The School of Mathematics

THE UNIVERSITY of EDINBURGH

Scenario Generation and Reduction for Stochastic Investment Planning

by
Thomas Märki

Dissertation Presented for the Degree of
MSc in Operational Research with Data Science

Supervised by
Professor Ken McKinnon

August 2019

Abstract

Every year, the global energy industry invests massively into new power generation units. In order to align these investments with future needs and challenges, planning how much to invest, when to invest and in what technology to invest is a significant task.

The field of Stochastic Investment Planning is concerned with finding an optimal investment policy in the presence of uncertainties such as future energy demand or emission limits. When these uncertainties are discretised, a scenario tree is a useful tool to visualise and model the problem. However, with finer discretisation, more uncertainties and more stages, the tree grows exponentially and solving the problem can become intractable. One approach to deal with this challenge is to adaptively expand and reduce the scenario tree during the solution process. In that way, relevant scenarios that drive the investment decisions are identified and used in the solution, while scenarios that do not make a difference are excluded. In this thesis, a method that assesses the gradients with respect to the investment decisions at different nodes of the tree is developed. Comparing these value surfaces makes it possible to identify similarities between scenarios.

The proposed method is tested on a large-scale investment planning problem, which is solved with a novel Benders decomposition algorithm. In two evaluated cases, the method is successfully applied to construct significantly reduced scenario trees, while maintaining the investment policies of the original trees. In a third case, the same method is applied to expand a deterministic tree by adding new scenarios. The results are encouraging and highlight the potential of the method, nevertheless, more research is needed in order to automate the generation and reduction of scenarios.

Acknowledgments

I would like to express my sincere gratitude to Professor Ken McKinnon, whose guidance along the way has been extremely valuable. He always gave me the feeling of being part of a research team, rather than a lonely student spending the summer in the library. It has been truly inspiring to work with him and to benefit from his experience.

In addition, I would like to thank Dr. Nicolò Mazzi for his invaluable support during the project. His research has been an inspiration and I have found a lot of joy in studying and learning from it.

I feel grateful for having had the opportunity to work with both of you.

Own Work Declaration

I declare that this thesis was composed by myself and that the work contained therein is my own, except where explicitly stated otherwise in the text.

Thomas Märki

Contents

1 Introduction **1**

2 Literature Review **4**

2.1 Stochastic Programming 4

2.1.1 Scenario Trees 5

2.1.2 Scenario Generation Methods 5

2.1.3 Scenario Reduction Methods 6

2.2 Benders Decomposition 8

2.2.1 Benders Decomposition with Adaptive Oracles 9

3 Case Study **10**

3.1 Approach 10

3.2 Investment Planning Model 11

3.2.1 Solving the Model 13

3.2.2 Assessing the Dual Variables 14

3.3 Scenario Reduction 18

3.3.1 Uncertainty in $c_i^{co_2}$ and v_i^E 18

3.3.2 Uncertainty in v_i^D and v_i^E 20

3.4 Scenario Generation 21

3.4.1 Uncertainty in $c_i^{co_2}$ and v_i^E 22

3.5 Review of Approach 23

4 Conclusions **26**

References **28**

Appendices **30**

A Operational Model: Mathematical Formulation **30**

B Operational Model: Technologies **32**

C Gradients of Original Inventory Constraint **33**

D Gradients of new Inventory Constraints **34**

E Case 1: Scenario Tree **36**

F Case 1: Reduced Scenario Tree **37**

G Case 1: Investment Decisions **38**

H Case 1: Gradients w.r.t. Investment Decisions **41**

I Case 2: Scenario Tree **44**

J Case 2: Reduced Scenario Tree **45**

K Case 2: Installed Capacity **46**

L Case 2: Installed Capacity in Reduced Tree **47**

M Case 3: Expanding the Deterministic Tree **48**

N Case 3: Gradients w.r.t Investment Decisions **50**

List of Tables

1 Set of $\mathcal{P}$ technologies 11
2 Gradients at iteration 0 17
3 Case 1: Newly installed capacity (GW) 18
4 Case 1: Gradients w.r.t. investment decisions 19
5 Reduced case 1: Newly installed capacity (GW) 20
6 Case 1: Comparison of computation time 20
7 Case 2: Comparison of computation time 21

List of Figures

1 Two-stage scenario tree with one uncertainty parameter . . . 2
2 Convergence of upper and lower bounds 14
3 Deterministic scenario tree 21
4 Expanding a deterministic scenario tree 23

List of Acronyms

IEA International Energy Agency.

SP Stochastic Programming.

EVPI Expected Value of Perfect Information.

OP Operational Problem.

MP Master Problem.

RMP Relaxed Master Problem.

CCGT Combined Cycle Gas Turbine.

OCGT Open Cycle Gas Turbine.

1 Introduction

In 2018, global power generation investment totalled USD 485 billion. While low-carbon technologies, such as solar photovoltaics, wind power, hydropower and nuclear, comprised almost three-quarters of generation spending, investment in coal- and gas-fired power generation slowed. Despite the transition to environmental-friendly technologies, the International Energy Agency (IEA) suggests that the current investment in power generation is poorly aligned with future needs and challenges. Depending on the depicted scenario, the total investment is between 15% and 35% less than the annual projected needs in order to meet the energy security and sustainability goals. Moreover, the IEA states that investment in gas power, renewables in general and battery storage are misaligned with what their models had predicted. [1, 2]

Planning how much to invest, when to invest and in what technology to invest are crucial questions for international organisations, governments and power generation companies. Decision-makers often rely on mathematical models that aim to solve such investment planning problems. These models can either be deterministic, that is model parameters such as future demand, oil price or CO_2 emission limit are known, or stochastic with uncertainty in some parameters. The earliest use of models that include uncertainty dates back to Massé [3] (as cited in [4]), who was looking for ways to optimally schedule hydro power plants. He realised that using deterministic parameters is often far too optimistic, because odd and special situations are automatically excluded and only expected values are considered. He came to conclude that deterministic solutions will underestimate the true cost of uncertainty that results - in the case of hydro power plants - from the risk of spilling water or from long draughts. The field of research that deals with finding optimal decisions for problems involving uncertain parameters became known as Stochastic Programming (SP) [5]. While stochastic refers to the element of uncertainty, programming means that parts of the problem can be modelled as mathematical programs. Models that take uncertainty into account are commonly used for investment planning problems and are the subject of this thesis.

