the output-distribution may change under changes in the first-stage decisions. Additionally, Feng and Ryan (2016) based the clustering variables on the solution of the deterministic formulation of the problem characterized by the expected value scenario or, as also done in (Sun et al., 2018), they solved the problem for each individual scenario, known as the solutions of perfect information, and based the clustering on the stochastic costs of those solutions. Both of these approaches are lacking in that the objective values they cluster on are based on solving the problem for only a single scenario at a time. This results in *structurally different* first-stage solutions from what is obtained by solving with a set of scenarios (Wallace, 2010), and they are therefore working within the realm of what-if analysis. See (King & Wallace, 2012, Section 1.3) for a detailed discussion on this fallacy.

A further downside of these approaches is that the methods are heavily tailored to the specific problem, meaning that applying the approaches to other problems may involve extensive testing to assure they work. That said, both the scenario generation approaches in (Sun et al., 2018) and (Feng & Ryan, 2016) proved very effective on their specific problems.

Similarly to these two approaches, we use variations of clustering variables both from the input and output domains and try two centroid-based clustering approaches; k-means and k-median. The distinction, however, is the use of stability arguments for variations in first-stage decisions which wasn't done in either of the mentioned approaches, and the use of more realistic scenario sets to obtain first-stage decisions.

Variations of clustering methods for scenario generation are present in the literature where some distribution-based alternatives include (Chen & Yan, 2018; Latorre, Cerisola & Ramos, 2007) and scenario reduction, which we discuss in more detail in the next section.

3.4 Scenario reduction

Scenario reduction is a clustering-based scenario generation method which is also motivated by stability arguments of stochastic programs. The paper (Dupačová et al., 2003) is the original paper for this approach. Scenario reduction has been used extensively in the literature on stochastic programming, but experience has also shown that it may not be appropriate for all kinds of problems (Sun et al., 2018) and comparing it with other approaches shows that it may result in significant bias in the in-sample evaluation (Löhndorf, 2016).

The starting point of Dupačová et al. (2003) is that they suggest the Fortet-Mourier type metric (2.4) be used as a canonical metric for stability in stochastic programs. Furthermore, they show that for stochastic programs, the Kantorovich functional is a valid upper bound for the Fortet-Mourier metric. The Kantorovich functional, also called the Monge-Kantorovich mass transportation problem, takes the form

$$\mu_c(P,\ Q) = \inf_{\eta} \left\{ \int_{\Omega\times\Omega} c(\omega,\ \tilde{\omega})\eta(d\omega, d\tilde{\omega}) \quad \text{s.t.} \quad \begin{array}{l} \eta \in \mathcal{P}(\Omega\times\Omega), \\ \eta(B\times\Omega) = P(B), \quad \forall B \in \mathcal{B} \\ \eta(\Omega\times B) = Q(B), \end{array} \right\}. \tag{3.1}$$

where $P, Q \in \mathcal{P}(\Xi)$ are two probability measures (i.e. distributions), $c(\omega, \tilde{\omega})$ is a transportation cost between ω and $\tilde{\omega}$, and η is the transportation plan. The minimum transportation plan η^* tells us how the distribution P can be transported to Q with the minimum amount of 'effort' determined by the cost function $c(\cdot,\cdot)$. The Monge-Kantorovich mass transportation problem is well studied and have been used in various applications, see (Rachev & Rüschendorf, 1998, 2006).

Considering P, Q as discrete distributions with respective cardinality N, M, the Kantorovich functional simplifies to the linear primal-dual representation

$$\hat{\mu}_c(P,\ Q) = \min \left\{ \sum_{i=1}^{N}\sum_{j=1}^{M} c(\omega_i,\ \tilde{\omega}_j)\eta_{ij} \quad \text{s.t.} \quad \eta_{ij} \geq 0,\ \sum_{i=1}^{N}\eta_{ij} = q_j,\ \sum_{j=1}^{M}\eta_{ij} = p_i, \quad \forall i,j \right\} \tag{3.2}$$

$$= \max \left\{ \sum_{i=1}^{N} p_i u_i + \sum_{j=1}^{M} q_j v_j \quad \text{s.t.} \quad u_i + v_j \leq c(\omega_i,\ \tilde{\omega}_j), \quad \forall i,j \right\} \tag{3.3}$$

Thus, the functional $\hat{\mu}_c(P,Q)$ can be used to evaluate distances between specific discrete distributions P, Q by a linear expression. The expression (3.3) is also referred to as the Kantorovich-Rubinstein distance or the 1-Wasserstein distance in the literature.[1]

In scenario reduction, the Kantorovich functional guides how to remove scenarios from a large discrete distribution, representing the historical data, to obtain a scenario set of a given cardinality. The resulting scenario set minimizes the Kantorovich functional, and to solve this for discrete distributions translates directly to a k-median clustering problem (Heitsch & Römisch, 2007). For reference, the Kantorovich-Rubinstein distance is equivalent to using what's called centroid-based distances in the context of clustering.

The advantage of using the Kantorovich distance is that we can solve the minimum transportation problem by an integer linear program for reasonably sized historical data. For larger discrete distributions, which may often be the case, approximations are needed (Heitsch & Römisch, 2003).

The disadvantage, however, is that when the bound is made on the Fortet-Mourier metric, all problem-based notions are lost, and scenario reduction is therefore distribution-based. In other words, the Kantorovich bound may be too loose on the Fortet-Mourier metric to be useful in all settings. Scenario reduction may, therefore, be more lacking in problems where the scenarios are very dependent on the problem itself.

Henrion and Römisch (2018) expanded on the scenario reduction theory to perform scenario reduction with respect to a better, problem-based metric. However, the resulting formulation is a generalized semi-infinite program (GSIP) which relies on evaluations of upper and lower bounds of the recourse function for different first-stage decisions. Generalized semi-infinite programs have an infinite amount of constraints whose index set depends on the decision, and methods to solve them is an active area of research, see explanation in (Henrion & Römisch, 2018, Section 2). Bounds on recourse functions exist but do not scale very well with the dimensions of the input-distribution and can only be reasonably solved up to about ten dimensions (Kall & Wallace, 1994, Section 3.4). Thus, to the author, it seems like this formulation scales poorly for large-scale problems, and numerical experiments to demonstrate the approach has not yet been conducted.

[1]see bibliographical note in (Villani, 2008, Chapter 6)

It may not be possible to formulate a more appropriate metric for large-scale problems which we can solve the scenario generation problem with respect to, but we may be able to get closer by empirical stability arguments. Empirical stability arguments are what distinguishes this thesis from the literature on scenario reduction and use of the Fortet-Mourier metric for scenario generation.

3.5 Problem-based scenario generation

The current literature on problem-based scenario generation is classified into three categories:

- Filtering outcomes
- Problem class insight
- Stability arguments

Filtering of outcomes means that some of the outcomes in the input-distribution have very low or no impact on the objective function and can, therefore, be aggregated or ignored. Approaches under this category include the use of ordering relations in (Prochazka & Wallace, 2018) and the scenario generation procedures for tail-risk measures in (Fairbrother et al., 2018, 2019).

Problem class insights are the cases when application-specific knowledge and expert intuition can be used to guide the scenario generation procedure. Typically, properties of the output-distribution can be deducted a priori or experience with solving the problem many times has given intuitions which can guide scenario generation. In (Feng & Ryan, 2016; Sun et al., 2018) problem-specific distinctions of the solutions of the stochastic model was used to guide scenario generation, Guo, Wallace and Kaut (2019) used only local dependence structures for travel times on routes because dependence across large distances would have small effects on the solution, and Zhao and Wallace (2016) made scenarios 'by hand' based on direct insight into the problem. This category of problem-based scenario generation may also be the most common occurring in various application papers.

Stability arguments use properties of how the mathematical formulation of the model reacts to perturbations of the scenario set used to solve it, both theoretically and empirically. Stability of stochastic programs was discussed in detail in Chapter 2 and the thesis' contribution falls under this category of problem-based scenario generation. In (Prochazka & Wallace, 2020) empirical stability arguments were used to heuristically fit scenario sets based on a large collection of approximate first-stage decisions. The use of the Fortet-Mourier metric is problem-based, but when used for scenario reduction in (Dupačová et al., 2003) the minimized bound is too loose on the Fortet-Mourier metric to explicitly consider the problem in itself. Henrion and Römisch (2018) attempted to correct for that.

4 Integrated Model of Scheduling and Operations in Airport Networks

Air traffic systems have for the last few years seen increased demands while the capacity has remained limited. This imbalance causes significant congestions and delays at airports, resulting in high costs. Wang and Jacquillat (2020) proposed a new stochastic programming model for air traffic scheduling named *Integrated Model of Scheduling and Operations in Airport Networks* (IMSOAN) to address this problem.

Their model augments prior approaches by implementing both scale integration and scope integration. *Scale integration* means that a whole network of airports is considered within one model. *Scope integration* means that interdependencies between strategic scheduling decisions and tactical decisions on delays are considered within the model. Scope integration is important due to uncertainty in weather conditions where the effects of the strategic decision on the tactical level cannot be known until the uncertainty is resolved. These considerations result in a model of significant size. For the largest network considered of 30 airports and including the second stages with 30 scenarios, the model yields 8.2 million variables (7.3 million binary) and 20.8 million constraints. For three scenarios, the size would be 0.9 million variables (0.8 million binary) and 2.4 million constraints. The work in (Wang & Jacquillat, 2020) mainly consisted of developing more efficient decomposition techniques to be able to solve the model.

Previous literature on air traffic flow management is rich in models for the tactical level, while the scope and scale integration have previously not been incorporated simultaneously. Note that it is only the strategic level planning IMSOAN improves on compared to previous literature. There exist better models for tactical decisions, but they are also more complicated. The model for tactical decisions in IMSOAN is only a simplification to achieve tractability of the integrated model.

The model shows significant improvements in scheduling. Wang and Jacquillat (2020) showed that as little as 1% change in schedules on the strategic level could have up to 30% reduction in delays by using IMSOAN. See (Wang & Jacquillat, 2020, Section 7) for a more in-depth discussion on the benefits of integration and the resulting spatial and temporal patterns resulting from the model.

In this chapter, we will expand on the work in (Wang & Jacquillat, 2020) by considering scenario generation more deeply for this problem. More specifically, the model has a binary input distribution and is large-scale. Binary distributions offer their own set of challenges while the large scale means that solution procedures would benefit significantly from a more efficient scenario representation.

4.1 Model description

IMSOAN is a two-stage stochastic programming model where both stages are Mixed-Integer Problems (MIPs). The first stage considers scheduling of air traffic routes between airports based on preferred departure and arrival times submitted by airlines, and these schedules must be determined four months ahead of time. Due to the large discrepancy between the time of scheduling and the time of the actual flights, bad weather on the day may cause additional unforeseen congestion at the airports. The second stage, therefore, considers the outcome of the weather and the chosen first-stage schedule to reschedule flights. The first-stage displacements can be both forwards and backwards in time, which also holds for second-stage arrivals, but second-stage departures may only be delayed.

The objective in each stage is to minimize the number of displacements and the amount of rescheduling, respectively. Displacement of preferred schedules ahead of time and the amount of rescheduling on the day does not compare one to one, and the problem is therefore bi-objective. This is addressed by a linear weighting parameter ρ between the stages' objectives.

The model considers a single day of operations, which is reasonable as flight patterns across different days are very similar. Furthermore, a day is partitioned into 15 minute periods within the time-span 6:00 to 24:00 (18 hours), which is when most flights are scheduled. There are 72 time periods during a day, and each departure and arrival is assigned to one such time period. It is assumed that en-route times and connection times are quantified by an integer number of such time periods.

The weather has two possible outcomes at a given airport, determined by two different operating modes. There are visual meteorological conditions (VMC) where pilots can separate terrain and other air-crafts by visual means, and instrumental meteorological conditions (IMC) where pilots rely on instruments. The distinction between the two is based on weather conditions. VMC is considered 'good weather' and IMC as 'bad weather' where each of them determines the capacity of departures and arrivals at a given airport. Thus, the stochastics in the problem are summarized by weather outcomes at each airport for each time period during a day.

The scenarios are based on historical data for five years, which results in 1826 data points. We assume that each of the historical outcomes has the same probability of occurring.

Lastly, schedules may be required to satisfy minimum connection times due to maintenance on the aircraft before its next trip.

4.2 Mathematical formulation

The mathematical model should determine departure and arrival times for all flights in both stages. The problem is combinatorial in nature, and we need to formulate it by using variables and constraints in a slightly non-intuitive way to get a concise and efficient representation. The representation is adapted from (Bertsimas, Lulli & Odoni, 2011; Bertsimas & Patterson, 1998).

We use a binary decision tensor with index i over the set of flights and index t over the set of time periods, one for arrivals and one for departures. Constraints are used in the model to ensure that the tensor is non-increasing in its time indices. Let y denote the decision tensor for either the first stage or the second stage. The interpretation of a given value is then

$$y_{it}^{\text{dep/arr}} = \begin{cases} 0, & \text{flight } i \text{ departs/arrives before time period t} \\ 1, & \text{flight } i \text{ departs/arrives in time period t or later} \end{cases}. \tag{4.1}$$

The last time index of value 1 is thus the time of departure/arrival for flight i. Figure 4.1 shows an illustration of the representation. The use of this representation is explained in more detail in the following paragraphs. A reference table for all variables and parameters in the model is given in Table 4.1.

Time index:	1	2	3	4	5	6	7	8	9	10
Binary value:	1	1	1	1	1	0	0	0	0	0

Figure 4.1: Modelling structure of departure and arrival time with binary variables. The departure/arrival time is in period 5 since 5 is the last time index of binary value 1.

The binary variables $w_{it}^{\text{dep/arr}}$ are the first stage decisions and $x_{it}^{\text{dep/arr}}$ the second-stage decisions. The en-route time for a flight i is within the interval $[\Delta_i^{\min},\ \Delta_i^{\max}]$ with scheduled en-route time Δ_i^{sch} defined as the difference between the scheduled departure and arrival, $\Delta_i^{\text{sch}} = S_i^{\text{arr}} - S_i^{\text{dep}}$ with the property that $\Delta_i^{\min} \leq \Delta_i^{\text{sch}} \leq \Delta_i^{\max}$. For the first-stage, there is a specified maximum displacement δ, while in the second stage there is a maximum delay $l_i^{\text{dep/arr}}$.

By restricting the number of displacements and delays, the size of the feasible set in both stages is significantly reduced as only a limited number of time periods need to be considered for each flight. For the first-stage decision variables, we need only consider the time index sets

$$\tilde{\mathcal{T}}_i^{\text{dep}} = \left\{ S_i^{\text{dep}} - \delta + 1,\ \dots\ ,\ S_i^{\text{dep}} + \delta \right\} \tag{4.2}$$

$$\tilde{\mathcal{T}}_i^{\text{arr}} = \{ S_i^{\text{arr}} - \delta + 1,\ \dots\ ,\ S_i^{\text{arr}} + \delta \} \tag{4.3}$$

while for the second-stage decision variables, we need only consider the time index sets

$$\mathcal{T}_i^{\text{dep}} = \left\{ S_i^{\text{dep}} - \delta + 1,\ \dots\ ,\ S_i^{\text{dep}} + \delta + l_i^{\text{dep}} \right\} \tag{4.4}$$

$$\mathcal{T}_i^{\text{arr}} = \left\{ S_i^{\text{dep}} - \delta + \Delta_i^{\min} + 1,\ \dots\ ,\ S_i^{\text{arr}} + \delta + l_i^{\text{arr}} \right\}. \tag{4.5}$$

Additionally, we use the convention that

$$w_{it} = \begin{cases} 1, & \text{if } t \prec \tilde{\mathcal{T}}_i \\ 0, & \text{if } t \succ \tilde{\mathcal{T}}_i \end{cases} \tag{4.6}$$

and

$$x_{it} = \begin{cases} 1, & \text{if } t \prec \mathcal{T}_i \\ 0, & \text{if } t \succ \mathcal{T}_i \end{cases} \tag{4.7}$$

where the notation $t \prec \mathcal{T}$ means that t is less than every element in $\mathcal{T}$, and correspondingly larger for $t \succ \mathcal{T}$. This ensures consistency for constraints defined over all time indices. Note that (4.2)–(4.3) also enforces the maximum displacement constraint for the first stage by not enabling displacements beyond δ.

Using this convention, we can express various quantities for the problem by linear expressions. Let y denote either of the decision tensors w, x. Then the en-route time for flight i is

$$\sum_{t \in \mathcal{T}} y_{it}^{\text{arr}} - y_{it}^{\text{dep}}, \tag{4.8}$$

and by switching departure and arrival variables, we get the on-ground time

$$\sum_{t \in \mathcal{T}} y_{it}^{\text{dep}} - y_{it}^{\text{arr}}. \tag{4.9}$$

The one-hot vector

$$(y_{it}^{\text{dep/arr}} - y_{i,t+1}^{\text{dep/arr}}), \quad \forall t \tag{4.10}$$

determines the time of departure/arrival for flight i, thus the number of departure/arrival events at airport k in time period t is

$$\sum_{i \in \mathcal{F}_k^{\text{dep/arr}}} y_{it}^{\text{dep/arr}} - y_{i,t+1}^{\text{dep/arr}}. \tag{4.11}$$

Finally, the number of rescheduling displacements for flight i in the second stage is determined by

$$\sum_{t \in \mathcal{T}} x_{it}^{\text{dep/arr}} - w_{it}^{\text{dep/arr}}. \tag{4.12}$$

A scenario outcome is represented by a binary vector ϕ_{kt} where k is the airport and t is the time of day. A function $Q_{kq}(\cdot)$ determines the capacity for a given airport and capacity type as a function of the weather conditions. There are three kinds capacity constraints; departures, arrivals and total arrivals and departures. The total capacity is less than the sum of departure and arrival capacities, thereby constructing a capacity envelope as illustrated in Figure 4.2. This is because different waiting times apply between consecutive departures or arrivals than for alterations between them.

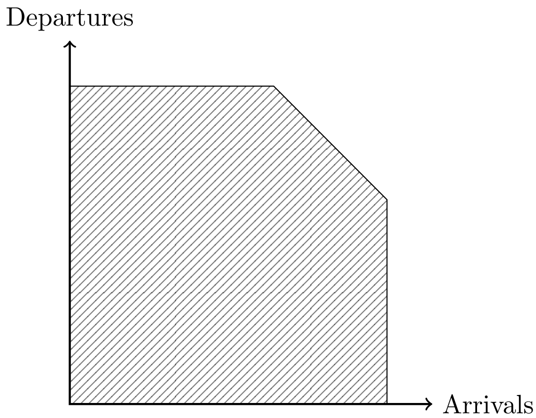


Figure 4.2: Capacity envelope at an airport.

Table 4.1: Overview of all variables and parameters.

General parameters	
$\mathcal{T}$	set of all time periods
$\mathcal{F}$	set of all flights
$\mathcal{K}$	set of all airports
$\mathcal{F}_k^{\text{dep/arr}}$	set of all flights departing/arriving at airport k
$S_i^{\text{dep/arr}}$	scheduled departure/arrival time for flight i (as indicated by the airline)
$\Delta_i^{\text{min/sch/max}}$	the minimum/scheduled/maximum en-route time for flight i
ρ	weighting parameter between the first- and second-stage objectives such that $\rho \in [0, 1]$ where lower values of ρ puts more emphasis on the second stage
$\mathcal{C}$	subset of flights $(i, j) \in \mathcal{F} \times \mathcal{F}$ with an aircraft connection
τ_{ij}	minimum connecting time between flights $(i, j) \in \mathcal{C}$

First-stage	
$w_{it}^{\text{dep/arr}}$	binary decision variable determining if flight i will depart/arrive at time t or later
δ	maximum displacement of a flight in the first stage
g_{it}	cost of displacement when flight i is scheduled to depart at time t
$\tilde{\mathcal{T}}_i^{\text{dep/arr}}$	set of possible departure/arrival times for flight i in the first stage

Second-stage	
$x_{its}^{\text{dep/arr}}$	binary decision variable determining if flight i will depart/arrive at time t or later under scenario s
$v_{is}^{\text{dep/arr}}$	variable for number of periods of delay for flight i under scenario s
$\mathcal{S}$	index set of all scenarios
ϕ_{kts}	binary stochastic outcome of either instrumental meteorological conditions (0) or visual meteorological conditions (1) for airport k at time t in scenario s
p_s	probability of scenario s
$l_i^{\text{dep/arr}}$	maximum departure/arrival delay deviating from the first-stage decision
$c_i^{\text{dep/arr}}$	unit cost of departure/arrival per time period for flight i, with the condition $c_i^{\text{dep}} \leq c_i^{\text{arr}}$
$\mathcal{L}_k$	set of capacity constraints at airport k
a_{kq}, b_{kq}, $Q_{kq}(\phi)$	parameters of capacity envelope at airport k for constraint $q \in \mathcal{L}_k$ under operating condition ϕ
$\mathcal{T}_i^{\text{dep/arr}}$	set of possible departure/arrival times for flight i in the second stage

4.2.1 First-stage formulation

IMSOAN is formulated as the following mathematical program

$$\min_{w} \quad \rho \sum_{i \in \mathcal{F}} \left(\sum_{t \in \tilde{\mathcal{T}}_i^{\text{dep}}} g_{it}(w_{i,t}^{\text{dep}} - w_{i,t+1}^{\text{dep}}) + \sum_{t \in \tilde{\mathcal{T}}_i^{\text{arr}}} g_{it}(w_{i,t}^{\text{arr}} - w_{i,t+1}^{\text{arr}}) \right) + (1-\rho)\mathbb{E}_{\phi}[\Psi(w)(\phi)] \quad (4.13a)$$

$$\text{s.t.} \quad w_{it}^{\text{dep}} \le w_{i,t-1}^{\text{dep}}, \ \forall i \in \mathcal{F}, \ \forall t \in \tilde{\mathcal{T}}_i^{\text{dep}} \quad (4.13b)$$

$$w_{it}^{\text{arr}} \le w_{i,t-1}^{\text{arr}}, \ \forall i \in \mathcal{F}, \ \forall t \in \tilde{\mathcal{T}}_i^{\text{arr}} \quad (4.13c)$$

$$\sum_{t \in \mathcal{T}} (w_{it}^{\text{arr}} - w_{it}^{\text{dep}}) = \Delta_i^{\text{sch}}, \quad \forall i \in \mathcal{F} \quad (4.13d)$$

$$\sum_{t \in \mathcal{T}} (w_{jt}^{\text{dep}} - w_{it}^{\text{arr}}) \ge \tau_{ij}, \quad \forall (i, \ j) \in \mathcal{C} \quad (4.13e)$$

$$w_{it}^{\text{dep}}, w_{it'}^{\text{arr}} \in \{0,1\}, \ \forall i \in \mathcal{F}, \ \forall t \in \tilde{\mathcal{T}}_i^{\text{dep}}, \ \forall t' \in \tilde{\mathcal{T}}_i^{\text{arr}} \quad (4.13f)$$

$$w_{it}^{\text{dep}}, \ w_{it'}^{\text{arr}} = 1 \qquad \forall i \in \mathcal{F}, \ \forall t \prec \tilde{\mathcal{T}}_i^{\text{dep}}, \ \forall t' \prec \tilde{\mathcal{T}}_i^{\text{arr}} \quad (4.13g)$$

$$w_{it}^{\text{dep}}, \ w_{it'}^{\text{arr}} = 0 \qquad \forall i \in \mathcal{F}, \ \forall t \succ \tilde{\mathcal{T}}_i^{\text{dep}}, \ \forall t' \succ \tilde{\mathcal{T}}_i^{\text{arr}} \quad (4.13h)$$

where $\mathbb{E}_{\phi}[\Psi(w)(\phi)]$ is the expected value of the second stage, which must be approximated by using the decomposition formulation (4.15) discussed later.

The objective of (4.13) is to minimize the cost of changing the preferred departure and arrival times in the schedule, weighted against the expected cost of delays. The interpretation of all constraints is as follows:

- (4.13b)–(4.13c) is the non-decreasing condition on the binary decision variables
- (4.13d) enforces that the en-route time corresponds to the scheduled time
- (4.13e) enforces minimum connection times between flights
- (4.13g)–(4.13h) is the convention for time indices outside the considered time index sets, and also ensures that maximum displacement is withheld

4.2.2 Second-stage formulation

The second stage problem, taking the first-stage decision w and stochastic outcome ϕ as arguments, has the formulation

$$
\begin{aligned}
\Psi(w)(\phi) = \min_{x} \quad & \sum_{i\in\mathcal{F}} (c_i^{\text{dep}} v_i^{\text{dep}} + c_i^{\text{arr}} v_i^{\text{arr}}) && (4.14\text{a})\\
\text{s.t.} \quad & && \\
& x_{it}^{\text{dep}} \le x_{i,t-1}^{\text{dep}}, \qquad \forall i \in \mathcal{F},\ \forall t \in \mathcal{T}_i^{\text{dep}} && (4.14\text{b})\\
& x_{it}^{\text{arr}} \le x_{i,t-1}^{\text{arr}}, \qquad \forall i \in \mathcal{F},\ \forall t \in \mathcal{T}_i^{\text{arr}} && (4.14\text{c})\\
& \sum_{t\in\mathcal{T}} (x_{it}^{\text{dep}} - w_{it}^{\text{dep}}) = v_i^{\text{dep}}, \qquad \forall i \in \mathcal{F} && (4.14\text{d})\\
& \sum_{t\in\mathcal{T}} (x_{it}^{\text{arr}} - w_{it}^{\text{arr}}) \le v_i^{\text{arr}}, \qquad \forall i \in \mathcal{F} && (4.14\text{e})\\
& v_i^{\text{dep}} \le l_i^{\text{dep}}, \qquad \forall i \in \mathcal{F} && (4.14\text{f})\\
& v_i^{\text{arr}} \le l_i^{\text{arr}}, \qquad \forall i \in \mathcal{F} && (4.14\text{g})\\
& \sum_{t\in\mathcal{T}} (x_{jt}^{\text{dep}} - x_{it}^{\text{arr}}) \ge \tau_{ij}, \qquad \forall (i,j) \in \mathcal{C} && (4.14\text{h})\\
& \sum_{t\in\mathcal{T}} (x_{it}^{\text{arr}} - x_{it}^{\text{dep}}) \ge \Delta_i^{\min}, \qquad \forall i \in \mathcal{F} && (4.14\text{i})\\
& \sum_{t\in\mathcal{T}} (x_{it}^{\text{arr}} - x_{it}^{\text{dep}}) \le \Delta_i^{\max}, \qquad \forall i \in \mathcal{F} && (4.14\text{j})\\
& \begin{matrix} a_{kq} \sum_{i\in\mathcal{F}_k^{\text{dep}}} (x_{it}^{\text{dep}} - x_{i,t+1}^{\text{dep}}) \\ + b_{kq} \sum_{i\in\mathcal{F}_k^{\text{arr}}} (x_{it}^{\text{arr}} - x_{i,t+1}^{\text{arr}}) \end{matrix} \le Q_{kq}(\phi_{kt}),\ \forall k \in \mathcal{K},\ \forall q \in \mathcal{L}_k,\ \forall t \in \mathcal{T} && (4.14\text{k})\\
& x_{it}^{\text{dep}} \ge w_{it}^{\text{dep}}, \qquad \forall i \in \mathcal{F},\ \forall t \in \mathcal{T}_i^{\text{dep}} && (4.14\text{l})\\
& x_{i,t-(\Delta_i^{\text{sch}}-\Delta_i^{\min})}^{\text{arr}} \ge w_{it}^{\text{arr}}, \qquad \forall i\in\mathcal{F},\ \forall (t-(\Delta_i^{\text{sch}}-\Delta_i^{\min}))\in\mathcal{T}_i^{\text{arr}} && (4.14\text{m})\\
& x_{it}^{\text{dep}},\ x_{it'}^{\text{arr}} \in \{0,1\}, \qquad \forall i\in\mathcal{F},\ \forall t\in\mathcal{T}_i^{\text{dep}},\ \forall t'\in\mathcal{T}_i^{\text{arr}} && (4.14\text{n})\\
& v_i^{\text{dep}},\ v_i^{\text{arr}} \ge 0, \qquad \forall i \in \mathcal{F} && (4.14\text{o})\\
& x_{it}^{\text{dep}},\ x_{it'}^{\text{arr}} = 1 \qquad \forall i\in\mathcal{F},\ \forall t\prec\mathcal{T}_i^{\text{dep}},\ \forall t'\prec\mathcal{T}_i^{\text{arr}} && (4.14\text{p})\\
& x_{it}^{\text{dep}},\ x_{it'}^{\text{arr}} = 0 \qquad \forall i\in\mathcal{F},\ \forall t\succ\mathcal{T}_i^{\text{dep}},\ \forall t'\succ\mathcal{T}_i^{\text{arr}} && (4.14\text{q})
\end{aligned}
$$

where the objective is to minimize the cost of additional delays in all airports given the outcome of the weather and the set schedule from the first-stage. The interpretation of all constraints is as follows:

- (4.14b)–(4.14c) is the non-decreasing condition on the binary decision variables
- (4.14d)–(4.14e) defines the variables for the delay relative the first-stage schedule, and note that (4.14e) is an inequality since an early arrival doesn't incur a cost
- (4.14f)–(4.14g) are the constraints for maximum delay for departures and arrivals
- (4.14h) enforces minimum connection times between flights
- (4.14i)–(4.14j) constrains the delays to adhere the minimum and maximum en-route travel time
- (4.14k) is the capacity constraint, which is also the only place where the stochastic outcome enters the model
- (4.14l)–(4.14m) are valid inequalities to tighten the feasible set
- (4.14p)–(4.14q) is the convention for time indices outside the considered time index sets

4.3 Solution procedure

Both stages of IMSOAN is a Mixed-Integer Problem (MIP) with a considerable amount of variables which causes problems when the model is solved. In (Wang & Jacquillat, 2020) they reported that the deterministic equivalent could be solved in CPLEX with no more than five scenarios, and a decomposition procedure was proposed.

Due to the extensive feasible set of the first-stage decision, the integer L-shaped method alone proved insufficient. Therefore, Wang and Jacquillat (2020) suggested that the linear relaxation of the second-stage problem would be used together with a new kind of cut called *dual integer cuts.* These cuts are based on the classic dual representation cuts of the LP second-stage, enhanced by adding the reduced cost of the second-stage variables multiplied by the first-stage decision variables. Using reduced costs for cuts is motivated by the tightening constraints (4.14l) and (4.14m) which implies that for different values of the first-stage decision w, the integer solution of the second-stage would scale by its reduced cost of the second-stage variable due to the bounds. It was shown (see Wang & Jacquillat, 2020, Appendix B.2–B.4) that this gives tighter valid cuts for the given problem.

Furthermore, they used techniques from LP cuts theory called *local branching* and *Pareto-optimality cuts* to enhance the dual integer cuts even further. Lastly, original neighbourhood constraints were added to change from exploration to exploitation, and bounds on the optimal solution value were derived to support the solution procedure. With the decomposition procedure, they solved IMSOAN with up to 30 scenarios, which was shown computationally to give better solutions than with five scenarios.

4.3.1 Decomposition formulation

To solve the model (4.13) by decomposition, the second-stage objective is relaxed into the variables θ_s for a finite scenario set $s \in \mathcal{S}$. The decomposition uses multi-cut relaxation, meaning that the objective of the second-stage objective is relaxed for each scenario instead of relaxing the expected value of all scenarios into a single variable. The master problem is then given by

$$
\begin{aligned}
\min_{w} \quad & \rho \sum_{i \in \mathcal{F}} \left(\sum_{t \in \tilde{\mathcal{T}}_i^{\mathrm{dep}}} g_{it}(w_{i,t}^{\mathrm{dep}} - w_{i,t+1}^{\mathrm{dep}}) + \sum_{t \in \tilde{\mathcal{T}}_i^{\mathrm{arr}}} g_{it}(w_{i,t}^{\mathrm{arr}} - w_{i,t+1}^{\mathrm{arr}}) \right) + (1-\rho) \sum_{s} p_s \theta_s \\
\text{s.t.} \quad & \theta_s \geq \Psi(w)(\phi_s), \quad \forall s \in \mathcal{S} \\
& w \in W, \\
& \theta_s \geq 0, \qquad\qquad \forall s \in \mathcal{S}
\end{aligned} \tag{4.15}
$$

where W is the constraint set (4.13b)–(4.13h) and $\Psi(w)(\phi_s)$ is the sub-problem (4.14).

4.4 Scenario generation in (Wang & Jacquillat, 2020)

Wang and Jacquillat (2020) used scenario reduction to construct the scenario set, which is a distribution-based approach. Among the 1826 historical data points they mapped all outcomes onto a smaller set of outcomes of a predetermined size as described in Section 3.4. Once the mapping is computed, the probabilities are set to the aggregation probability of all outcomes which are mapped to the same point. See (Wang & Jacquillat, 2020, Section 5) for more details.

A shortcoming of this approach is that it is solely distribution-based, meaning that there is no way for the scenario generation to consider which parts of the distribution is more critical for the problem at hand. It is discussed later that a lot of the historical data with much good weather result in similar recourse costs, i.e. those cases will not trigger significant bottlenecks in the network and are therefore not qualitatively very different. Similar recourse costs hold for up to 50% of the data, meaning that the distribution-based approach will emphasize this part of the distribution very much, which may be counterproductive. Furthermore, using scenario reduction, we only consider the 1826 historical data points, while the distribution, in reality, has 2^{432} to 2^{2160} possible outcomes.

It should be mentioned that (Wang & Jacquillat, 2020) was not focused on scenario generation in detail, and using the reduction approach will get stable results as long as the number of scenarios is large enough.

4.5 Problem characteristics

The IMSOAN model has some interesting characteristics which are worth considering for the scenario analysis.

First, it can be observed that the objective is monotonically decreasing with the binary stochastic variables. Intuitively, this means that an additional time period with good weather can only improve the objective; otherwise, it will stay the same. The monotonicity means that the scenarios have a particular ordering relation according to which outcomes are strictly larger in all the binary stochastic variables.