In an ideal world, uncertainty parameters, for example future energy demand, would be represented by a continuous distribution. However, limited computing power coupled with the complexity of the decision model mean that such problems cannot be solved and the distributions of the stochastic parameters have to be approximated by discrete distributions with a limited number of outcomes [6]. This discretisation is usually represented with a scenario tree, as depicted in Figure 1. The figure shows a scenario tree with one uncertainty parameter, three possible outcomes and two investment stages. The result is a tree with nine possible scenarios.

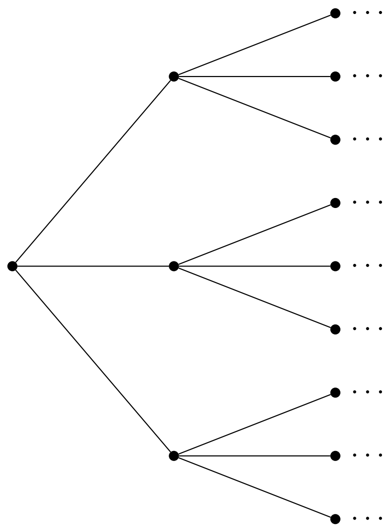


Figure 1: Two-stage scenario tree with one uncertainty parameter

With more uncertainty parameters and more decision stages, the scenario tree grows exponentially. How to discretise the uncertainty (that is how to generate scenarios that accurately represent the uncertainty) is one of the challenges in SP. While fine discretisation may lead to problems that are numerically impossible to solve, coarse discretisation may lead to unrepresentative models that overlook possible scenarios.

Each individual scenario is an optimisation problem in itself that is dependent on decisions in previous stages. These so-called subproblems share the same mathematical composition and differ only in terms of the uncertainty parameter(s). This kind of block structure naturally lends itself to the use of decomposition techniques such as Benders decomposition [7]. When Benders decomposition is applied to investment planning problems, the problem is divided into a first-stage master problem (the investment problem) and a sequence of subproblems that represent the scenarios. These subproblems are solved independently over a given first-stage solution in order to generate so-called Benders cuts that can be added to the master problem. This iterative process is repeated until a certain convergence limit is reached or until no more cuts can be generated and the problem is solved to optimality. Since new constraints (cuts) are added during the solution process, Benders decomposition is also called a row generation algorithm. This is an active field of research, as researchers are still finding new ways to apply the technique and optimise its performance.

This thesis aims to apply a novel Benders decomposition techniques to solve a large-scale investment planning problem. Although recent work such as

Mazzi's use of adaptive oracles [8] has significantly reduced the solution time of the algorithm, the model still relies on solving a full scenario tree to find the optimal investment decisions. In the real world, however, investment decisions are often driven by a small number of scenarios that have the greatest impact. Therefore, the goal of this thesis is to find new ways to adaptively expand and reduce the scenario tree during the solution process, so that scenarios that drive the investment decisions are identified and used in the solution.

The structure of this thesis is as follows: In Chapter 2, the relevant theory related to stochastic programming, scenario generation, scenario reduction and Benders decomposition is reviewed. The main part of the work is summarised in Chapter 3 in the form of a case study. First, the approach is outlined and the investment planning model, as well as the solution procedure are described in detail. The second section summarises different methods for assessing the future value surface at different nodes in the scenario tree. In the third section, one of the methods is applied in order to provide some practical examples for scenario reduction and scenario generation. The final section of this chapter is devoted to a review of the approach. Chapter 4 provides conclusions on the thesis and offers some ideas about further research.

2 Literature Review

This section aims to provide a brief overview of some of the theories that are relevant for this thesis. First, a quick introduction to Stochastic Programming (SP) and scenario trees is provided. Methods that deal with scenario generation and scenario reduction are introduced next and the most relevant papers are reviewed. Finally, a quick look at the Benders decomposition algorithm aims to provide the necessary understanding for the upcoming chapters.

2.1 Stochastic Programming

This brief overview of SP is aimed to provide just enough context for what is to come in the following chapters. A thorough review of the topic can be found in the textbooks by Birge et al. and Kall et al. [5, 9].

SP is a field of mathematics that can be applied to a broad area of topics, most notably portfolio optimisation, network optimisation and energy systems optimisation. Essentially, it is an extension to the field of mathematical programming which is concerned with introducing uncertainty into the problem. Whereas deterministic problems are formulated with known parameters, real world problems almost always include some uncertainty. For example, the amount of sunshine tomorrow, the return on an investment or the time it takes to drive from Edinburgh to Glasgow. When such elements of uncertainty are included in a stochastic program, they need to be approximated somehow. One way to do so is to use a random variable and probability distribution. However, this usually has no practical large-scale application because the problem quickly becomes intractable. In order to avoid this, the uncertainty is often modelled using discrete scenarios. For example, a 50% chance of sunshine and a 50% chance of no sunshine means that the stochastic problem has two scenarios to deal with.

Two-stage linear programs There are different ways to formulate a stochastic program. This paragraph introduces the one that is relevant for the remainder of this thesis. The idea of a two-stage stochastic program is that the first-stage decision is based on information that is available at that stage and cannot depend on future observations. Only the second-stage decision takes into account uncertainty and the probability of occurrence of each scenario in order to find an optimal decision. In a minimisation problem, the objective is to minimise the cost of the first-stage decision plus the expected cost of the optimal second-stage decision. The first-stage decision is also known as the "here-and-now" decision, while the second-stage decision is known as the "wait-and-see" decision. Finally, a two-stage problem is linear if the objective function and the constraints are linear.