However, since the outcomes are binary and the span in each dimension is only two outcomes, this ordering relation isn't beneficial. There is, however, still a notion that outcomes of more bad weather result in higher costs than those with less bad weather but the combinatorial space of which time periods this accounts for is very large, and the outcomes need to be evaluated to find this out. Some variation of this property for scenario generation was tested and shown to be less effective than scenario reduction in (Wang & Jacquillat, 2020). This property is however observable in the results shown later.

Secondly, the weighting parameter between the different stages' costs ρ largely determines both the instability of the problem and the run-time of the decomposition procedure. Lower ρ corresponds to longer run-time and more instability. This instability occurs because a lower emphasis on the first-stage cost will open up for a broader set of relevant first-stage solutions. If we set $\rho = 0.0$, the whole set of first-stage solutions are more or less relevant; therefore, the decomposition approach which considers only one first-stage solution at a time does not converge within a reasonable time. On the other hand, if we set $\rho = 1.0$ there will be no emphasis on the second-stage and the optimal solution will be not to displace any of the scheduled departure and arrival times. The lower ρ is, the more amount of displacements will be relevant.

For the values $\rho \in \{0.46, 0.67, 0.82, 0.95\}$ used by Wang and Jacquillat, the decomposition procedure will finish in reasonable time (never more than 120 hours for the largest instances), especially for higher values of ρ. However, if $\rho = 0.2$, the decomposition procedure will struggle to finish with the current implementation (smaller instances can run for more than 150 hours without reaching the 1% tolerance gap).

5 Scenario generation

This chapter introduces how problem-based scenario sets for binary distributions can be generated, supported by stability arguments for stochastic problems. The framework is general and agnostic to the specific problem, and can in principle be applied to any problem specified by a binary distribution.

The scenario generation problem is to select a set of outcomes with corresponding probabilities such that the combined set represents the response of the first-stage decision in the output-distribution well enough to solve the problem. Because of the significance of the simplification of the scenario set, we need to find outcomes which, when hedged against, also cause the first-stage decision to very effectively hedge against possible outcomes not included in the constructed scenario set. We name this the *hedging property* of outcomes.

Assume for this chapter that we have an empirical discrete distribution of historical data with a finite number of outcomes. For simplicity of exposition, we assume outcomes are equiprobable, although that isn't strictly needed. We assume that the historical data is an accurate representation of the future. Out-of-sample evaluation is done over the empirical distribution.

The scenario set that approximates the empirical distribution is directly associated with the specific first-stage solution that results from solving the stochastic program using that specific scenario set. Further, a first-stage decision is directly associated with an output-distribution since that first-stage decision is used to compute the empirical output-distribution by out-of-sample evaluation. This relationship is essential to understand this chapter, and has been illustrated in Figure 5.1.

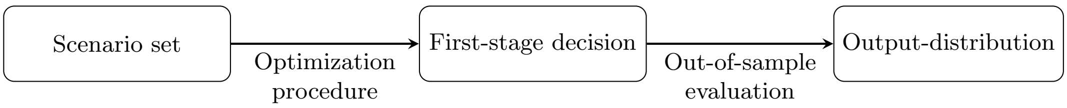


Figure 5.1: Flowchart of how various quantities are obtained.

We use clustering for scenario generation and remind the reader of the three steps of scenario generation by clustering (see Section 3.3):

- Selection of clustering variables and distance metric
- Selection of clustering technique
- Selection of a representative scenario from each cluster

where points one and three are addressed first. Similar outcomes are considered to be in the same *cluster*, and each cluster should have one *representative outcome*. The similarity between outcomes is determined by the hedging property, discussed in Section 5.2, and clustering techniques are introduced in Section 5.5.

Note on feasibility

When we solve a stochastic model with a small scenario set and afterwards evaluate out-of-sample, it may become an issue that the first-stage solution makes the second-stage infeasible for some outcomes. Infeasibility was never an issue for the IMSOAN model presented in Chapter 4 because there always existed a feasible solution.

The counter-argument against having to deal with infeasibility for out-of-sample evaluation is that the model of your system may not be very good. If it is possible to solve the problem such that some outcomes which you first-stage solution did not hedge against could cause infinite costs, then the issue lies in the model formulation, because infinite costs (almost) never occur in reality. If, however, this is a well-founded concern, a robust optimization formulation may be more appropriate for the given application, see (Ben-Tal, El Ghaoui & Nemirovski, 2009).

For stochastic optimization, we promote the use of soft constraints in second-stage formulations, i.e. assigning a cost penalty in the objective function for feasibility violations, to disregard issues with infeasibility within the scenario generation framework presented in this thesis. See (King & Wallace, 2012, Section 2.4) for details on modelling with feasibility.

5.1 Departing from accurate representations of the input-distribution

A vital distinction when doing scenario generation by problem-based methodologies is that we allow the scenario set to depart from statistically accurate representations of the underlying stochastic phenomenon. Leaving statistical representability is justified by the fact that, at the same time as representing the empirical distribution, we simultaneously consider how the second-stage will transform it. Remember, it is only the transformed distribution, the output-distribution, which is considered by the solution algorithm. In problem-based scenario generation, we evaluate the importance of including various outcomes from the empirical distribution by quite different qualities than statistical representability of the input-distribution.

For the modeller, it may seem intuitively wrong at first to depart from statistically accurate representations of the stochastic phenomenon. One source of confusion may be that we represent the scenario set in the input domain but approximate something in the output domain. We recognize this difficulty and explain this in more detail by looking at evidence from two papers.

Zhao and Wallace (2016) looked at the facility layout problem in a stochastic setting and established that redundancy (duplication of machines) was needed to obtain enough flexibility to deal with variations in demand. The continuous empirical distribution was partitioned into three intervals, and the representative outcome of each interval was chosen to be the maximum outcome in each interval. The rationale behind this unconventional choice is that the machine pairs used for upper demand levels in each interval are feasible in the problem for lower demand levels as well (Zhao & Wallace, 2016, Appendix 1). The fact that feasibility is conserved across outcomes is precisely the hedging property of outcomes. They further comment that this is a weak representation of the statistical phenomenon, but that it works better for the given problem than any other three-point distributions they tried because it best captures the problem characteristics.

Prochazka and Wallace (2020) found the scenario set by a 'black box' fitting algorithm based on relations between in-sample and out-of-sample evaluation of the expected objective value, see details in Section 3.1. Interestingly, they relaxed probabilities in the scenario set to 'weights' and made them free variables in the fitting algorithm, meaning that they didn't have to sum to one, which resulted in much better results. The relaxation caused a correction of overestimation in the in-sample expected objective of the scenario sets because the weights generally summed to below one. The following arguments are an interpretation of the result in (Prochazka & Wallace, 2020) by this thesis' author.

Relaxing probabilities to weights without constraints can be equivalently described by a more intuitive representation we use to explain what is going on. Consider that instead of relaxing probabilities to weights, we multiply each recourse value by a correction coefficient, expressed as

$$\sum_s w_s Q(x)(\xi^s) \iff \sum_s c_s p_s Q(x)(\xi^s) \quad \text{s.t.} \quad \sum_s p_s = 1 \tag{5.1}$$

where w_s are weights and c_s are the suggested coefficients. The coefficient c_s represents a correction of the recourse value of the outcome ξ^s so that it better represents the set of outcomes which ξ^s fulfils the hedging property for.

Consider the index set $\mathcal{S}'_s$ of all outcomes in the empirical distribution which are under the hedging property of outcome ξ^s. There is no rationale supporting that the expected recourse value within $\mathcal{S}'_s$ corresponds to $Q(x)(\xi^s)$, mainly because $\mathcal{S}'_s$ is single-handedly derived from problem-based properties based on the outcome ξ_s. If we change the empirical distribution's probability of each outcome, $\mathcal{S}'_s$ stays the same and the expected recourse value within $\mathcal{S}'_s$ will change. To accurately represent the output-distribution, we need $Q(x)(\xi^s)$ to approximate $\mathbb{E}_{s' \in \mathcal{S}'_s} Q(x)(\xi^{s'})$ as precisely as possible; however, we have already argued that this is only coincidental if fulfilled. To correct for the expected value, we can multiply by the correction coefficient derived from both the problem-based properties by $\mathcal{S}'_s$ and the input-distribution by the probabilities $p_{s'}$ $\forall s' \in \mathcal{S}'_s$ so that

$$c_s Q(x)(\xi^s) \approx \frac{\sum_{s' \in \mathcal{S}'_s} p_{s'} Q(x)(\xi^{s'})}{\sum_{s' \in \mathcal{S}'_s} p_{s'}}. \tag{5.2}$$

The argument here is that $\mathcal{S}'_s$ is single-handedly problem-based and we cannot change that. If we want an as concise and effective scenario set as possible, it is wise to pick outcomes for the scenario set that are as mutually exclusive and collectively exhaustive as possible by their covering of the hedging property for all possible outcomes in the empirical distribution.

We infer from the result in (Prochazka & Wallace, 2020) that the hedging property can be covered to a great extent, but because the hedging property does not conserve expected values, a correction coefficient may be needed to remove the resulting bias. By relaxing probabilities to weights, they not only corrected the bias but also simultaneously opened up for the fitting algorithm to explore a larger set of potential outcomes which cover the hedging property to a much larger extent. We postulate that the larger set of potential combinations of outcomes made for much of the improvements they experienced.

The takeaway from these arguments is that the hedging property of outcomes is essential and should be at the centre of consideration. Correction coefficients can be useful, but we establish that, *for now*, it is sufficient to determine probabilities by aggregating probabilities among the outcomes in a cluster.[1]

If we, for a given problem, could determine the hedging property analytically, problem-based scenario generation would be much easier. This is, however, often not the case, and the rest of this chapter explains how we can find it by empirical estimation.

5.2 Proximity of outcomes

We use the notion of proximity of outcomes to estimate the hedging property of outcomes, and in practice, this is used to perform scenario generation by clustering. *Proximity of outcomes* means that for a given problem formulation, hedging the first-stage decision against one outcome causes the first-stage decision to, partly or completely, hedge against other outcomes in that outcomes' proximity. Three notions of proximity are introduced for this thesis in the context of general stochastic problems:

[1] Bias correcting coefficients was tested numerically for the case study problem to be useful, but not enough to be an important point in itself within the framework for the rest of this thesis.

- Proximity in the input domain
- Proximity in the output domain
- Proximity by similar recourse response

where one notion of proximity need not imply any of the others, and all are important in their own ways.

Proximity in the input domain is important because we represent the scenario set in that domain. We define a *connectible binary set* as a set of similar binary points that can all be incrementally changed to reach each of the other points while still conserving its notion of similarity.[2] If a cluster is non-connectible in the input domain, it's difficult to represent the cluster by a single outcome without ending up outside the cluster. By using L_1-distances in the input domain, we avoid making clusters which are very highly non-connectible in the input domain. That is because an incremental change between binary vectors is the smallest non-zero L_1-distance between binary vectors. Thus, it is more likely to keep clusters close in terms of being connectible when we combine it with other notions of proximity. As a comparison, scenario reduction uses only proximity in the input domain as a clustering variable.

Proximity in the output domain is a way of, very approximately, distinguishing the general recourse value different outcomes will take. Simply put, only clustering by this notion of proximity is equivalent to 'bucketing' the range of recourse values from an output-distribution into equally sized intervals and using all outcomes within each interval as clusters. When combined with other notions of proximity, the intervals will no longer be same-sized, but there will be a notion of splitting the range of recourse values into intervals. That ensures that we incorporate into the scenario set an accurate representation of the range of recourse values in the output domain.

In practice, we need not evaluate this notion of proximity for more than one output-distribution, but using multiple different first-stage decisions can help with accuracy. As a comparison, the most straightforward clustering approach in (Sun et al., 2018) using only objective values is using only this notion of proximity.

Proximity by similar recourse response is the most important notion of proximity and is an original contribution of this thesis. To evaluate this notion of proximity, a set of multiple output-distributions is needed, i.e. found by many first-stage decisions. These first-stage decisions must be of reasonable quality related to the problem at hand, which is discussed in more detail in the next section. The rationale is that we want to capture how sets of outcomes may change recourse value in the same way by a change in the first-stage decision.

For this we define *co-deviation of outcomes* over sets of first-stage decisions. Two outcomes that change similarly in the recourse value with changes in the first-stage decision *co-deviates* over that set of first-stage decisions. If they change in opposite directions, they *anti-deviate*.

Co-deviating outcomes are more likely to both be hedged against if one is hedged against because they, by definition, change similarly with changes in the first-stage decision. Thus, we define co-deviating outcomes to be in proximity to each other in the context of making effective scenario sets.

This notion of proximity extracts the problem-based structure of changes in first-stage decisions. As justified in Section 2.3, replicating how the output-distribution change with changes in the first-stage decision is essential in scenario generation and is also the reason why proximity by similar recourse response is the most important notion of proximity. A further explanation of proximity by similar recourse response is given in Section 5.4.

[2]This is different from the topological notion of connectedness. In a continuous setting, we would have used the notion of non-convexity. In lack of a better word, the term 'connectible' was invented for this purpose.

5.3 Collections of output-distributions

This section explains the foundation for determining proximity of outcomes by problem-based qualities. The main difficulty comes from the fact that we need to incorporate how output-distributions change qualitatively with changes in the first-stage decision. This is fundamentally unavailable for most problem formulations; therefore, we present a way of estimating it empirically and argue for the rationale behind why it should work well.

Substantial empirical evidence supports that stochastic programs are often more stable with perturbations of the distribution and easier to solve by approximating scenario sets than their mathematical formulation implies, which can be induced from every paper published where reasonably large stochastic programs are solved in applications. No theoretical argument explains this completely, although (Römisch & Wets, 2007a) on the Lipschitz continuity of stochastic programs is supporting evidence. Even though there is fascinatingly high stability in stochastic programs, some problems require larger scenario sets than others to give reliable solutions. That can, in some cases, be the determining factor for tractability. Addressing this challenge is the role of scenario generation.

On a high level of explanation, a good scenario set should compensate against the structure of the problem formulation that makes it unstable. In most cases, we never know, nor can find, such a problem structure for our problems. The underlying assumption is, however, that there is a reasonable level of stability; otherwise, we wouldn't be able to solve the problem at all. To summarize, the problem is stable to a certain extent, but not enough to solve the model reliably. It is this last bit of instability we attempt to find in the problem structure to later compensate against in our scenario set.

We postulate that naively generated scenario sets can be used as a proxy for determining the structure of a problem formulation which explains its stability properties. Naive, in this context, means a *reasonable* scenario set which by empirical stability testing would be too unreliable to solve the problem. Furthermore, the problem structure with respect to the stochastic variables is inferred by the collection of output-distributions which corresponds to the first-stage solutions from solving the problem with those scenario sets.

In this thesis, we do not attempt to prove why this is true since that would involve proving why stochastic programs are more stable than we expect. We can, however, verify it empirically and explain its validity by relating to other results.

First, scenario sets should approximate the output-distribution for better first-stage solutions, argued by the fact that a solution procedure would easily disregard terrible solutions. By considering only the set of first-stage decisions which results from solving the problem with a naive scenario set, we ignore other more or less irrelevant first-stage solutions. First-stage solutions which cannot be obtained by even a naively generated scenario set are not relevant to consider.

Second, there is a reason why the naive scenario sets are not sufficient to solve the problem. We want our representative set of first-stage decisions to represent the instability we need to address as sincerely as possible.

Three alternative approaches have been observed in previous literature for finding first-stage decisions to represent problem characteristics; the expected value solution, the solutions of perfect information and solutions which differ by what a (bad) heuristic can find in consecutive runs (Feng & Ryan, 2016; Prochazka & Wallace, 2020; Sun et al., 2018).

The first two are disregarded by the fact that both approaches result in only a single output-distribution. Thus variations across first-stage solutions, which is the most important property, is not considered. Furthermore, their first-stage solutions are only considering a single outcome at a time which results in *structurally different solutions* than considering multiple scenarios at the same time (Wallace, 2010). This puts the methods in the realm of what-if analysis, which is what we attempt to avoid by using stochastic programming. Thus, problem properties inferred from such solutions may only be useful by coincident and is not necessarily transferable to other problem classes.

The third approach considers variations across multiple first-stage decisions, but here we can argue that the variations among them are as much due to the heuristic as to the instability of the problem.

This thesis argues that solving the problem by the best possible solution algorithm with reasonable scenario sets leaves the variability in the set of resulting first-stage decisions only due to the actual instability from the structure of the problem formulation.

The collection of first-stage decisions is referred to as the *approximate solution set* and the corresponding *collection of output-distributions* is the object of analysis from which we extract the problem structures by the introduced notions of proximity.

5.3.1 Minimum transportation distance from collections of output-distributions

This section introduces the notation for the proposed scenario generation method. Let $\mathcal{U}$ be a collection of naively generated scenario sets from the empirical distribution. Then we consider the *approximate solution set*

$$X_{\mathcal{U}} = \{x : x \in \operatorname{argmin}\{f_U(x)\},\ U \in \mathcal{U}\}, \tag{5.3}$$

which is the set of first-stage solutions from solving the problem with each of the scenario sets in $\mathcal{U}$. A property of $X_{\mathcal{U}}$ is that it is a subset of the feasible region

$$X_{\mathcal{U}} \subseteq X, \tag{5.4}$$

and in practice much smaller. The corresponding set of output-distributions is expressed by

$$\begin{aligned} \mathcal{D}_{\mathcal{U}} &= \{f(x)(\cdot) : \Xi \to \mathbb{R} \quad \text{s.t.} \quad x \in X_{\mathcal{U}}\} \\ &= \{f(x)(\cdot) : \Xi \to \mathbb{R} \quad \text{s.t.} \quad x \in \operatorname{argmin}\{f_U(x)\},\ U \in \mathcal{U}\} \end{aligned} \tag{5.5}$$

which, empirically, are the out-of-sample evaluations of the first-stage decisions in the approximate solution set $X_{\mathcal{U}}$.