2.1.1 Scenario Trees

A decision tree is a simple but useful visualisation tool. In its simplest form, a *deterministic* decision tree consists of nodes, which represent states, and arcs, which represent decisions. For each possible decision that can be taken in a state, an arc must lead to a node in a next stage or to a final leaf node. The first node in the tree is referred to as the root node and a sequence of decisions is also known as a policy.

The same idea can be applied to visualise *stochastic* problems, which are then called scenario trees. Instead of decisions, arcs represent outcomes of random variables and nodes represent points at which something happens, for example a change in the fuel price. The outcome of a random variable is also referred to as a realisation and it is important to remember that the realisation of a node is not known to previous nodes. A path from the root node to the leaf node is known as a scenario, thus the name scenario tree. Throughout this thesis, the aim is to keep the notation as simple as possible, therefore, scenarios, realisations and stages are described as needed within the text. However, for more general interpretations of outcomes, the base element ω is used to refer to specific nodes in the tree where investment decision are made.

In the SP literature [5, 9], scenario trees are almost always associated with scenario generation and scenario reduction. The difference between the two is not obvious at first sight, especially because scenario reduction is considered part of scenario generation in some literature [6]. Within this thesis, scenario generation refers to the generation of scenarios from distributions - that is the discretisation of the random variables - in order to create a scenario tree. On the other hand, scenario reduction assumes a given scenario tree and aims to reduce the size of the tree with as little impact on the solution as possible. The scenario reduction and generation methods which are reviewed next are based on two principles. They either decide on scenarios using only data, or they decide on scenarios by taking into account how they influence the optimal decisions or the optimal objective value.

2.1.2 Scenario Generation Methods

A good overview of scenario generation methods is provided by Kaut et al. and Mitra et al. [6, 10]. "Pure" scenario generation methods, for example methods that are based on sampling values from the stochastic process or matching some statistical properties, are not the main focus of this thesis. However, it is worth exploring some algorithms that generate scenarios iteratively, as more information becomes available during the solution process. One such algorithm is reviewed next.

Using dual variables to add scenarios Casey et al. [11] developed a method that uses the dual variables from the current solution to iteratively add scenarios to the tree. They consider a Multistage Stochastic Linear Program where the uncertainty is in the right-hand side.

The algorithm they use to solve this problem can be summarised as follows: In a first step, the scenario tree is initialised in its deterministic form by replacing all random variables with their expectations. This aggregated problem is then solved to get primal-dual pairs and optimal values for each stage or each node in the tree. Given these primal decisions, dual multipliers and the optimal basis of each node, a linear subproblem for each node is derived analytically. For each node, the uncertainty varies only within the range represented by the respective node. For each node, the upper and lower bounds are then calculated and the gap between them is used to decide which node to split on. In this sense, a large gap allows to identify low probability / high costs events. The probability of occurrence of each node is determined by assessing whether some random variables cause infeasibility at the node and if so, the probability is redistributed up to that point.

2.1.3 Scenario Reduction Methods

One of the challenges of SP is about finding effective ways to evaluate the importance of scenarios and to use that information to reduce the scenario tree in such a way that the reduced tree produces a solution that does not deviate much from the solution of the original optimisation problem. Scenario reduction methods were first introduced by Dupačová et al. [12] with significant contributions by Heitsch, Römisch and Kuska [13, 14, 15]. In [16], Heitsch et al. concluded that optimal scenario reduction is in fact a combinatorial optimisation problem of k-median type and thus NP-hard[1].

A selection of relevant scenario reduction algorithms are reviewed next.

Backward Reduction and Fast Forward Selection Kuska et al. [13] propose two heuristics-based algorithms to reduce a scenario tree. In the Backward Reduction algorithm, two steps are repeated until a defined num-

[1]NP-hard means it is unknown whether there exists an efficient algorithm that can find a solution to the problem.

ber of scenarios are left: First, the scenario, whose removal will cause the smallest error in the Kantorovich distance between two adjacent scenarios, is identified. Second, this scenario is removed and its probability is redistributed. In the Fast Forward Selection algorithm, the full tree is known but the first iteration starts from an empty tree. Again, two steps are repeated until a specified number of scenarios is reached: First, the scenario whose addition will cause the biggest improvement in terms of the Kantorovich distance is identified. Then, this scenario is added and the probability is split among the new set of scenarios.

Kuska et al. tested their algorithms on a portfolio management problem for hydro-thermal generation system and concluded that the optimal value of the optimisation model can be approximated using a small number of scenarios.[2]

EVPI-based importance sampling Dempster et al. [19] measure the importance of scenarios by calculating the Expected Value of Perfect Information (EVPI) at each node. EVPI is defined as the amount a decision maker would be ready to pay in return for accurate information about the future [5]. It is calculated by subtracting the expected value of the optimal solution - with all uncertainties about the future removed - from the solution corresponding to the stochastic problem. Dempster et al. then use an EVPI-based importance sampling algorithm to approximate the optimal solution of the problem. By sampling more often from branches with high relative EVPI and less from branches with negligible EVPI, they push the sampled approximations towards the optimal value. During the solution process, they update the tree as follows: When the nodal EVPI is above a certain tolerance level and the descendants of that node have not been considered previously (which means there was no branching from them in the previous version of the tree), the tree is expanded by adding one or more stages. On the other hand, when the nodal EVPI value is below the tolerance level over a sequence of iterations, the subtree is collapsed. The authors state that this procedure can greatly reduce the size of the tree and thus more detailed problems and even problems with continuous distribution can be solved.