We formulate clustering variables $u^{\mathcal{U}}(\xi)$ for each of the outcomes which are explicitly based on the collection of output-distributions and define distances between them as

$$d(\xi', \xi) = \|u^{\mathcal{U}}(\xi') - u^{\mathcal{U}}(\xi)\| \tag{5.6}$$

with $\|\cdot\|$ some norm like the L_1- or L_2-norm. These clustering variables are explained in Section 5.4.

Next, we generate a scenario set such that it minimizes the transportation distance between the empirical distribution and the scenario set by the Kantorovich-Rubinstein distance with respect to distances between the clustering variables $u^{\mathcal{U}}(\xi)$

$$\hat{\mu}_d(P, \mathcal{T}) = \min_{\eta} \quad \sum_{i=1}^{N}\sum_{j=1}^{M} d(\xi_i^P,\ \xi_j^{\mathcal{T}})\eta_{ij} \tag{5.7}$$
$$\text{s.t.} \quad \eta_{ij} \geq 0,\ \sum_{j=1}^{M}\eta_{ij} = p_i^P,\ \sum_{i=1}^{N}\eta_{ij} = p_j^{\mathcal{T}}, \quad \forall i, j$$

where $p_i^P, p_j^{\mathcal{T}}$ are the probabilities of the discrete outcomes $\xi_i^P, \xi_j^{\mathcal{T}}$ for the empirical distribution and the scenario set, of sizes N, M, respectively. Thus, our scenario set is determined by

$$\mathcal{T} = \operatorname{argmin}\{\hat{\mu}_d(P, \mathcal{T})\ :\ \mathcal{T} \text{ a discrete distribution of cardinality } M\}. \tag{5.8}$$

which is non-trivial to solve, especially since the scenario set outcomes are in a binary domain. This is discussed in detail in Section 5.5.

To summarize, the minimum transportation distance $\hat{\mu}_d$ is a suggested new metric to determine the discrepancy between the empirical distribution and scenario sets based on empirically determining $d(\xi', \xi)$ based on collections of output-distributions. Since all quantities are empirically obtainable, this metric can be minimized by applying centroid-based clustering methods. We suggest this serves as a viable alternative to scenario generation motivated by the Fortet-Mourier metric.

It is an original contribution of this thesis to determine distances between outcomes by notions of proximity on collections of output-distributions motivated by a combination of the literature on both mathematical and empirical stability theory.

5.3.2 Comparison with the Fortet-Mourier probability metric

Based on the Fortet-Mourier probability metric, which has previously been suggested as the canonical metric for scenario generation (Dupačová et al., 2003), we argue for the use of the approximate solution set and the proposed notions of proximity on collections of output-distributions instead.

The Fortet-Mourier (FM) probability metric between the empirical distribution P and a scenario set $\mathcal{T}$ is formulated as

$$d_{\mathcal{F}_\rho}(P, \mathcal{T}) = \sup\left\{ \left| \int_\Xi f(x)(\xi)P(d\xi) - \int_\Xi f(x)(\xi)\mathcal{T}(d\xi) \right| \quad \text{s.t.} \quad f(x) \in \mathcal{F}_\rho \right\} \tag{5.9}$$
$$\text{where} \quad \mathcal{F}_\rho = \{f(x)(\cdot) : \Xi \to \bar{\mathbb{R}} \quad \text{s.t.} \quad x \in X \cap \rho\mathbb{B}\},$$

and we stress that the FM metric is a theoretical tool which cannot be obtained in itself, but has been used as motivation for alternative scenario generation methods.

First, the set of output-distributions considered in FM, $\mathcal{F}_\rho$, is quite large. The ball $\rho\mathbb{B}$ is centred in the origin, and need to contain the true optimal solution set $S(P) = \operatorname{argmin}_{x\in X} f_P(x)$ as well as satisfying $v(P) = \min_{x\in X} f_P(x) \geq -\rho$ by the perturbation results in (Rockafellar & Wets, 2009, Section 7J). Taking the case study problem IMSOAN as an example, such a set also includes the first-stage decision from not considering the second-stage cost at all.

The alternative we suggest is to use the (argued) more relevant, more restricted and empirically obtainable set of first-stage decisions, namely the approximate solutions set $X_\mathcal{U}$ with the corresponding set of output-distributions $\mathcal{D}_\mathcal{U}$.

Second, we address the use of the utility metric $f_Q(x) = \mathbb{E}_Q[f(x)(\xi)] = \int_\Xi f(x)(\xi)Q(d\xi)$ to assess distances between distributions, which includes use of the supremum of the distance among the considered set of output-distributions. We have already pointed out in Section 2.3 that matching the exact out-of-sample and in-sample evaluation of the utility metric for a large set of first-stage decisions is not that important. What's important is the quality of the first-stage decision the solution procedure converges to in the end. The supremum in FM and the use of the utility metric on the output-distribution supposes that matching the out-of-sample utility metric is important and that it is important for all first-stage decisions in the first-stage decision set since it is evaluated at the worst first-stage decision. This can be effective to make theoretical bounds on the quality of the solution, but, we argue, not to construct scenario sets to solve the problem.

The combination of a very large first-stage decision set and conservative evaluation of the scenario set quality means that the FM can end up telling us almost nothing of the practical utility of a scenario set. An appropriate way of evaluating the quality of a scenario set must be tighter on the real practical performance of the scenario set. We suggest this is better evaluated by empirical claims on stability, also for motivating generation of scenario sets.

Thus, we relax the use of both the supremum, and the collapsing of the output-distribution into its utility metric. Instead, we consider the distance between the empirical distribution and the scenario set by minimum transportation distance where the distance is determined by the introduced notions of proximity of outcomes which directly involves evaluation on the collection of output-distributions from the approximate solution set.

Some important distinctions from scenario reduction are that we don't restrict ourselves to reduce the set of points to a set of already existing outcomes in the empirical distribution and that the determination of distances between outcomes considers problem-based properties explicitly.

What we suggest may seem similar to scenario reduction since we use minimum transportation with a different distance metric between outcomes. It is, however, no longer evident that our minimum transportation distance (5.8) bounds the FM metric; instead, we expect them to be non-contained in each other. The argument for departing from scenario reduction in the manner of (Dupačová et al., 2003) is that the bound on FM is too loose and that FM in itself may be too conservative.

5.4 Clustering variables on collections of output-distributions

To cluster outcomes, we need to decide clustering variables and an appropriate distance metric between them. The distance between clustering variables should reflect the three proposed notions of proximity. All these notions of proximity are important, and the clustering variables are therefore a combination to include all of them, meaning that proximity in one property and great distance in another is not sufficient to cluster two outcomes. They must exhibit a trade-off of proximity in all properties.

The input domain must be considered together with the other notions of proximity since the scenario set is represented in the input domain. This is to avoid making clusters in the output domain that are highly non-connectible in the input domain. If only the output domain is used for clustering, the selected representative outcome may be outside the cluster or at the very edge of the cluster in the input domain. It may therefore not represent the cluster very well.

To characterize each outcome in the empirical distribution, we use the feature vector consisting of the concatenation of the recourse cost of each first-stage decision and the outcome in the input domain it is evaluated by. This makes the feature vector

$$u^{\mathcal{U}}(\xi) = \begin{bmatrix} c_f Q(x_1)(\xi) \\ \vdots \\ c_f Q(x_n)(\xi) \\ \xi \end{bmatrix} \tag{5.10}$$

where ξ is the outcome vector in the input domain, $\{x_1, \ldots, x_n\} = X_{\mathcal{U}}$ and the coefficient c_f balances the weighting between the input and output domains. Only the recourse cost is used to only capture the stochastic effects of first-stage decisions. We have good control of the deterministic cost and don't want that to interfere with how the uncertainty is represented.

The feature vector captures all three notions of proximity. Proximity in the input domain is conserved by using the outcome vector ξ, while the other two are captured by the recourse vector $[Q(x_1)(\xi), \ldots, Q(x_n)(\xi)]^T$.

Distinctions by large differences in the recourse vector conserve proximity in the output domain. If we would sort the recourse vectors by being strictly less in all elements, the difference in rank numbers is a good indication of proximity in the output-domain.

Proximity by similar recourse response is slightly more subtle to understand, and is determined by similar relationships among elements within the recourse vector. This can be understood more accurately by considering the *anti-deviation of outcomes over the approximate solution set*

$$\mathrm{AnD}_{X_{\mathcal{U}}}(\xi^{s'}, \xi^{s}) = \left\| \begin{bmatrix} Q(x_1)(\xi^{s'}) - \bar{Q}(\xi^{s'}) \\ \vdots \\ Q(x_n)(\xi^{s'}) - \bar{Q}(\xi^{s'}) \end{bmatrix} - \begin{bmatrix} Q(x_1)(\xi^{s}) - \bar{Q}(\xi^{s}) \\ \vdots \\ Q(x_n)(\xi^{s}) - \bar{Q}(\xi^{s}) \end{bmatrix} \right\|_{L_1} \tag{5.11}$$

where $\xi^{s'}, \xi^{s}$ are two different outcomes, $\{x_1, \ldots, x_n\} = X_{\mathcal{U}}$ and

$$\bar{Q}(\xi) = \frac{1}{|\mathcal{U}|} \sum_{x \in X_{\mathcal{U}}} Q(x)(\xi) \tag{5.12}$$

is the average recourse value among the first-stage decisions in the approximate solution set for a given outcome ξ. The anti-deviation between outcomes is simply the pairwise L_1-distance between recourse vectors, corrected by the average recourse for each outcome. It is analogous to correlation over the approximate solution set $X_{\mathcal{U}}$, but instead of multiplication there is the absolute value of the difference between two elements.

A large anti-deviation between outcomes means that they change very differently with changes in the first-stage decision. Zero anti-deviation means that they co-deviate perfectly and give the same response in the recourse function with changes in the first-stage decision.

We may also isolate the effect of proximity by similar recourse response from proximity in the output domain by correcting the feature vector by the average recourse values and get

$$u^{\mathcal{U}}(\xi) = \begin{bmatrix} c_f(Q(x_1)(\xi) - \bar{Q}(\xi)) \\ \vdots \\ c_f(Q(x_n)(\xi) - \bar{Q}(\xi)) \\ \xi \end{bmatrix}. \tag{5.13}$$

The distance matrix which contains all pairwise distances between scenarios is then computed by using the L_1-norm as a distance metric between clustering variables. Thus, the distance matrix is expressed as

$$d_{s's} = \sum_i |u_i^{\mathcal{U}}(\xi^{s'}) - u_i^{\mathcal{U}}(\xi^{s})| = ||u^{\mathcal{U}}(\xi^{s'}) - u^{\mathcal{U}}(\xi^{s})||_{L_1}. \tag{5.14}$$

In principle, we could make clustering variables in other ways based on collections of output-distributions to determine problem-based distances between outcomes. The proposed clustering variables are based on what we consider to be proximity of outcomes in a general context of stochastic programs, and the currently proposed set-up showed promising results in the case study problem.

Example

Consider an example two-stage stochastic programming problem where the input distribution is one-dimensional and can take ten different values from the set $\{1, \ldots, 10\}$. We now give an intuitive explanation of the notions of proximities on such a problem, illustrated in Figure 5.2.

We have sampled a collection $\mathcal{U}$ of five scenario sets and solved our model for each of them to obtain the approximate solution set $X_{\mathcal{U}}$. Afterwards, out-of-sample evaluation is performed using each first-stage decision $x \in X_{\mathcal{U}}$ to obtain the collection of output-distributions $\mathcal{D}_{\mathcal{U}}$. One output-distribution $D \in \mathcal{D}_{\mathcal{U}}$ has a corresponding recourse value for each outcome in the empirical distribution. The recourse values of the five output-distributions are plotted in Figure 5.2. The input domain is along the first axis and the recourse value in the output domain along the second axis. There are ten outcomes in the empirical distribution referred to by their values in the input domain, $\{1,\ldots,10\}$.

First, we observe that the distances in the input domain are quite regular and that by proximity in the input domain, outcome 1 is furthest apart from outcome 10 and closest to outcome 2.

Second, we observe an even increase in recourse values best recognized by the slope of the red dashed average line. Thus, by proximity of the output domain, outcome 1 is furthest apart from outcome 10 and closest to outcome 2.

Lastly, we observe the very pronounced flip in recourse values across first-stage decisions between the outcome pairs $(4,5)$ and $(6,7)$. This can be characterized by the notion of proximity by similar recourse response. If we compute the co-variation over the approximate solution set (5.11) we would discover that there are two clusters which are very close by this metric: $\{5,6\}$ and $\{1,\ldots,4,7,\ldots,10\}$. Had we clustered only based on proximity by similar recourse response, these two clusters would appear.

Now, consider that we combine all notions of proximity by using the feature vector (5.10). A very appropriate choice of the number of clusters would be three in this case which would result in the clusters: $\{1,\ldots,4\}$, $\{5,6\}$ and $\{7,\ldots,10\}$.

Less ideally, had we chosen two clusters, the outcomes would likely be clustered into sets of consecutive outcomes split between the outcome pairs $(4,5)$ or $(6,7)$.

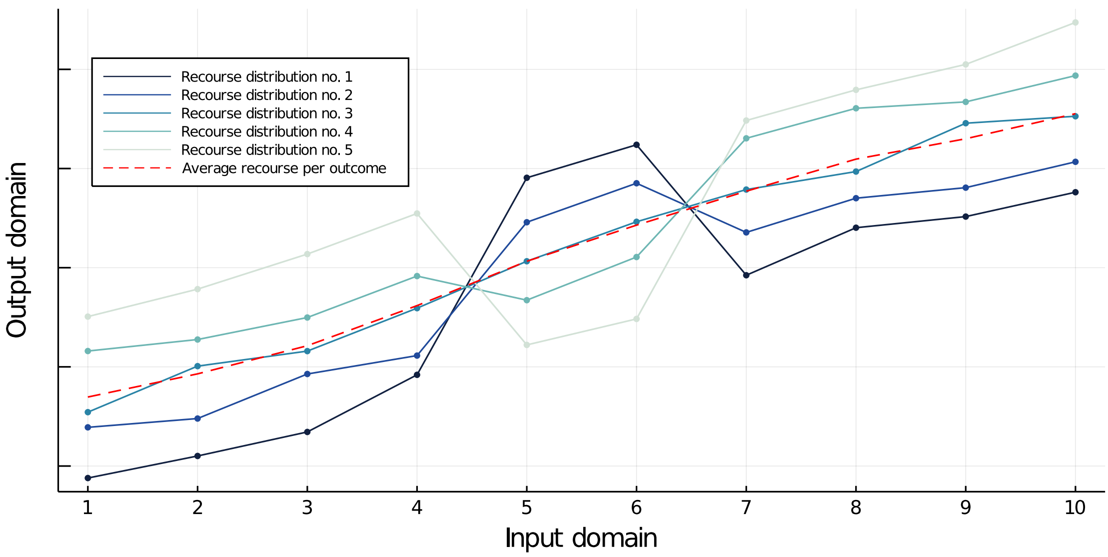


Figure 5.2: Illustration of different notions of proximity, see explanation in text.

This is a simple illustration to explain the concept of measuring proximity by collections of output-distributions. In reality, the input domain has hundreds of dimensions and with a much less intuitive or consistent recourse function.

5.5 Clustering method

This section discusses alternatives for solving (5.8) as best as possible. An exact solution is in most cases intractable, but viable approximations or heuristics are available. We are concerned only with centroid-based clustering as that is the form of the problem (5.8).

Two existing clustering methods for solving (5.8) approximately are discussed, namely the k-means and k-median clustering methods, and a third alternative, binary point centroid clustering, is suggested. The most important distinctions between the different centroid-based clustering methods are

- Which domain the centre-point is represented in, real or binary
- The number of possible values the centre-point can take, either a restricted set or the whole input domain
- How the clusters are formed with respect to the representative outcome of each cluster
- Tractability of solving the clustering problem

Advantages and disadvantages of the different methods are laid out in this section. The methods are tested numerically on the case study problem in Chapter 6.

5.5.1 K-median

The k-median clustering method has the advantage that the centre-point is chosen among the available data points and is therefore binary. The clusters are also then formed with respect to that binary outcome. A disadvantage from selecting centre-points among available data points is that the representative outcome must be chosen among a very restricted set of points. The representative outcome could be better if the whole space of possible outcomes is utilized because the discrepancy between the number of possible outcomes and the size of the dataset is very significant. This argument is most important if the set of historical data points is small, or the number of dimensions of the outcome vector is large.

The tractability of k-median clustering can also be an issue, and the problem can be formulated an integer linear program (5.15) which is NP-hard. The size of the formulation scales as the number of historical data points squared but can be reasonably solved up to 2000 data points based on experience from this thesis. For larger datasets, the approximations forward and backward propagation can be used to solve the problem in polynomial time (Heitsch & Römisch, 2003).