In their book on SP, Kall et al. [9] consider the question of whether a large EVPI at a node in the tree means that it is important to solve a stochastic model. While a large EVPI shows that randomness plays an important role in the problem, it does not necessarily prove that a deterministic model would not work as well. Furthermore, it is possible to have a very low EVPI at one node, but at the same time have a node far down in the tree with a very high EVPI. On the other hand, they are quite certain, that a small

[2]The original algorithm was implemented in GAMS [17]. Furthermore, a wrapper is available for the Julia programming language [18].

EVPI means that randomness plays a minor role in the model.

Clustering-based techniques Clustering techniques that select representative scenarios from a given tree are another method that is becoming increasingly popular in the transmission and generation planning literature. For example, a method is proposed in [20] that clusters on input variables such as demand and wind generation, while in [21], the authors propose to cluster on operational variables (for example power flow in lines).

Another approach is presented by Sun et al. [22] who suggest an objective-based method that clusters on the investment decisions. They argue that clustering in the input domain may not lead to an efficient scenario reduction, as some significantly different scenarios can lead to identical investment decisions. Their framework is motivated by the fact that usually only a limited number of scenarios drive investment decisions. Although their clustering method is computationally expensive because each subproblem needs to be solved in order to gain information about investment decision, they are able to reduce scenarios significantly while still achieving the optimal investment decision.

2.2 Benders Decomposition

Benders Decomposition is a technique that exploits the block structure of a mathematical problem in order to split a large problem into several smaller problems [7]. The idea is that solving two or more smaller problems is much faster than solving the original problem, even if the smaller problems have to be solved several times to arrive at the solution. Benders originally applied his decomposition idea to mixed integer programming problems, however, the technique is also very well suited for stochastic programming as each node in the tree (each realisation of an uncertainty parameter) can be solved as an independent subproblem. In the case of an investment planning problem, the idea behind Benders decomposition can be summarised as follows: All variables related to investment decisions are part of the master problem, while each node corresponding to one realisation of the uncertainty is a subproblem that is solved independently for a given solution of the master problem. Each time a subproblem is solved, a cutting plane - known as a Benders cut - is generated and added as a constraint to the master problem. The master problem is then resolved and a new solution, which is ideally closer to the optimal solution, is found. This way, the algorithm iteratively progresses towards the optimal solution. However, the truly optimal solution is usually never reached and, therefore, a convergence limit between the upper bound and the lower bound is set. The algorithm terminates once two bounds have converged past the limit.

2.2.1 Benders Decomposition with Adaptive Oracles

In the original Benders decomposition algorithm, each subproblem is solved once per iteration in order to generate a full set of cutting planes. When the number of subproblems is large and the subproblems are difficult to solve, this process can be very slow. An alternative approach is to solve only a subset of subproblems and to derive approximate cutting planes for the subproblems that are not solved in that iteration. In the literature, the technique of adding such inexact cutting planes is sometimes referred to as "calling the oracle".

Mazzi et al. [8] developed two such oracles that exploit the mathematical properties of the subproblems to generate inexact but valid cutting planes of those subproblems that are not solved in an iteration. These oracles use the knowledge of having solved other subproblems in previous iterations and so progressively adapt towards more asymptotically exact cutting planes. The specifics of the Mazzi's algorithm are explained in more detail in section 3.2.1, as this is the algorithm that is used in the case study.

3 Case Study

This section summarises the results of a case study on a stochastic investment planning model for energy systems. During this study, various scenario reduction and generation methods were tested and applied to a large-scale planning problem that is based on previous work by Mazzi et al. [8]

All computation was done using a MacBook Pro with a 2.2 GHz Intel Core i7 processor and 16 GB of RAM. The models were implemented in Julia 1.1, using JuMP 19.0 and solved with Gurobi 7.5. The code that was used to test these methods, generate the data and produce the graphs is available at `https://github.com/thomasmaerki/msc_thesis` .

3.1 Approach

In the literature review, a selection of papers that deal with large-scale stochastic optimization problems are summarised. While some offer interesting insights, such as Casey's use of dual variables to add scenarios or Dempster's way of iteratively adding and removing scenarios, none of these solutions are directly applicable to the problem that is introduced in the next section. The main reason for this is that the algorithm that was used to solve the problem (see section 3.2.1) is itself based on very recent work. Nevertheless, the previous approaches serve as an inspiration for the methods that are tested in this section.

With more uncertainties and stages added to the problem, the size of the scenario tree grows exponentially. This thesis therefore seeks to find ways to iteratively incorporate scenarios that behave badly under the current investment plans and to remove or merge scenarios that do not add any benefit to the policy that results from the solution of the problem. The underlying idea is the assessment of the value surface at different nodes in the tree. Like any minimisation problem, a stochastic planning problem can be imagined as having a point where the cost of the current investment plan is minimal.[3] At this point, the gradients with respect to the investment decisions (the decision variables) have a value of zero, meaning that the value surface is flat. With large optimisation problems it is a hard task to find this point and with decomposition methods, this point is only approached asymptotically. Therefore, the current investment policy at any node in the tree is likely sub-optimal and the value surface is not flat. The idea is that this value surface not only guides the iterative solution process, but also provides information about the uncertainties that make a difference.

[3] The optimisation problem studied in this section is a large-scale linear problem, therefore, a minimum is always a global minimum.

The next sections are structured as follows: First, the investment planning problem and the algorithm used to solve the problem are introduced. Next, different methods for assessing the value surface in a node of the tree are presented. Finally, the results of using one of the methods as a basis for merging scenarios in order to produce a reduced scenario tree from an originally much larger tree are summarised.

3.2 Investment Planning Model

In this section, a stochastic investment planning model for energy systems that is based on previous work by Mazzi et al. [8] is introduced. The model considers a planning horizon of 15 years, during which both investment and operational decisions are made. Investment decisions, that is how much and in what technologies to invest, are made at the start of the planning horizon and in 5 years time. Construction time is assumed to be five years, therefore, capacity that is being added at the beginning of the planning horizon will be available from year 5 until 15 and capacity that is being added in 5 years time will be available from year 10 until 15. For a given installed capacity, the cost of optimally operating the system for 5 years is computed by planning the optimal power output of each technology at each hour of the day for 365 days, multiplied by 5 years. The model considers a set of 12 technologies which are summarised in Table 1. In Appendix B, the table is expanded to include their efficiencies, fuel cost, variable cost, fixed cost and investment cost.