For the results in Chapter 6 of this thesis, the k-median clustering problem was solved by the integer linear program

$$\min_{x,\,y} \quad \sum_{s} d_{s's} x_{s's} \tag{5.15a}$$

s.t.

$$\sum_{s} y_s \leq S \tag{5.15b}$$

$$\sum_{s} x_{s's} = 1 \qquad \forall s' \in \mathcal{S} \tag{5.15c}$$

$$x_{s's} \leq y_s \qquad \forall s', s \in \mathcal{S} \tag{5.15d}$$

$$x_{s's}, y_s \in \{0, 1\} \tag{5.15e}$$

where $\mathcal{S}$ is the index set of all outcomes in the dataset, S is the number of clusters, $d_{s's}$ is the distance from outcome s' to outcome s, y_s is a binary variable determining if outcome s should be used as a centre-point and $x_{s's}$ determines if outcome s' is assigned to the cluster with centre-point s. Constraint (5.15b) ensures the number of scenarios is less than or equal to S, (5.15c) ensures each scenario is only assigned to one cluster and (5.15d) ensures that outcomes are only assigned to active centre-points. Probabilities are aggregated for all outcomes which are assigned to the same centre-point. The centre-point of each cluster is used as the representative outcome.

5.5.2 K-means

K-means clustering uses the average outcome in each cluster as the centre-point. The advantage of this is that the whole (fractional convex hull of the) input domain can be used to represent the centre-point, but the disadvantage is that this generally results in a fractional centre-point. The centre-point have to be converted to a binary outcome which may no longer represent the cluster very well because the clusters are no longer formed with respect to the representative outcome. In this thesis, we choose the binary representative outcome as the one closest to the fractional centre-point in each cluster.

Computationally, k-means is simpler to solve approximately, while solving k-median clustering can be more time-consuming. For this thesis, the k-means clustering problem is performed by the standard implementation in the Julia library 'Clustering.jl'. This implementation requires distances between clustering variables to be evaluated by the L_2-norm.

5.5.3 Binary point centroid clustering

The disadvantage of k-means clustering is that the resulting centre-point is in general fractional, while for k-median the issue is that the set of possible centre-points is very restricted. Given that we find a binary representative point for each cluster after solving k-means, the clusters are no longer formed with respect to that new representative point and the approximation may result in representative points outside the clusters.

An ideal clustering method for scenario generation with binary outcomes would consider the centre-points as decision variables in a binary domain and then form clusters with respect to those centre-points. We call this *binary point centroid clustering*. This method would collect the advantages of both k-means and k-median clustering and rid us of their respective disadvantages. The issue with this clustering method is the size and complexity of solving it. Note that, if tractable, this method solves (5.8) exactly.

The binary point centroid clustering problem can be formulated as a quadratic binary programming problem, for which the quadratic terms can be linearised (Crama & Rodríguez-Heck, 2017). However, this formulation scales very fast, namely as the product of the dimensions of the binary outcome vector, the number of scenarios and the number of historical data points. Additionally, the linearisation of the quadratic binary terms usually is not very efficient.[3]

An alternative to an exact procedure with the quadratic program formulation is to solve by a heuristic. The representation for the heuristic would be very concise, with only one vector to represent the centre point for each cluster. The cost is evaluated greedily by assigning data points to the closest cluster and summing the distances between points and their respective centre points.

The heuristic can be benchmarked against, or initialized with, the solutions from the k-median or the k-means algorithms to guarantee that it gives better solutions. K-means, with an appropriately chosen representative outcome, is a good alternative for initialization and benchmarking due to its ease of implementation and availability in standard software libraries.

The disadvantage of this heuristic approach is that the problem-based recourse values could take a significant amount of time to evaluate inside the heuristic. Evaluating distances in the input domain is very simple, and it could thus be more advantageous to make the scenarios distribution-based by this method. It's difficult to determine ex ante if that trade-off from departing from problem-based methods is valuable, and this should be considered for each specific problem.

Again, binary point centroid clustering may be more valuable if the number of historical data points is small or the dimensions of the outcome vector is large. That's because the discrepancy between the size of the full domain and the number available data points worsens with both fewer historical data point and more dimensions.

If the in-sample evaluation can be done efficiently, problem-based binary point centroid clustering is tractable. For the presently considered ATFM case study problem, this was intractable and has therefore not been tested. Experimenting with binary point centroid clustering is a very

[3] This formulation was tested for the ATFM case study problem, and there was no chance of solving it for even the smallest problem instances.

relevant consideration for further research on problems with easier to evaluate second-stage formulations and binary input distribution.

5.6 Recourse deviation

This section explains how analysing collections of output-distributions can provide insight into the problem we model and can serve as a proxy for determining the presence of uncertainty for a given problem instance.

Presence of uncertainty is defined as the impact uncertainty has on the specific problem formulation, or stated differently, how rich the representation of the uncertainty by the scenario set has to be to solve the problem reliably. It also reflects how challenging scenario generation will be for a given problem. This property is difficult to pinpoint exactly but is nonetheless very important. Refining the understanding of the presence of uncertainty in problem formulations is part of what this thesis addresses. Now, we suggest one proxy for assessing it.

Define the *recourse deviation* as

$$R_{\mathcal{U}}(\xi) = \max\{|D^Q(\xi) - E^Q(\xi)| : D, E \in \mathcal{D}_{\mathcal{U}}\}. \tag{5.16}$$

where Q indicates the stochastic component (the recourse) of the output-distribution. The recourse deviation quantifies the span of possible recourse values for a given outcome ξ.

Scenario generation is increasingly challenging if a given outcome can take widely different recourse values for different first-stage decisions in the approximate solution set. The challenge comes from the fact that large shifts in the recourse values allow substantial variations in the entire output distribution, which also makes it difficult to represent accurately by a scenario set. Thus, the recourse deviation within the collection of output-distributions from the approximate solution set can be analysed to infer the presence of uncertainty for the problem.

A perfectly stable scenario generation procedure producing the collection of scenario sets $\mathcal{U}$ would result in almost no recourse deviation. Evaluating the recourse deviation informs us of how much the ranking of outcomes with respect to its recourse value is likely to change with different first-stage decisions. If the ranking is expected to change much, the problem is unstable and larger scenario sets might be needed to represent the uncertainty accurately.

Recourse deviation as a tool for analysis is exemplified in Section 6.2.3 for the ATFM problem with the recourse deviation illustrated in Figure 6.1.

6 Numerical experiments

Numerical experiments have been conducted to showcase how the insights from this thesis can be used for scenario generation in the Integrated Model of Scheduling and Operations in Airport Networks (IMSOAN).

Section 6.1 is an overview of the experimental set-up and the various parameter configurations that were used for numerical experiments, Section 6.2 is a stability analysis of two benchmarks for scenario generation, and Section 6.3 shows the results of computational experiments with the new suggested scenario generation method.

6.1 Experimental set-up

6.1.1 Parameter configurations

Two problem instances have been selected for the numerical experiments, one with 6 airports (K6) and one with 30 airports (K30). We have used the same dataset as in (Wang & Jacquillat, 2020), and some key figures are given in Table 6.2.

The weighting between the two stages was done with two values; the configuration $\rho = 0.67$ was used because this was one of the lower values used in (Wang & Jacquillat, 2020), while the configuration $\rho = 0.2$ was added in this thesis to make the problem more unstable.

Distances are measured in the input domain, output domain or the feature vector domain (both input and output), by the L_1-norm for k-median and the L_2-norm for k-means. After clustering by k-means, the selected representative binary outcome is the one closest to the centre-point of each cluster. For k-median, the median outcome is chosen to be the representative outcome. Probabilities are determined by the sum of probabilities of all outcomes within a cluster.

A summary of the possible parameter configurations is also given in Table 6.1.

Table 6.1: All parameter configurations used in the numerical experiments.

Parameter	Possible configurations
Problem instance	K6, K30
First-stage weight, ρ	$0.67, 0.2$
Distance metric	Input domain distance, Output domain distance, Feature vector distance
Clustering method	k-median, k-means
Set sizes	S03, S05, S10, S20, S30

Table 6.2: Model constants and dataset key figures.

Quantity	Value	Quantity	Value	
			K6	K30
Cost of displacement; c_d	2.0	Number of airports; $\lvert\mathcal{K}\rvert$	6	30
Cost of ground holding; c_g	1.0	Number of flights; $\lvert\mathcal{F}\rvert$	13453	26368
Cost of airborne delay; c_a	1.2	Number of aircraft connections; $\lvert\mathcal{C}\rvert$	8512	18788
Maximum displacement; δ	1	Smallest maximum delay; $\min_i\{l_i^{\mathrm{dep/arr}}\}$	0	0
Number of time periods; $\lvert\mathcal{T}\rvert$	72	Largest maximum delay; $\max_i\{l_i^{\mathrm{dep/arr}}\}$	21	21
		Number of stochastic variables	432	2160

Sampling Average Approximation (SAA) (Shapiro, 2003) is used as the naive scenario generation procedure for its ease of implementation, and the size of the sampled set can help us adjust what a 'reasonable' scenario set should entail. If sampled scenario sets give only noise in the solutions, the problem is very unstable, and we should increase the size of the sampled scenario sets to make it slightly more stable. In principle, a different naive generation procedure could be used.

The collection of output-distributions were found by sampling ten scenario sets of size three. Using ten scenario sets seemed to give a uniform and consistent recourse deviation, and sets of size three could be used to solve the problem within a reasonable time compared to larger sizes. Using larger sampled scenario sets to construct the collection of output-distributions gave seemingly non-significant differences in the quality of the new scenario sets made from the collection of output-distributions.

The weighting parameter c_f in the feature vector was set so that the maximum possible L_1-distance in the output domain would correspond to the maximum possible L_1-distance in the input domain. This gives approximately equal weight to each of the two respective domains.

6.1.2 Implementation details

The algorithm was implemented in the Julia programming language using the Julia Mathematical Programming (JuMP) package (Dunning, Huchette & Lubin, 2017) with the Gurobi v9 Mixed Integer Programming solver. The problems were solved on a computational cluster with nodes of hardware specifications up to 2×3.5GHz Intel Xeon Gold 6144 CPU (8 core) and 384GB of RAM. The additional computational capability compared to what was used in (Wang & Jacquillat, 2020) meant that the deterministic equivalent could be solved up to the largest problem instance K30 with 30 scenarios with occasional failures due to insufficient memory.

The decomposition procedure was implemented and tested, but due to availability of especially good hardware for this work, all runs are solved by the deterministic equivalent, i.e. solving one large model with both stages together. This was done to rule out any source of error that lies in possible inaccuracies by the relatively large convergence tolerance of 1 % that was needed for the decomposition procedure to converge in a reasonable time. Thus, the scenario sets can be compared with higher certainty that any differences are only due to the scenario sets themselves. Additionally, by using the deterministic equivalent, it was possible to solve the more unstable problem instances where $\rho = 0.2$ which otherwise, for the given experiments and decomposition implementation, would have taken an unreasonable amount of time to complete.

For this model formulation, we are working at the very limit of what can be solved, meaning that both the decomposition procedure and scenario generation are paramount to be able to solve these problems also for larger instances than the ones solved for this thesis.

The out-of-sample evaluations for this problem are especially computationally demanding due to the very large second-stage MIP formulation. A single second-stage evaluation could take between 20–7000 seconds, varying between problem instances and the amount of strain an outcome puts on the air traffic network. More strained networks cause more difficult to solve combinatorial problems. To speed up out-of-sample evaluations, warm-starts where used combined with heuristics to sort scenarios so that the most similar outcomes came in consecutive order. Parallelized distributed computing was utilized for additional speed-ups.

6.2 Stability benchmarks

Stability analysis of the problem is presented to provide a benchmark of what can be expected of a good scenario generation procedure for this specific problem. Three different approaches are used to evaluate stability; in-sample stability, out-of-sample stability and bias. These are described in detail in Section 2.2. Additionally, the recourse deviation presented in this thesis is used to evaluate problem characteristics.

The benchmarks were computed for Sampling Average Approximation (SAA) (Shapiro, 2003) and for the reduction scenarios used in (Wang & Jacquillat, 2020). Lastly, we show how the newly introduced recourse deviation can be used to assess stability.

Stability is recognized in two ways for various generation procedures: (i) A stochastic scenario generation procedure is evaluated by constructing a scenario set multiple times and evaluating stability by the standard deviation among the results. Sample Average Approximation is the only procedure in this thesis for which this holds. Ten sampled sets have been used to find standard deviations. (ii) For deterministic scenario generation approaches stability must be evaluated on whether scenario sets of similar sizes give similar results. This regards all other scenario generation procedures but Sample Average Approximation.[1]

6.2.1 Sample Average Approximation

The computational results for Sample Average Approximation scenario sets are summarized in Table 6.3. There are a few interesting points to note about these results.

[1]Note that there is randomness in some of the other procedures, for example in which scenarios are used to find output-distributions to make the problem-based scenarios or in the k-means clustering algorithm. These sources of randomness are too small to give a significant impact on the results, and are considered deterministic in this context.

Table 6.3: Stability of Sample Average Approximation. Mean and standard deviation of the evaluation of the in-sample and out-of-sample expected objective values among ten sampled scenario sets of respective sizes. "—" = memory issues.

K6								
	$\rho = 0.67$				$\rho = 0.2$			
Set size	In-sample		Out-of-sample		In-sample		Out-of-sample	
	μ	σ	μ	σ	μ	σ	μ	σ
S03	932.652	689.236	1762.080	27.771	1449.115	1343.946	3586.296	102.858
S05	1406.257	830.252	1757.875	39.013	2523.069	1812.786	3527.265	103.277
S10	1802.120	883.423	1754.271	33.770	3320.760	1821.276	3460.731	164.079
S20	1473.303	456.266	1725.853	5.856	2684.094	1009.597	3315.479	65.817
S30	1882.894	208.462	1726.515	5.162	3562.952	421.497	3290.015	21.332

K30								
	$\rho = 0.67$				$\rho = 0.2$			
Set size	In-sample		Out-of-sample		In-sample		Out-of-sample	
	μ	σ	μ	σ	μ	σ	μ	σ
S03	1580.607	941.178	1955.246	30.617	2615.173	1838.152	3937.784	146.783
S05	1496.570	564.953	1952.798	23.688	2450.198	1108.897	3872.433	63.825
S10	1883.778	791.222	1926.572	29.775	3346.126	1583.193	3744.710	123.278
S20	2108.888	453.404	1909.665	11.631	3911.903	1003.990	3572.293	17.265
S30	—	—	—	—	—	—	—	—

The in-sample stability is really bad for all sampled scenario sets. This can be recognized by the large standard deviation of the in-sample expected recourse values for both problem instances with both configurations of ρ. It does, however, noticeably converge for larger scenario sets but never starts to become in-sample stable.

Related to the low in-sample stability, we see that the bias can be quite high. This is recognized by comparing the mean expected objective value in-sample against the out-of-sample value. Because of the high in-sample standard deviation, the bias can also be very variable. This means that the sampled scenario sets do not provide a very good representation of the expected value of the output-distribution.

A very interesting point about these results is that the solutions are quite stable out-of-sample. This is recognized by the fact that the standard deviation is low for out-of-sample objectives, at only about 0.3% of the mean expected objective value. With scenario sets of size 20, we see that the sampled scenario sets are starting to get out-of-sample stable, but they are not stable in-sample with a standard deviation of about 11% of the mean expected objective value.

The instability of the instances with $\rho = 0.2$ is very noticeable larger than for $\rho = 0.67$ with standard deviations being three to four times as high for the out-of-sample objectives, meaning that the presence of uncertainty is higher for instances with $\rho = 0.2$.

6.2.2 Scenario Reduction

The computational results for scenario reduction is summarized in Table 6.4. The scenario reduction scenario sets seem to be much more successful in solving the problem than sampling, recognized by the lower and more stable out-of-sample objective values.

Still, the reduction scenarios are not very stable in-sample either, inferred by the relatively large deviations in the in-sample expected recourse values across different scenario sizes. The deviation can be up to $8 - 10\%$ of the out-of-sample expected objective values. Thus, the bias is significant and also highly variable.

Scenario reduction, however, also seem to converge quite well out-of-sample. It converges faster

Table 6.4: Stability of scenario reduction.

K6				
Set size	$\rho = 0.67$		$\rho = 0.2$	
	In-sample	Out-of-sample	In-sample	Out-of-sample
S03	886.419	1741.458	1503.069	3493.849
S05	1891.029	1721.559	3695.687	3285.242
S10	1453.773	1720.377	2723.282	3275.846
S20	1921.877	1721.203	3735.994	3258.013
S30	1884.182	1720.080	3647.623	3253.018

K30				
Set size	$\rho = 0.67$		$\rho = 0.2$	
	In-sample	Out-of-sample	In-sample	Out-of-sample
S03	949.323	1921.340	1555.927	3786.156
S05	1573.296	1904.091	2963.303	3581.135
S10	1444.484	1902.571	2663.082	3571.502
S20	1618.764	1903.060	3015.574	3544.971
S30	1529.140	1902.486	2803.300	3539.716

than for sampling, to lower objective values which are at the lowest end, or below, what could be achieved by sampling. This means that they are quite successful in solving the problem.

For scenario reduction too, the difference in instability is noticeably larger for $\rho = 0.2$. At $\rho = 0.67$, the reduction scenario sets stabilise out-of-sample at set sizes of only five to ten, while for $\rho = 0.2$ it may not have entirely converged at scenario set size 30.

6.2.3 Recourse deviation

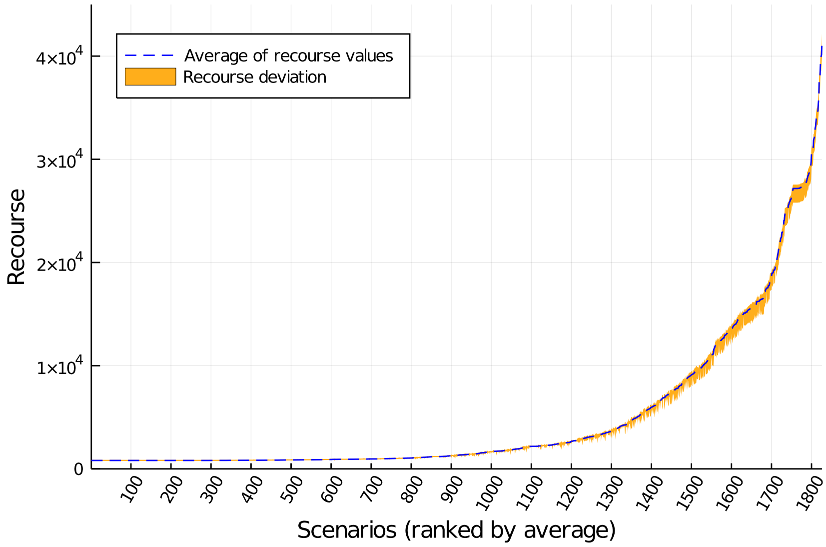
The empirical recourse deviation for the problem is illustrated in Figure 6.1. We see that a set of first-stage solutions obtained from solving IMSOAN with different sampled scenarios of size three give small variations in the output-distribution. Hence, the presence of uncertainty is relatively low for this problem. The fact that the recourse deviation is not extremely high also supports that the approximate solution set can be used to infer problem structures because it is unlikely that extreme changes in the output-distribution can occur for different first-stage solutions than those in our approximate solution set.

The noticeable difference in recourse deviation in changes of ρ in Figure 6.1b also confirm that the problem is more unstable for $\rho = 0.2$. A further interesting observation is that the deviation is larger for the middle part of the plots, meaning that the costs deviate more with changes in first-stage decisions for those scenarios. It may, therefore, be important to represent the output-distribution more accurately for this part of the output-distribution.

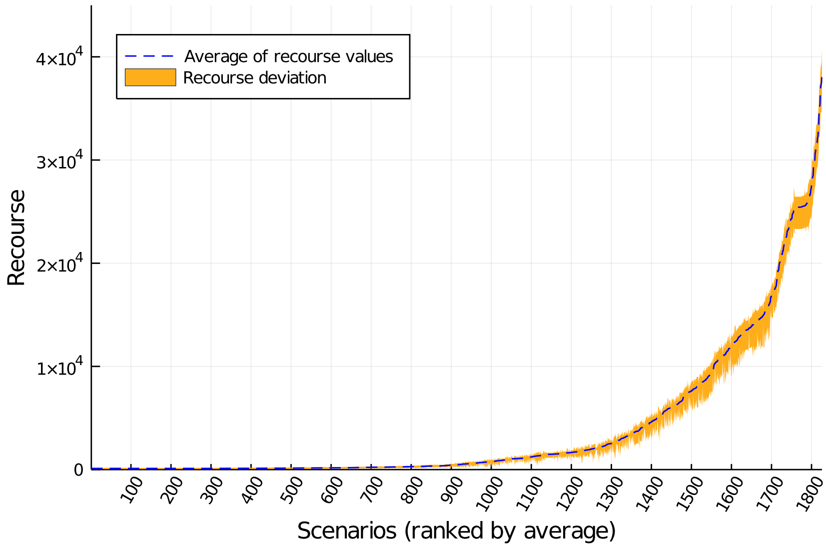
6.2.4 Discussion on stability

The most interesting observations from these results are that, first, the problem seems to be very stable out-of-sample, and second, that there is a large discrepancy between the in-sample and out-of-sample stability. The problem is regarded as stable because of its high out-of-sample stability, and the reason for this stability given the in-sample instability is a bit mysterious.

The high out-of-sample stability can be inferred from the results on both the reduction scenarios and the sampling scenarios. Even though the reduction scenarios perform much better than sampling, it is an uncommon observation that the stability is at the start of sufficient convergence at only sized 20 scenario sets. It seems like not too much would be needed for the scenario set to give stable results for the given problem; hence the presence of uncertainty is relatively low.



(a) $\rho = 0.67$



(b) $\rho = 0.2$

Figure 6.1: Recourse distribution for problem instance K6. The plot shows the out-of-sample distribution of ten first-stage solutions from solving the problem with ten sampled scenarios, and the recourse deviation is the span between the minimum and maximum recourse value for each scenario.

This does, however, not imply that the value of solving the stochastic formulation is low. The presence of uncertainty is high enough that a scenario set size of one would perform terribly, which can be recognized by the large discrepancy in solution quality between a set size of three and the larger ones. Extrapolating the change in quality to size one scenario sets, this would give very poor solution qualities. This was also confirmed by the high value of the stochastic solution (VSS) shown in (Wang & Jacquillat, 2020).

It is a finding in itself that this problem, for the given dataset, is inherently quite stable, and the analyses showing why can be considered a contribution in itself. To rationalized why this is the case, the airport network has a quite high number of connections between airports, meaning that the effects of weather changes at one place easily transfers to displacements in schedules at other airports. Hence, representing the entire range of possible outcomes at each airport at each time has a rapidly diminishing value. To summarize, the number of connections between the stochastic variables are high within the problem, affecting how we need to represent the uncertainty.

6.3 Comparison of scenario sets

The different distance metrics and clustering methods have been combined to see how they perform compared to each other. Since the configuration $\rho = 0.2$ was more unstable and thereby more interesting, the comparison of the new scenario sets is only for $\rho = 0.2$. The computational results are summarized in Table 6.5.

Table 6.5: Comparison of clustering scenario sets solved with $\rho = 0.2$. Objective evaluations for scenario generation procedure using each distance metric and clustering method for different set sizes. The suggested new scenario generation method using feature vector distances highlighted in green, scenario reduction as a benchmark highlighted in blue. "—" = memory issues.

K6						
	Out-of-sample					
Set size	k-median			k-means		
	Input	Output	Feature	Input	Output	Feature
S03	3493.849	3341.884	3310.128	3466.387	3327.264	3321.453
S05	3285.242	3317.381	3293.777	3464.289	3314.700	3320.001
S10	3275.846	3325.032	3275.387	3269.379	3311.480	3263.640
S20	3258.013	3281.324	3258.211	3263.602	3290.990	3262.025
S30	3253.018	3273.309	3257.017	3256.705	3269.304	3258.865
	In-sample					
Set size	k-median			k-means		
	Input	Output	Feature	Input	Output	Feature
S03	1503.069	2848.526	2686.065	1443.220	2332.781	2329.392
S05	3695.687	3047.413	2755.767	1739.846	2222.130	2388.633
S10	2723.282	3131.339	3049.227	2698.778	2504.057	2607.042
S20	3735.994	3220.478	3081.864	3577.062	2772.617	3360.884
S30	3647.623	3221.172	3112.911	3083.526	2947.899	3372.236
K30						
	Out-of-sample					
Set size	k-median			k-means		
	Input	Output	Feature	Input	Output	Feature
S03	3786.156	3625.248	3614.400	3812.708	3612.216	3612.215
S05	3581.135	3660.674	3621.525	3740.818	3598.023	3597.707
S10	3571.502	3570.456	3562.582	3580.736	3593.968	3593.663
S20	3544.971	3559.478	3548.314	3540.837	3573.667	3561.703
S30	3539.716	3556.408	3541.756	—	—	3545.366
	In-sample					
Set size	k-median			k-means		
	Input	Output	Feature	Input	Output	Feature
S03	1555.927	3059.009	3012.236	2206.539	2503.378	2503.378
S05	2963.303	3252.706	3210.983	1407.362	2433.186	2433.269
S10	2663.082	3438.243	3282.740	2179.492	2863.810	2799.795
S20	3015.574	3481.944	3352.507	2945.360	3163.188	3445.395
S30	2803.300	3496.734	3379.702	—	—	3463.471

Most interestingly, we see that the problem-based scenarios all improve on the out-of-sample score considerably for scenario sets of size three. This validates the rationale behind problem-based scenario generation showing that it can give useful corrections where distribution-based methods are insufficient. In this case, there was a 5% improvement in the solution quality, which is expected to be more significant for more unstable problems.

At larger sizes, the different scenario generation procedures behave more similarly. At the order of differences in out-of-sample objective values of 5 units, which is 0.15 %, it is a bit arbitrary to compare the different approaches as long as they are all within the same vicinity of each other without any outliers.

Another interesting observation from Table 6.5 is that the in-sample stability of the problem-based approaches is much better than for scenario reduction, as well as having a smaller and more consistent bias. This means that the proposed feature-distance based scenarios are representing the expected value of the output-distribution much more reliably. This didn't have a great effect for IMSOAN, but for other problems, this could have a considerable impact.

The purely output-distance based scenario sets perform worse than other approaches for larger scenario sets. This was expected by the previously given argument that clusters in the output-distribution may be highly non-connectible in the input domain. This observation validates the claim that proximity in the input domain must also be considered.

However, at lower scenario set sizes, the purely output-distance based scenario sets are correcting the distribution-based alternative considerably. This is a significant finding because the output-distance based scenario sets are solely considering the inferred problem-based properties and have no direct relation to the input-distribution. The fact that they make useful corrections, are more in-sample stable and have less bias has significant implications for the rationale behind using problem-based scenario generation.

Comparing k-means and k-median, it seems like k-median was often more effective for binary distributions, and this is attributed to the fact that the clusters in k-means were not formed with respect to the chosen representative outcomes. As k-means clustered outcomes does not have a very large discrepancy with k-median, and less so for larger scenario sets, we conclude that it can be a viable, less computationally demanding alternative to k-median for experimentations or benchmarking, and is likely more appropriate for larger scenario sets than for small ones.

6.3.1 Average corrected feature vectors

In Section 6.2.3 we saw that the low recourse deviation rendered that the problem is quite stable and that ranks of scenarios cannot change very much. Because of this, the effect from the third notion of proximity, proximity by similar recourse response, is not very strong compared to the other two. Therefore, new scenario sets have been generated by considering the average corrected feature vectors, expressed in Equation (5.13). The results are shown in Table 6.6.

Comparing the out-of-sample scores in Table 6.6, it is evident that this kind of scenario set performs strictly better than all other tested for this problem. This is attributed to the introduction of evaluation across sets of first-stage decisions to extract the problem-based properties that make for proximity by similar recourse response.

We see that bias is present, but compared to scenario reduction, the average corrected feature-distance based scenarios are much more consistent, thus, more in-sample stable as well. The use of bias correcting coefficients, as mentioned in Section 5.1, could be useful in such cases because of the consistency in the bias.

6.3.2 Explanatory capabilities of output-distributions

This section compares the different scenario sets in terms of output-distributions, showing how the output-distribution centric view of scenario generation can also explain why other scenario generation procedures don't work as well as the proposed new method.

Table 6.6: Average corrected feature distance by k-median clustering for $\rho = 0.2$. Benchmark is scenario reduction, also highlighted in blue in Table 6.5. The difference from the benchmark is highlighted in green.

K6			
Set size	Out-of-sample		In-sample objective
	Objective	Benchmark diff.	
S03	3305.795	−5.382%	3454.550
S05	3269.925	−0.466%	3317.953
S10	3259.181	−0.509%	3434.703
S20	3257.456	−0.017%	3515.791
S30	3256.699	+0.113%	3527.927
K30			
Set size	Out-of-sample		In-sample objective
	Objective	Benchmark diff.	
S03	3592.356	−5.119%	3241.894
S05	3582.372	+0.035%	3532.563
S10	3555.396	−0.451%	3441.910
S20	3541.440	−0.1%	3643.106
S30	3540.318	+0.017%	3556.436

Positioning of scenario sets in cumulative output-distributions and problem insight

Analysis of the output-distribution and the position of the scenarios within it illustrates why some scenario sets have performed better than others. Figure 6.2 shows the cumulative output-distribution from the sized three k-median clustered scenarios based on distances in the input domain (scenario reduction) and the feature vector domain. These are the ones with the most significant difference in out-of-sample expected values.

What is evident is that the scenario reduction scenarios are placed quite low, with two scenarios attaining very similar objective values. The problem-based scenarios are better dispersed in the upper end of the output-distribution, and we can see that the output-distribution has a reduced cost in the vicinity of those scenarios. This difference is what makes for the 5% improvement in the out-of-sample expected objective value.

The issue that comes up with scenario reduction here is that it considers all possible outcomes as equally important, while it is reasonably evident that almost half of them cause an equal amount of strain in the air traffic network. This means that it is reasonable to assume that all low impact scenarios can be represented by one outcome which covers the hedging property for all those. We see that the feature-distance based scenarios have done precisely this; only one scenario is placed in the low impact range, with a suitably high probability to correct for the statistical properties. This scenario has a probability of 81 %, meaning that one scenario accounts for 81 % of the input-distribution in this case. Scenario reduction, on the other hand, has placed two scenarios in the low range, meaning that only one scenario is left to represent the high impact range of outcomes, which is the essential range to describe in detail.

Figure 6.3 shows the same two scenario sets plotted in the input domain. We see that the problem-based scenarios in Figure 6.3a have made a distinction between the first airport and the others. That is because the weather at the first airport has a higher impact on the air traffic network, and we should consider its impact in particular. That the first airport has a high impact is also reflected in the correspondingly high cost for scenario no. 2, shown in Figure 6.2.

Another interesting observation is that for scenario no. 2 in Figure 6.3a, there are still some bad weather outcomes during the later time periods. That corresponds exactly with the observation in (Wang & Jacquillat, 2020) that strain from bad weather during the late hours of the day has a

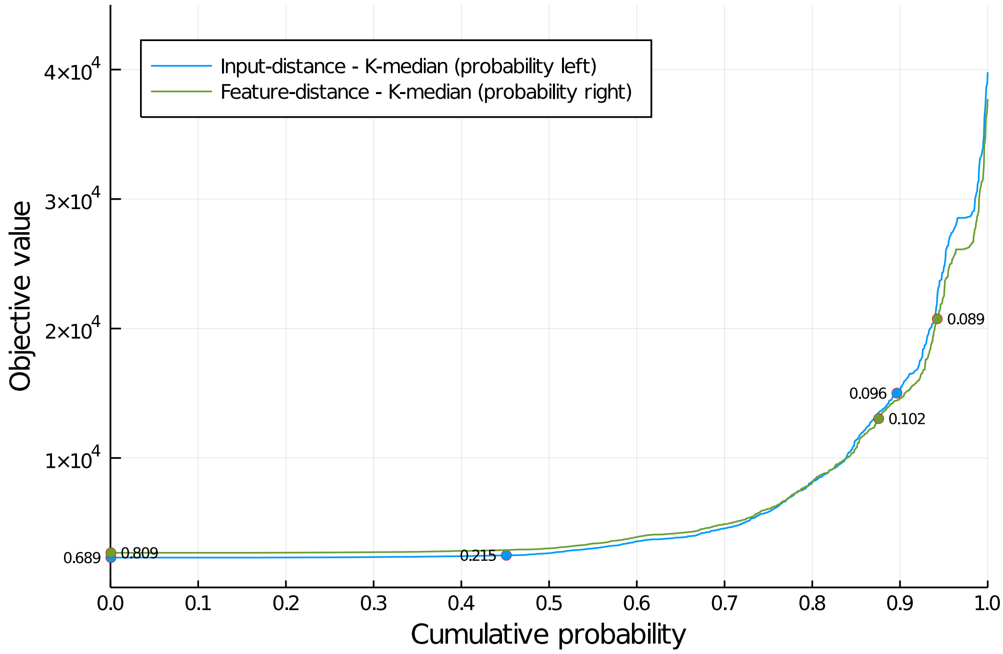


Figure 6.2: Comparison of scenario sets in-sample (dots with annotated probabilities) and out-of-sample (line).

higher impact on the total cost in the network. The problem-based scenarios reflect this well by including some bad weather outcomes in these time periods.