Technology	Type
Coal	Thermal generation unit
Coal with carbon capture and storage	Thermal generation unit
Open Cycle Gas Turbine (OCGT)	Thermal generation unit
Combined Cycle Gas Turbine (CCGT)	Thermal generation unit
Diesel	Thermal generation unit
Nuclear	Thermal generation unit
Pumped hydroelectric storage	Storage unit (low capacity)
Pumped hydroelectric storage	Storage unit (high capacity)
Lithium-ion battery storage	Storage unit
Onshore Wind	Renewable generation unit
Offshore Wind	Renewable generation unit
Photovoltaics	Renewable generation unit

Table 1: Set of $\mathcal{P}$ technologies

In this model, the underlying uncertainties that affect operational decisions are assumed to be stationary, because their effect on a stable physical system is limited. However, investment planning decisions are subject to a changing economical, technological and political environment that results in a non-stationary planning process. Therefore, different uncertainties, such as relative level of energy demand or yearly CO_2 emission limits, are modelled as possible future scenarios.

Mathematically, the stochastic investment planning problem can be expressed as

$$\min_{\mathbf{x}\in\mathcal{X}} \quad f(\mathbf{x}) + \sum_{i\in\mathcal{I}} \pi_i \, g(x_i, c_i) \tag{1}$$

where $\mathcal{I}$ is the set of stochastic decision nodes, each associated with a probability π_i. The $\mathbf{x}$ represent investment decisions which yield the expected total investment and fixed cost $f(\mathbf{x})$:

$$f(\mathbf{x}) = \sum_{i\in\mathcal{I}} \pi_i \sum_{p\in\mathcal{P}} (c_{pi}^{inv} x_{pi}^{inst} + c_{pi}^{fix} x_{pi}^{ACC})$$

The parameters c_{pi}^{inv} and c_{pi}^{fix} are the investment cost and fixed costs of technology p at node i respectively. The decision variable x_{pi}^{inst} is the installed capacity of technology p at node i, which is available in all successor nodes of the tree. x_{pi}^{ACC} is the accumulated capacity, meaning the capacity that the operational model can work with, which is given by

$$x_{pi}^{ACC} = x_{pi}^{hist} + \sum_{i_0\in\mathcal{I}_0} x_{pi_0}^{inst} \quad \forall p \in \mathcal{P}, i \in \mathcal{I}, \tag{2}$$

where $\mathcal{I}_0$ defines the set of predecessor nodes to the node i and x_{pi}^{hist} is the historically available capacity (which can also include the decommissioning of power plants).

Finally, $g(x_i, c_i)$ in equation (1) is the optimal solution of the operational subproblem, that is the cost of operating the energy system for 5 years:

$$g(x_i, c_i) = \min_{y_i\in\mathcal{Y}} \{c_i^\intercal C y_i \,|\, A y_i \leq B x_i\}, \quad \forall i \in \mathcal{I}, \tag{3}$$

where x_i and c_i are the coefficient vectors

$$x_i = (\{x_{pi}^{ACC}, \forall p \in \mathcal{P}\}, -v_i^D, v_i^E), \quad \forall i \in \mathcal{I},$$
$$c_i = (c_i^{nucl}, c_i^{co_2}), \quad \forall i \in \mathcal{I},$$

with v_i^D as the relative energy demand, v_i^E as the yearly CO_2 limit, c_i^{nucl} as the uranium fuel price and $c_i^{co_2}$ as the CO_2 emission price. The objective of the operational subproblem (3) is to minimise the operational costs at each node i, given the cost matrix C and the operational decisions y_i. The matrices A and B impose various operational constraints, such as generator and storage capacities or gas turbine ramp up constraints. A full mathematical formulation of the Operational Problem (OP) is included in Appendix A.

The operational subproblem is difficult to solve, as operational decisions must be made on an hourly basis (8,760 hours per year) with varying energy demands, wind patterns and solar activity. This results in a minimization problem with 140,191 variables and 411,738 constraints. Furthermore, it is worth noting that the model includes a load shedding variable with a very high cost associated with it. This essentially means that the problem will always have a feasible solution. For example, instead of investing and using any generation unit, the model can decide to shed the load of all demand, which yields a feasible, although very far from optimal, solution.

3.2.1 Solving the Model

In order to solve this minimisation problem, Mazzi's decomposition algorithm with adaptive oracles is employed (as previewed in section 2.2.1). The full description of the algorithm can be found in [8], while this section offers a summary of its main steps.

The original stochastic planning problem (1), that is the Master Problem (MP), is relaxed to obtain the Relaxed Master Problem (RMP):

$$\begin{aligned} \min_{\mathbf{x}\in\mathcal{X}} \quad & f(\mathbf{x}) + \sum_{i\in\mathcal{I}} \pi_i \, \beta_i \\ \text{s.t.} \quad & \beta_i \geq \theta + \lambda^\top (x_i - x) \quad \forall i \in \mathcal{I}, \quad \forall (x, \theta, \lambda) \in \Theta_i^{(j-1)} \end{aligned} \tag{4}$$

In each iteration j, the following steps are executed until a pre-defined convergence tolerance (ϵ) is reached: First, the RMP is solved and the lower bound of the MP is computed from the objective value of the RMP:

$$\underline{\mathrm{L}}^{(j)} := f(\mathbf{x}^{(j)}) + \sum_{i\in\mathcal{I}} \pi_i \beta_i^{(j)}$$

Second, given the investment decision from the previous step, one subproblem is solved to obtain an exact cutting plane. Third, the oracles are called to obtain inexact but valid cutting planes for all other subproblems. The oracles also compute upper bounds ($\overline{\theta}_i^{(j)}$) and lower bounds ($\underline{\theta}_i^{(j)}$) for each subproblem. Next, the upper bound of the MP is computed with the upper bounds of the subproblems obtained in the previous step:

$$\overline{\mathrm{U}}^{(j)} := \min\left(\underline{\mathrm{U}}^{(j-1)},\ f(\mathbf{x}^{(j)}) + \sum_{i\in\mathcal{I}} \overline{\theta}_i^{(j)}\right)$$

Finally, the cutting planes are added to the RMP and the degree of convergence is calculated ($\overline{\mathrm{U}}^{(j)} - \underline{\mathrm{L}}^{(j)}$).