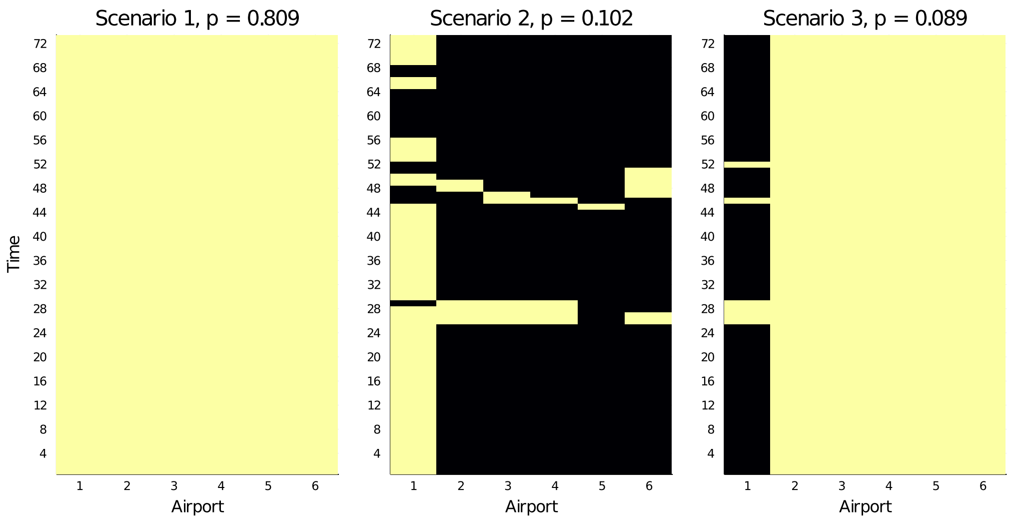
To compare, these kinds of problem insights cannot be found by analysing Figure 6.3b, which shows the reduction scenarios. These make a distinction between airport no. 5 and the others based on occurrences of bad weather, but that distinction is less relevant for the given problem because the corresponding increased cost, shown in Figure 6.2, is not very significantly different from the outcome with only good weather. Furthermore, airport no. 1, which we found out has a high impact on the airport network, has been represented only with good weather outcomes in the reduction scenario set, meaning that the scenario set lacks an essential property for the present problem.

Effectiveness of simple heuristics

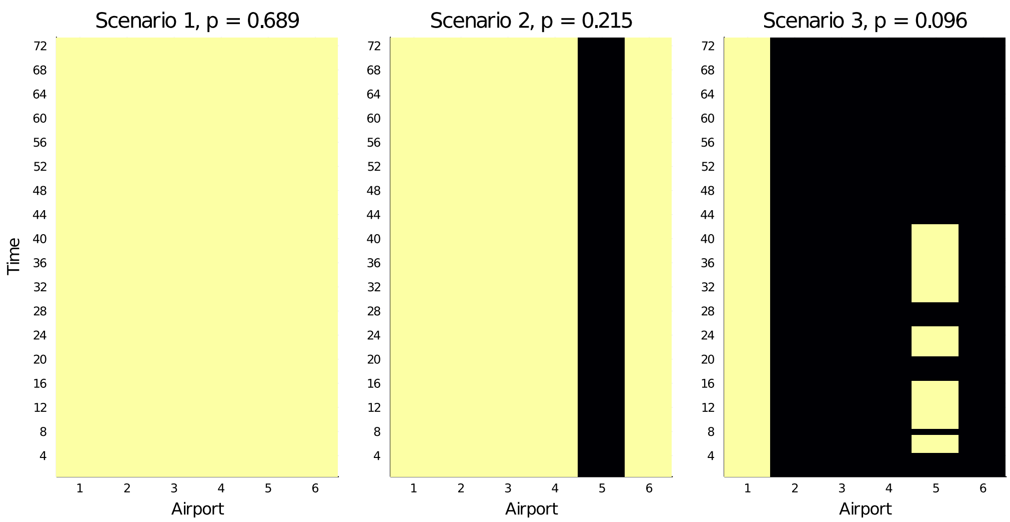
By the monotonicity property of IMSOAN we know that more bad weather results in higher costs, but we cannot know how the cost changes as the specific airports and times change with the same number of occurrences. Figure 6.4 illustrates that there is a very large spread in objective values for each number of occurrences of bad weather.

It was suggested previously that the monotonicity property of the problem could be exploited to construct a very simple scenario generation heuristic for the specific problem by partitioning the empirical distribution based on occurrences of bad weather. This was tested to be significantly less effective than scenario reduction in (Wang & Jacquillat, 2020). What is interesting is that analysing the objective output values highlights why this simple heuristic wasn't sufficient. The very significant difference between objective values for the same number of occurrences of bad weather means that the heuristic is more or less random in how it represents the output-distribution and will result in highly variable solution quality.

For the problem, this also explains that which airports and times bad weather occurs at has a great impact on the cost, which solidifies the rationale for why the problem should be solved stochastically.



(a) Feature vector distance generated scenarios



(b) Input distance generated scenarios

Figure 6.3: Comparison of scenario sets in the input domain for different generation methods. Black is bad weather, yellow is good weather. Probabilities for each scenario is annotated above.

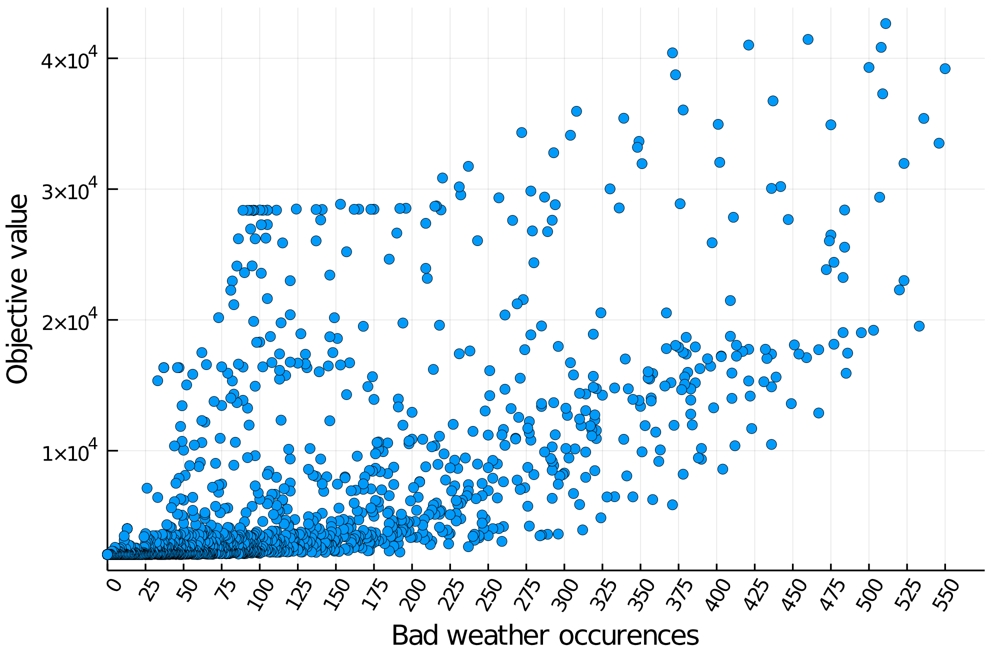


Figure 6.4: Objective values plotted against occurrences of bad weather. First-stage decision from solving with a sampled scenario set with $\rho = 0.2$.

7 Discussion and conclusion

This chapter starts with a discussion about the results and advancements from this thesis, and then we conclude the work and comment on potential future research.

7.1 Discussion

There are three interrelated main topics addressed in this thesis. First, a deep dive into the usefulness and purpose of problem-based scenario generation, second, mathematical and empirical stability theory of the impact of changes in the first-stage decisions on output-distributions, and third, binary distributions and how to generate scenario sets for them in specific. The main argument to address scenario generation with binary distributions is that problem-based scenario generation is especially useful for these problems, although this is valid for continuous input-distributions as well.

This section first discusses the proposed new scenario generation procedure and the results from applying it to the Air Traffic Flow Management problem. Next, we discuss how problem-based scenario generation can give modelling insights beyond providing more concise scenario sets to solve the problem. Then we discuss stability in stochastic programs and the implications of the added understanding of problem-based scenario generation, which follows from this thesis. Lastly, we discuss these results in light of generalized problem-based scenario generation, which has been an additional underlying research question for this thesis and the preceding project report (Narum, 2019).

7.1.1 Scenario generation using collections of output-distributions

This thesis has presented a new problem-based scenario generation method for stochastic problems with binary distributions. The results chapter showed that the proposed method was more accurate and more reliable than the alternative approach, scenario reduction. Out-of-sample stability was good, and for small-sized scenario sets the proposed procedure gave large improvements in solution quality. In-sample stability and bias were also significantly improved, which is promising for the applicability of the method to other problems.

It is acknowledged that problem-based scenario generation is often more laborious than distribution-based alternatives, but also more precise. When approaching a problem for the first time, it could often be wise to apply distribution-based approaches first and analyse the stability to determine the need for more precise problem-based methods.

The proposed scenario generation method requires a collection of output-distributions from the solutions of a set of naively generated scenario sets. If you have first tested your problem with distribution-based methods, that is already available to you, and the stability results can be reused to construct problem-based scenarios instead, meaning that the present approach adds a very marginal amount of additional work for added precision.

Another important property of the method is that the entire procedure can be done without solving the stochastic model for very large scenario sets. The collection of output-distributions can be computed from scenario sets of sizes which are reasonable to solve given the available computational capabilities. This thesis has shown that significant improvement on solution quality can be obtained without ever solving for larger scenario sets than the distribution-based alternative was too unstable with.

Compared to (Prochazka & Wallace, 2020), the proposed method in this thesis suggests that a smaller set of first-stage decisions is needed because analysis of the entire output-distribution gives more detailed insights than a correspondingly large set of expected values. Additionally, the use of k-means or k-median clustering methods circumvents the need for evaluating scenario sets in-sample within the scenario generation procedure, which was required in (Prochazka & Wallace, 2020). We suggested the binary point clustering method where in-sample evaluations would be necessary. We argued that IMSOAN had a too computationally demanding second-stage for this to work, which indicates that the fitting procedure of (Prochazka & Wallace, 2020) might not have been tractable for the considered case study problem either. Lastly, the fitting algorithm in (Prochazka & Wallace, 2020) and selection of clustering variables and accompanying testing in (Feng & Ryan, 2016; Sun et al., 2018) required considerable amounts of tailoring, while our proposed method can in principle be applied to any problem without considerable tuning or tailoring. To argue why this is reasonable, the advantage and significant difference lies in the evaluation of stability across first-stage decisions on the whole spectrum of available out-of-sample outcomes based on a restricted and more relevant set of first-stage decisions that the alternatives.

Performing out-of-sample evaluation of first-stage decisions can be computationally intensive, especially if the set of out-of-sample outcomes is very large or if the second-stage is very computationally demanding, as for IMSOAN. The suggested scenario generation procedure in this thesis requires out-of-sample evaluations, but we need not use the entire set of out-of-sample outcomes to infer problem structures. To decrease computation times, we could apply distribution-based scenario generation methods to construct a new empirical distribution for out-of-sample evaluations. As long as the newly generated empirical distribution is much richer than the scenario set, this is a perfectly sound approximation, and can significantly decrease computational times.

A direct consequence of making a new input-distribution to extract problem-based properties from is that we can employ a large set of already existing distribution-based scenario generation methods as a preprocessing step which specifically addresses other challenges than we have addressed directly in this thesis. We could incorporate notions of higher dependence structures by copula sampling (Kaut, 2014; Kaut & Wallace, 2011) and aggregation sampling for tail risk measures (Fairbrother et al., 2019). Choosing an appropriate distribution-based method for reducing the size of the input-distribution for out-of-sample evaluation is important not to lose important characteristics of the original input-distribution. More generally, we recommend property matching for continuous input domains with strong correlations (Høyland et al., 2003) and scenario reduction for discrete input domains (Dupačová et al., 2003).

In applications, it can often 'feel safer' for the modeller to make scenario sets of outcomes which are already part of the historical dataset. In this thesis, we argue explicitly against that. Likely, the historical data is not nearly rich enough to contain the outcomes that make for the best possible scenario set due to high dimensionality. The use of 'non-real' data points as a means to reach better solutions should be seen as a necessity to get the best possible scenario sets because the behaviour of both the distribution and the problem is captured better. As long as we do out-of-sample testing of the solutions, it is entirely irrelevant which tools we have used to construct them.

The second-stage of IMSOAN was too computationally demanding to incorporate in-sample objective evaluations into a heuristic procedure to solve binary point centroid clustering, which would illustrate the value of using the entire binary input domain for scenarios. Instead, k-means clustering became the test case to highlight this value. The conclusion from the results section was that k-means was worse than k-median, but not considerably, mostly attributed to the inappropriate selection of representative outcomes from the fractional centre points from k-means. We can therefore not make conclusive arguments about the usefulness of making use of the entire input domain. For a problem with a simpler second-stage, a binary point centroid clustering approach would be interesting to implement within the presented framework in this thesis, which is expected to be more effective.

The Air Traffic Flow Management was more stable in nature than preferred, which meant that the distinctions between the different scenario generation procedures weren't always obvious. This does not invalidate that the new method works well but makes it is a bit more difficult to see.

The proposed new scenario generation procedure improved the solutions significantly for small scenario sets, and at no point compromised on the quality of solutions for larger scenario sets. The limitation is that we cannot determine how effective it could be for more unstable problems where other approaches might not be sufficient to get reliable results. Based on the rationale underlying the analysis of output-distributions across the approximate solution set, we expect that it could make highly effective corrections. This thesis' supporting evidence for this is that the clustering scenarios based only on the output domain were comparably effective to all other approaches, and also gave better results than scenario reduction for scenario sets of size three. This is a significant result because the scenario set has proved effective based on a method which is agnostic to the specific kind of problem but can still incorporate problem-based properties into the scenario generation. The reason application-specific corrections haven't been made in this work is that we aimed for a method which can be useful on any problem class without tailoring.

Consider that all implemented scenario generation procedures gave very similar solutions for larger scenario sets. It is not unprecedented to consider whether we are reaching a lower bound on the possible solution quality for a given scenario sets size. It is known that stochastic programs require more than one scenario; otherwise, there is no way to invest in flexibility since no options exist (Wallace, 2010). No literature addresses such a lower bound for more than one scenario, and this is likely because it would be problem- or even instance-specific. Problem-based scenario generation is a framework which may lead to advancements in formulating estimates of such lower bounds, but for now, we cannot confirm nor deny whether we have reached such a bound on solution quality.

The monotonicity property of IMSOAN supports that changes in the output-distribution with different first-stage solutions would be relatively stable, and a low recourse deviation also confirmed this. Considering that binary stochastic variables often represents the presence of a customer, failure of a link or similar, it's inferred that this monotonicity property should be found in a broad set of problems specified by binary distributions, which argues that *other problems specified by binary distributions could exhibit similar problem behaviour to the case study problem in this thesis*, generalizing the implications of this work.

7.1.2 Modeling insights from output-distributions

A presupposition when we model a real-world problem by a mathematical formulation is that, hopefully, what occurs in the model would also happen in the real world, at least within the bounds drawn around the problem when the model was formulated. In the realm of problems so complicated that humans have no chance of apprehending its complexities, interpretation of models is vital when it supports decision making. They are not only crucial for insight beyond the optimal solution in itself but also for verifying that the model reflects the real-world problem to a satisfactory degree by sanity checks.

A stable set of scenarios can be interpreted as a set of uncertain outcomes of the future that are sufficient to consider for a given decision problem. The practical implication of this is that for a complicated decision problem with an incomprehensible amount of potential outcomes, we can distil the uncertainty down to a comprehensible amount and infer why the uncertainty matters for the given application. Thus, problem-based scenario generation can not only help us solve intractable problems by using smaller scenario sets, but they may also help us interpret the impact of uncertainty for our real-world application and understand why uncertainty has an impact at all. This was exemplified for IMSOAN in Section 6.3.2 with the analysis accompanying Figure 6.3.

Stochastic programming is an optimization of distributions and should be analysed as such. Solving a stochastic model without looking at the output-distribution and how it changes with first-stage decisions is comparable to not checking the stability of your scenario sets; you have no idea what is going on and if it works as you wish. The recourse deviation is a better tool for stability evaluation because it gives more information than simply the continuum of stable – non-stable. It can tell you what kind of outcomes are more or less stable. What is especially interesting about the results from this thesis is that the output-distribution centric view of scenario generation also gives explanatory power to argue why the other scenario generation methods did or didn't give reliable results. This was also exemplified in Section 6.3.2 by the analysis accompanying Figure 6.2 and Figure 6.4.

In principle, the proposed scenario generation method in this thesis is independent of the utility metric that is used to formulate the problem. The utility metric itself could, for example, be decided by analysing the output-distribution. Finding that the upper tail is heavy and shifts significantly with first-stage decisions might mean that tail risk measures is a reasonable utility metric, motivated by application-specific reasons for why that is not good. If the output-distribution is mostly limited within a span of recourse values but shifts its mass for different first-stage decisions, it might be wise to use the expected value as a utility metric.

The presence of uncertainty in a specific problem, which is equivalent to the amount of instability, is a metric of how necessary it is to solve the given problem by its stochastic formulation. Additionally, it determines how many scenarios we may need to solve the problem reliably or how well constructed they need to be. A problem with a low presence of uncertainty could require almost nothing of the scenario generation procedure, or could even be solved sufficiently well by the deterministic version of the problem. By this distinction, the tools within problem-based scenario generation can also help to determine the degree to which a problem would need to be solved by its stochastic formulation. The notion of lower bounds on the required number of scenarios, or solution quality for a given scenario set size, is also related and problem-based scenario generation could also bring us closer to insights into the existence of such lower bounds.