Figure 2 shows how such an iterative solution process can look like. The x-axis shows the degree of convergence, the y-axis shows the objective value and each dot represents one iteration. The blue lines show the upper and lower bounds of the MP converging towards the optimal value, while the red line shows the optimal objective value given the investment decision at the current iteration. Interestingly, the lower bound converges much faster, which is a useful insight to take into the next section, when the gradients of the problems are assessed.

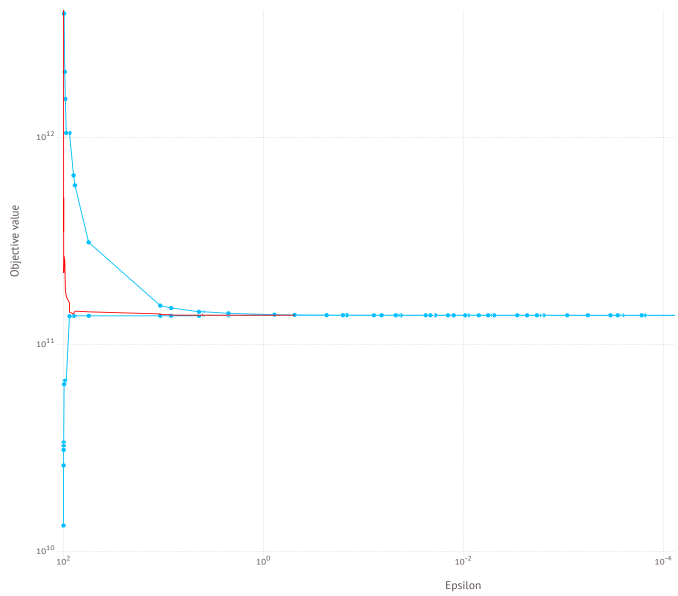


Figure 2: Convergence of upper and lower bounds

3.2.2 Assessing the Dual Variables

As outlined in 3.1, it is the value surface at a particular node in the tree, in particular the gradients with respect to the investment decisions, which are of interest for the analysis. The gradient with respect to a particular technology can also be interpreted as the value (positive or negative) of having a little bit more of that technology in a node (and potentially at all successor nodes). According to the duality theory, it is the dual variables that provide exactly this information. The value of the dual variable is defined as the improvement in the objective value in response to an infinitesimal relaxation (on the scale of one unit) in the right-hand side of a constraint. If that constraint is the inventory constraint (see equation 2), the value of the dual variable should show how much the objective value changes if there is one additional unit of a particular technology available.

One difficulty arises due to the fact that the algorithm decomposes the problem in an investment problem and multiple operational subproblems. The question is whether the dual variable truly capture the future value of having one additional unit of a particular technology in one node of the tree and subsequently in all successor nodes. As an example, it would be inconsistent if the dual variable considers only operational costs or only investment costs. It would also be inconsistent if only the current node and not all successor nodes are considered.

Therefore, an alternative formulation of the original inventory constraint in equation 2, another formulation of that same constraint is tested. Instead of calculating the accumulated capacity of technology p in node i by adding up the investment decisions of all previous investment nodes, only a single predecessor node is considered in the following formulation:

$$x_{pi}^{acc} = 0 \quad \forall p \in \mathcal{P}, i = 1$$

$$x_{pi}^{acc} = x_{pi_{i0}}^{inst} + x_{pi_{i0}}^{acc} \quad \forall p \in \mathcal{P}, i \in \mathcal{I} \setminus i = 1 \tag{5}$$

Here, x_{pi}^{acc} is the accumulated new capacity of technology p in node i, which is given by the equivalent variable of the predecessor node plus the additionally installed capacity in the predecessor node. Because no new capacity can be available in the root node, the first constraint is forced to equal zero. The total accumulated capacity of technology p in node i, that is the capacity that the operational model can work with, is then given by

$$x_{pi}^{ACC} = x_{pi}^{hist} + x_{pi}^{acc} \quad \forall p \in \mathcal{P}, i \in \mathcal{I}, \tag{6}$$

The original formulation (equation 2) and the above formulation with three constraints are tested on a planning problem with two sources of uncertainty. Both formulations resulted in the same investment policy so it can be confirmed that the formulation does not change the mathematical structure of the problem.

Appendix C summarises the gradient assessment of the original inventory constraint, while Appendix D summarises the assessment of the new formulation. Besides the gradients at the final iteration (after a certain convergence tolerance was reached), the gradients at iteration 0 are also included. At this point in the solution procedure, no cuts are added to the RMP yet (that is no information from solving operational problems is available). At this iteration, the model only knows about investment cost c_{pi}^{inv} and fixed cost c_{pi}^{fix} of technology p at node i, which is a value that can be calculated manually. For each constraint, the appendices also include one graph which depicts the gradients as they change during the solution process (the graphs always show the same investment node).

The original inventory constraint and the constraint associated with equation 6 have the same values at the final iteration and, in addition, the graphs display a very similar pattern. Equation 5 resulted in different values, but again, a similarity in the general pattern can be observed. At iteration 0, none of the values could be verified by calculating the gradient manually. At this stage, it is unclear why these values could not be verified.

The next method is based on a slightly different approach and involves an extra step in the solution procedure. It can be described as follows: At each iteration j, the RMP (equation 4), the SP and the oracles are solved and the cuts are added to the RMP. The optimal investment decision $\hat{\mathbf{x}}^{(j)}$ is then added as a fixed constraint to the RMP:

$$x_{pi}^{inst} = \hat{\mathbf{x}}^{(j)} \quad \forall p \in \mathcal{P}, i \in \mathcal{I} \tag{7}$$

Once the investment decision is fixed, the RMP is solved once more[4]). The dual variables associated with the above constraint, hereafter referred to as $\mu_{pi}^{(j)}$, can then be assessed. As before, the validity is tested by manually calculating the value at iteration 0 ($c_{pi}^{inv} + c_{pi}^{fix} = \mu_{pi}^{(j=0)}$). For example, without any information from the subproblems, the cost of installing one more unit (GW) of a diesel generator in stage 1 (@ 0 years) is £ 158,000,000. This amount equals the investment cost of £ 128,000,000 plus the operating cost for 10 years $10 \cdot 3,000,000$ (Appendix B includes a list of the investment and operating costs of each technology). Table 2 shows the gradients at iteration 0 for each node and each technology. $\omega 1$ refers to the first stage investment decision, while $\omega 2$ - $\omega 10$ refer to the second stage investment decisions. All these values could be verified by manually calculating the gradient, therefore, of all the approaches presented so far, this one is the only one that could be verified to work at iteration 0. Furthermore, Table

[4]With fixed decision variables, this optimisation problem requires very little computation. For example, in case 1 (see section 3.3.1), only 0.008% of the total computation time was used to solve the problem and store the gradients.

4 in the next section shows that the gradients can also be interpreted once the algorithm converged to near optimality. For example, technologies such as onshore wind, nuclear or CCGT have a value of zero in many investment nodes, which means that these technologies are optimally invested. On the other hand, a clearly unfavourable technology such as coal in node $\omega 2$ (in this node, a very high CO_2 tax of 100 is realised) has a value that translates to almost exactly the investment and fixed costs. Finally, Table 4 also shows that investing in stage 1 ($\omega 1$) has bigger impact than investing in stage 2, which also makes sense since the technology would be in operation twice as long.

Bases on these observations, it is reasonable to assume that the method presented above allows for the assessment of gradients that consider both operational and as investment costs, as well as the effects on the current node and successor nodes.

	$\omega 1$	$\omega 2$	$\omega 3$	$\omega 4$	$\omega 5$	$\omega 6$	$\omega 7$	$\omega 8$	$\omega 9$	$\omega 10$
Coal	8.64e8	4.8e7	4.8e7	4.8e7	4.8e7	4.8e7	4.8e7	4.8e7	4.8e7	4.8e7
Coal&CCS	1.43e9	7.94e7	7.94e7	7.94e7	7.94e7	7.94e7	7.94e7	7.94e7	7.94e7	7.94e7
OCGT	2.1e8	1.17e7	1.17e7	1.17e7	1.17e7	1.17e7	1.17e7	1.17e7	1.17e7	1.17e7
CCGT	3.51e8	1.95e7	1.95e7	1.95e7	1.95e7	1.95e7	1.95e7	1.95e7	1.95e7	1.95e7
Diesel	1.58e8	8.78e6	8.78e6	8.78e6	8.78e6	8.78e6	8.78e6	8.78e6	8.78e6	8.78e6
Nuclear	1.64e9	9.13e7	9.13e7	9.13e7	9.13e7	9.13e7	9.13e7	9.13e7	9.13e7	9.13e7
PumpL	7.45e8	4.14e7	4.14e7	4.14e7	4.14e7	4.14e7	4.14e7	4.14e7	4.14e7	4.14e7
PumpH	8.79e8	4.88e7	4.88e7	4.88e7	4.88e7	4.88e7	4.88e7	4.88e7	4.88e7	4.88e7
Lithium	5.15e8	2.86e7	2.86e7	2.86e7	2.86e7	2.86e7	2.86e7	2.86e7	2.86e7	2.86e7
Onwind	5.95e8	3.31e7	3.31e7	3.31e7	3.31e7	3.31e7	3.31e7	3.31e7	3.31e7	3.31e7
Offwind	1.41e9	7.83e7	7.83e7	7.83e7	7.83e7	7.83e7	7.83e7	7.83e7	7.83e7	7.83e7
PV	2.01e9	1.12e8	1.12e8	1.12e8	1.12e8	1.12e8	1.12e8	1.12e8	1.12e8	1.12e8

Table 2: Gradients at iteration 0

Assessing the dual variables of operational problems If the node is an operational node (in this case a leaf node in the tree), the value surface of the operational subproblem can also be used in the assessment. When a subproblem is solved, the available capacity per technology is fixed and thus the gradient with respect to each technology can be assessed from the dual variable of the fixed capacity constraint. Since these gradients are used to generate the cutting planes for the Benders decomposition algorithm, they are readily available. If the Benders algorithm with adaptive oracles is used, solving one subproblem will yield (inexact) gradients for all subproblems, so the correlation between these gradients can be assessed from iteration 1 onwards.

3.3 Scenario Reduction

In this section, investment planning problems with various scenario tree structures are solved and their gradients with respect to the investment decisions are assessed at each node.

The graphs that are included in the appendices and are referred to in this section display either the investment decisions per technology (in GW) for one node or the gradients with respect to the investment decisions for one node. Because Benders decomposition is an iterative algorithm, the x-axis shows these values for each iteration. This provides a useful tool to assess how decisions and gradients change during the solution procedure. However, it is worth noting that only the final value matters.