7.1.3 Stability driving tractability of scenario generation

A crucial observation is that the scenario sets we aim to generate are of sizes at the order of three, five and ten to represent a binary input-distribution of 400 to 2000 dimensions. At this outstanding level of simplification of the empirical distribution, the statistical accuracy of the scenario set's representation of the input-distribution should be expected to be very low. Furthermore, as this new representation of the empirical distribution goes through as complicated a transformation as a two-stage model, any representability in the input-domain transferred to the output-domain usually is even worse. Still, the results in this thesis show that it can clearly be done.

The successfulness of scenario generation, in general, cannot only be explained by an extraordinarily good representation of the input-distribution but rather that the stochastic programming model is stable enough to give reliable results for a large variety of representations. This means that *stability properties of the problem formulation are what drives the tractability of scenario generation.* The theory supporting this follows from implications of the results in (Römisch & Wets, 2007a), although this extraordinary observation of the properties of stochastic programs cannot be entirely described by that result either. It is primarily an empirically observed property.

Given that stability of stochastic models determines the successfulness of scenario generation, there's no doubt exploring the properties of problems to guide scenario generation is crucial, and this was the motivation for the onset of this thesis. We have, in this thesis, explored more deeply how the problem structures of a model formulation can be found, namely by examining commonalities and differences within a collection output-distributions resulting from a restricted set of first-stage decisions.

Intuitively, the suggested new scenario generation method works because analysing collections of output-distribution is in practice a dimension reduction of the huge input domain of the stochastic variables, where the input-distribution is projected onto the problem formulation for a given first-stage decision by out-of-sample evaluation. However, the projection can be made for each feasible first-stage decision, which for continuous problems is infinite-dimensional or for integer programs very large dimensional. By using approximate solution sets, the number of potentially considered output-distributions is significantly reduced, although, still very large. The next crucial property we rely on is that most problem formulations we deal with exhibit reasonably stable results with perturbations in the scenario sets used to solve the problem, meaning that the differences in output-distributions are also relatively small. This property of stability is supported by implications of (Römisch & Wets, 2007a), and by the fact that we can empirically verify the stability for the specific problem. The recourse deviation indicates the distinctiveness of different output-distributions, also referred to previously as the presence of uncertainty in a given problem.

We have argued that analysing problem behaviour with respect to changes in first-stage decisions is most important, and for the rest of this section, we discuss more deeply why. Analysing the output-distribution with respect to changes in the first-stage decision is much more essential than merely perturbing the recourse function in 'the other' argument besides the stochastic variables. The first-stage decision is indeed the only thing that matters when solving a stochastic model, while the stochastic representation and the second-stage are simply a means to model the utility of that first-stage decision in a way that's tractable to solve.

Furthermore, we argue that the exact representation of the whole output-distribution is also mostly not interesting. *When representing the output-distribution by a scenario set, the only thing that matters are the qualitative changes in the recourse function with respect to the stochastic variables as we change the first-stage decision.* The results in this thesis and the empirically observed fact that stochastic programs are exceptionally stable can be inferred to mean that the number of such qualitative distinctions are relatively few.

Efforts have been made to bound the recourse function or to approximate it more and more accurately, for example by quadrature rules, Quasi-Monte Carlo sampling, or similar methods. These are missing the point of what needs to be approximated, and a simple thought experiment can make it clear why: If it is vital to represent the entire output-distribution accurately, we would not be able to solve stochastic programs at all. The substantial empirical stability results of stochastic programs would be not nearly as good if the solutions relied on very accurate descriptions of the output-distributions because many scenario generation procedures are nowhere near approximating the entire distribution accurately but still perform perfectly well. Hence, something else makes for these exceptional stability results, and we should exploit those properties instead.

Representing everything more accurately is always a solution, but not a very good one when tractability is one of the prime issues we face.

Let us describe more precisely what we mean by qualitative distinctions in the output-distribution. The following arguments are a hypothesis for how changing output-distribution can be understood, motivated by the gained insight from working with this thesis.

A qualitative distinction of changes in output-distributions with respect to the first-stage decision is distinguished by different modes of change, namely, expressed by a set of primary basis functions with range in the output domain and support in the input domain. We hypothesise that first-stage decisions in the vicinity of the approximate solution set results in changes in the output-distribution which can primarily be described by the components of such basis functions.

We do not know the form of these basis functions, and they are likely problem-dependent, but we can infer them by the collection of output-distributions resulting from the approximate solution set. The critical insight that is concluded by the empirically observed stability property of stochastic programs is that we need only a small set of basis functions to describe most of the variations in the output-distribution.

After identifying the primary basis functions, we formulate a scenario set, residing in the support of the basis functions, to cover the main modes of change, which is good enough to distinguish first-stage decisions apart in terms of solution quality. Doing this may be non-trivial, but it is fair to assume that it can be done with not too many more scenarios than there are basis functions. Furthermore, this would infer a lower bound on how many scenarios you need to represent changes in the output-distribution sufficiently.

The primary result of this thesis implies that there exists a very limited set of primary basis functions in the output domain and that they are very much tractable, but maybe non-trivial, to find. The basis decomposition is a more concise and precise interpretation of what we have done in this thesis and is a promising direction to develop problem-based scenario generation further.

7.1.4 Generalized problem-based scenario generation

One of the aims from the outset of this thesis was to figure out *what it is* that the 'black box' fitting in (Prochazka & Wallace, 2020) finds. Especially interestingly, the authors highlighted that their fitting algorithm could replicate the findings that applied to scenario generation for tail risk measures in (Fairbrother et al., 2019), meaning that there exists a common pattern their fitting algorithm is able to find.

The research question which emerged from this was whether it's possible to solve *generalized problem-based scenario generation*, which we define to be a scenario generation method which can exploit problem specific properties without making assumptions on what these properties might be, named an *agnostic method*, and at the same time does not rely on solving the problem first, referred as the capability of doing *scenario generation ex ante*.

The implication of finding a generalized problem-based scenario generation method is that it could be used across any problem class and would be more effective than any distribution-based alternative. An essential assumption is that the method scales well and can be used for large-scale and combinatorial stochastic problems, which is why the ex ante property is important. The end goal is to be able to solve stochastic problems of a scale that are otherwise intractable.

The goal of the project report (Narum, 2019) was to see if the properties of a two-stage stochastic linear program with respect to its stochastic variables could give useful insights so that a problem-based scenario set could be generated directly from the mathematical formulation of the problem itself. The goal was to see if that was possible, which is why it was simplified to linear programs with a stochastic right-hand side (or stochastic objective).

It was established that the input domain of the distribution could be decomposed into a finite set of partitions with respect to the problem which each represents unique problem behaviours by differences in linearity. The caveat was that the granularity of this decomposition was far too fine and with too high computational complexity to be found explicitly (even by more pragmatic empirical approaches). It could therefore not be used ex ante.

What is important is how the partitioning changes with different first-stage decisions, and one of the most interesting takeaways from (Narum, 2019) was that the partitioning of the input-distribution with respect to the problem has a correspondence with extreme points in the dual feasible set of the formulation. Moreover, the dual feasible set is independent of the first-stage decision and can be directly related to the output-distribution since larger dual variable values imply increased objective values. Thus, distinctions between outcomes of the stochastic variables with respect to the problem formulation can be made by extreme point enumeration of the second-stage's dual feasible set. Note that when dual extreme points are linked with specific stochastic outcomes, the first-stage decisions must also be included. Lastly, such extreme points will have notions of proximity in the dual space which can be utilized to distinguish them further apart to reach a greater span of difference in the behaviour of the second-stage formulation. Note that these results may also hold for integer programs by relaxing them to a linear formulation. It is not evident that this would invalidate the problem-based distinctions made about the stochastic variables.

The conclusion from (Narum, 2019) was that problem behaviours can be inferred by exploring the second-stage formulation, but that we are facing high computational complexity and need to know what we want to find.

This thesis aimed to explore how first-stage decisions change output-distributions and which qualities in the output-distributions are important to capture in a scenario set. We pivoted from a bottom-up approach to a top-down approach, which also involved a higher acceptance for fundamentally approximate explorations of qualitative distinctions in problem behaviour.

This thesis answered the question of whether a problem-based scenario generation method which is agnostic to the specific problem is possible. Additionally, the suggested metric for determining discrepancy between the out-of-sample outcomes and a scenario set shows great promise, and it can be explored further how this can be inferred without solving the two-stage model itself.

Combining the conclusions from this thesis and (Narum, 2019) there is a path forward for how the result from this thesis on a problem-based agnostic metric of proximity between distributions can be expanded by not relying on an out-of-sample evaluation of each first-stage decision in the approximate solution set. By exploration of the dual feasible set of the linear relaxation of the second-stage, we could infer qualitative distinctions between output-distributions without explicitly solving the second-stage. Exactly how these two approaches could be combined is a question for further research. Such an effort would be more valuable for problems with integer second-stage which are difficult to solve, but it is not completely clear how well properties from the linear relaxation transfers to the integer formulation in those cases.

In a longer time-horizon, it could also be explored whether we need to explicitly solve the stochastic program with naive scenario sets to get properties of the approximate solution set or if this can be inferred ex ante of solving the stochastic model. The way forward on this is, however, not very clearly defined.

7.2 Conclusion

This thesis concludes that a collection of output-distributions obtained from a restricted set of relevant first-stage decisions can be analysed to infer the structure of stochastic problems which in turn can guide scenario generation to construct more concise scenario sets to solve stochastic problems than possible by distribution-based methods. These stability properties are complicated, difficult to find and have not previously been interpreted in previous literature. The proposed approach to attain them is computationally demanding but very much tractable.

The presented scenario generation method is effective enough to guide scenario generation for binary input-distributions. It is, by numerical experiments, shown to outperform every other alternative approach to scenario generation with binary distributions on a large-scale, combinatorial and stochastic case study problem in Air Traffic Flow Management. The monotonicity property present in many stochastic problems with binary distributions generalizes the implications of this result to other problems.

7.3 Future research

The most relevant continuation of the work in this thesis would be to explore further whether approximate solution sets or similar structures, which were shown to contain sufficient information on stability properties, can be inferred ex ante of solving the stochastic model.

Another interesting continuation would be to expand the distribution centric view of scenario generation to multi-stage programs, which are often more complicated and require more sophisticated approximations. The challenge then is to transfer the notion of an output-distribution when there are multiple consecutive stages, all with interdependent decision variables and transitions of uncertainty. Being able to infer approximate solution sets ex ante might be a valuable stepping stone for multi-stage programs because they, in principle, are recursive two-stage problems and being able to infer solutions to decision variables at the intermediate stages could be very valuable.

This thesis has been a deep dive into scenario generation considering problem structure, but some problems have very important dependence structures where the solution procedure immediately exploits 'false' negative correlations appearing in the scenario set. An interesting topic of future research would be to explore how to combine scenario generation for significant dependence structures together with very high exploitation of the problem structure. A starting point could be to use distribution-based scenario generation as a preprocessing step as laid out in Section 7.1.1, whose effectiveness for higher dependence structures or strong correlations would first have to be tested numerically on a practical problem.

Bibliography

Ball, M. O., Colbourn, C. J. & Provan, J. S. (1995). Network reliability. *Handbooks in operations research and management science*, *7*, 673–762.

Ben-Tal, A., El Ghaoui, L. & Nemirovski, A. (2009). *Robust optimization*. Princeton University Press.

Bertsimas, D., Lulli, G. & Odoni, A. (2011). An integer optimization approach to large-scale air traffic flow management. *Operations research*, *59*(1), 211–227.

Bertsimas, D. & Patterson, S. S. (1998). The air traffic flow management problem with enroute capacities. *Operations research*, *46*(3), 406–422.

Birge, J. R. & Louveaux, F. (2011). *Introduction to stochastic programming*. Springer Science & Business Media.

Chen, Z. & Yan, Z. (2018). Scenario tree reduction methods through clustering nodes. *Computers & Chemical Engineering*, *109*, 96–111.

Crama, Y. & Rodríguez-Heck, E. (2017). A class of valid inequalities for multilinear 0–1 optimization problems. *Discrete Optimization*, *25*, 28–47.

Dunning, I., Huchette, J. & Lubin, M. (2017). Jump: A modeling language for mathematical optimization. *SIAM Review*, *59*(2), 295–320. doi:10.1137/15M1020575

Dupačová, J., Gröwe-Kuska, N. & Römisch, W. (2003). Scenario reduction in stochastic programming. *Mathematical programming*, *95*(3), 493–511.

Fairbrother, J., Turner, A. & Wallace, S. W. (2018). Scenario generation for single-period portfolio selection problems with tail risk measures: Coping with high dimensions and integer variables. *INFORMS Journal on Computing*, *30*(3), 472–491.

Fairbrother, J., Turner, A. & Wallace, S. W. (2019). Problem-driven scenario generation: An analytical approach for stochastic programs with tail risk measure. *Mathematical Programming*, 1–42.

Feng, Y. & Ryan, S. M. (2016). Solution sensitivity-based scenario reduction for stochastic unit commitment. *Computational Management Science*, *13*(1), 29–62.

Guo, Z., Wallace, S. W. & Kaut, M. (2019). Vehicle routing with space-and time-correlated stochastic travel times: Evaluating the objective function. *INFORMS Journal on Computing*, *31*(4), 654–670.

Heitsch, H. & Römisch, W. (2003). Scenario reduction algorithms in stochastic programming. *Computational optimization and applications*, *24*(2-3), 187–206.

Heitsch, H. & Römisch, W. (2007). A note on scenario reduction for two-stage stochastic programs. *Operations Research Letters*, *35*(6), 731–738.

Henrion, R. & Römisch, W. (2018). Problem-based optimal scenario generation and reduction in stochastic programming. *Mathematical Programming*, 1–23.

Høyland, K., Kaut, M. & Wallace, S. W. (2003). A heuristic for moment-matching scenario generation. *Computational optimization and applications*, *24*(2-3), 169–185.

Kall, P. & Wallace, S. W. [Stein W.]. (1994). *Stochastic programming*. Springer.

Kaut, M. (2014). A copula-based heuristic for scenario generation. *Computational Management Science*, *11*(4), 503–516.

Kaut, M. & Wallace, S. W. [Stein W]. (2007). Evaluation of scenario-generation methods for stochastic programming. *Pacific Journal of Optimization*, *3*(2), 257–271.

Kaut, M. & Wallace, S. W. [Stein W]. (2011). Shape-based scenario generation using copulas. *Computational Management Science*, *8*(1-2), 181–199.

King, A. J. & Wallace, S. W. [Stein W]. (2012). *Modeling with stochastic programming*. Springer Science & Business Media.

Latorre, J. M., Cerisola, S. & Ramos, A. (2007). Clustering algorithms for scenario tree generation: Application to natural hydro inflows. *European Journal of Operational Research*, *181*(3), 1339–1353.

Löhndorf, N. (2016). An empirical analysis of scenario generation methods for stochastic optimization. *European Journal of Operational Research*, *255*(1), 121–132.

Narum, B. S. (2019). Problem-based scenario generation in stochastic linear programming.

Prochazka, V. & Wallace, S. W. [Stein W]. (2018). Stochastic programs with binary distributions: Structural properties of scenario trees and algorithms. *Computational Management Science*, *15*(3-4), 397–410.

Prochazka, V. & Wallace, S. W. [Stein W.]. (2020). Scenario tree construction driven by heuristic solutions of the optimization problem. *Computational Management Science*. doi:https://doi.org/10.1007/s10287-020-00369-2

Rachev, S. T. & Rüschendorf, L. (1998). *Mass transportation problems: Volume i: Theory*. Springer Science & Business Media.

Rachev, S. T. & Rüschendorf, L. (2006). *Mass transportation problems: Applications*. Springer Science & Business Media.

Rockafellar, R. T. & Wets, R. J.-B. [Roger J-B]. (2009). *Variational analysis*. Springer Science & Business Media.

Römisch, W. & Wets, R.-B. (2007a). Stability of ε-approximate solutions to convex stochastic programs. *SIAM Journal on Optimization*, *18*(3), 961–979.

Römisch, W. & Wets, R.-B. (2007b). Stability of ε-approximate solutions to convex stochastic programs. *SIAM Journal on Optimization*, *18*(3), 961–979.

Sen, S. (2019). *Pre-conference tutorials*. International Conference of Stochastic Programming.

Shapiro, A. (2003). Monte carlo sampling methods. *Handbooks in operations research and management science*, *10*, 353–425.

Sun, M., Teng, F., Konstantelos, I. & Strbac, G. (2018). An objective-based scenario selection method for transmission network expansion planning with multivariate stochasticity in load and renewable energy sources. *Energy*, *145*, 871–885.

Villani, C. (2008). *Optimal transport: Old and new*. Springer Science & Business Media.

Wallace, S. W. [Stein W]. (2010). Stochastic programming and the option of doing it differently. *Annals of Operations Research*, *177*(1), 3–8.

Wang, K. & Jacquillat, A. (2020). *A stochastic integer programming approach to air traffic scheduling and operations*. Accepted for *Operations Research*.

Wets, R. J. (1996). Challenges in stochastic programming. *Mathematical Programming*, *75*(2), 115–135.

Zhao, Y. & Wallace, S. W. [Stein W]. (2016). Appraising redundancy in facility layout. *International Journal of Production Research*, *54*(3), 665–679.

NTNU

Norwegian University of
Science and Technology