3.3.1 Uncertainty in $c_i^{co_2}$ and v_i^E

The first case considers two sources of uncertainty with three possible outcomes each: Uncertain CO_2 emission price ($c_i^{co_2}$) and uncertain yearly CO_2 limit (v_i^E). This case is chosen on purpose because the two uncertainties address the same issue, which means that some redundancy can be expected. The resulting scenario tree (see Appendix E) has 10 investment nodes, 90 operational nodes and a total of 91 nodes (the root node is an investment but not an operational node).

The problem is solved using the algorithm presented in 3.2.1 with a convergence tolerance of 0.001%. The resulting investment policy is summarised in Table 3.

	$\omega 1$	$\omega 2$	$\omega 3$	$\omega 4$	$\omega 5$	$\omega 6$	$\omega 7$	$\omega 8$	$\omega 9$	$\omega 10$
Coal										
Coal&CCS	0.84	6.59	6.59	6.59			1.58			1.58
OCGT										
CCGT	11.74				5.57	2.77		5.55	2.52	
Diesel	2.07				2.33	2.02	1.23	2.49	2.41	1.26
Nuclear		10.0	10.0	10.0	4.94	8.02	10.0	4.8	7.9	10.0
PumpL										
PumpH										
Lithium										
Onwind	19.0									
Offwind										
PV										

Table 3: Case 1: Newly installed capacity (GW)

Note that ω 1-10 corresponds to the newly installed capacity in each of the 10 investment nodes. The graphs corresponding to these investment decisions can be found in Appendix G. For the interpretation of these numbers, it is also important to note that onshore wind in $\omega 1$ and nuclear power in $\omega 2$, $\omega 3$, $\omega 4$, $\omega 7$ and $\omega 10$ are at their maximum capacity.

	$\omega 1$	$\omega 2$	$\omega 3$	$\omega 4$	$\omega 5$	$\omega 6$	$\omega 7$	$\omega 8$	$\omega 9$	$\omega 10$
Coal	6.84e8	4.79e7	4.79e7	4.79e7	3.74e7	3.75e7	3.81e7	3.68e7	3.69e7	3.77e7
Coal&CCS	-1.96e5	-8.86e4	-8.86e4	-8.86e4	1.15e6	1.63e6	0.0	1.06e6	1.42e6	0.0
OCGT	2.96e7	1.13e7	1.13e7	1.13e7	1.49e6	1.58e6	1.83e6	2.32e5	2.09e5	7.73e5
CCGT	0.0	1.5e7	1.5e7	1.5e7	0.0	0.0	1.03e6	0.0	0.0	1.65e6
Diesel	0.0	8.78e6	8.78e6	8.78e6	0.0	0.0	0.0	0.0	0.0	0.0
Nuclear	2.96e6	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
PumpL	6.95e8	3.94e7	3.94e7	3.94e7	3.89e7	3.87e7	3.85e7	3.88e7	3.86e7	3.84e7
PumpH	8.54e8	4.77e7	4.77e7	4.77e7	4.76e7	4.75e7	4.74e7	4.75e7	4.74e7	4.73e7
Lithium	1.81e8	-9.35e6	-9.35e6	-9.35e6	7.8e6	0.0	0.0	0.0	0.0	0.0
Onwind	0.0	3.4e8	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Offwind	1.09e8	7.46e5	7.46e5	746e5	6.51e6	7.06e6	6.33e6	6.5e6	7.03e6	6.29e6
PV	1.57e9	8.36e7	8.36e7	8.36e7	8.79e7	8.77e7	8.7e7	8.79e7	8.77e7	8.7e7

Table 4: Case 1: Gradients w.r.t. investment decisions

The gradients with respect to the investment decision at the final iteration are summarised in Table 4 and the corresponding graphs can be found in Appendix H. Based on both the table and the graphs, a visual inspection shows that for some nodes, the gradients with respect to the investment decisions perfectly coincide after a few iterations. In particular, this can be seen in the investment nodes $\omega 2$, $\omega 3$ and $\omega 4$. In these nodes, the CO_2 emission price ($c_i^{co_2}$) increases from 40 in the first investment stage (at year 0) to 100 in the second investment stage (at year 5). At the operational stage (at year 10), the CO_2 price can be either 80, 100 or 120. Due to this high price, the policy given by the model encourages investment in CO^2 neutral technologies such as nuclear and coal with CCS. The CO_2 limit has no effect on the decision, therefore, it would seem reasonable to merge these three investment nodes. Note that even though the policy shows a similar investment decision for $\omega 7$ and $\omega 10$, the gradients are not the same.

Furthermore, a large amount of operational nodes show correlated gradients. Whenever a node splits up into three child nodes (remember: each uncertainty is discretised into three values), the top two have the same gradients, while the bottom one (the scenario with the lowest CO_2 limit of the three) has a different gradient[5]. As before, it would also seem reasonable to merge the operational nodes with coinciding gradients.

Merging all scenarios with correlating gradients results in a reduced scenario tree with 8 investment nodes and 39 operational nodes, which corresponds to a tree just slightly larger than half of the original one. When merging nodes, the new node takes the mean value of the previous ones while the probability of that node equals the sum of the probabilities of the merged nodes. The reduced scenario tree is depicted in Appendix F and the resulting investment

[5]The impact of the load shedding cost is discussed further in section 3.5.

policy is summarised in Table 5.

	$\omega1$	$\omega2$	$\omega3$	$\omega4$	$\omega5$	$\omega6$	$\omega7$	$\omega8$
Coal								
Coal&CCS	0.83	6.7			1.58			1.58
OCGT								
CCGT	11.77		5.79	2.93		5.44	2.45	
Diesel	2.05		2.13	1.91	1.25	2.59	2.47	1.25
Nuclear		10.0	4.92	8.01	10.0	4.81	7.9	10.0
PumpL								
PumpH								
Lithium								
Onwind	19.0							
Offwind								
PV								

Table 5: Reduced case 1: Newly installed capacity (GW)

As the two policies show, the original problem and the reduced problem indeed result in the same policy. Even though the reduced problem is almost half the size of the original problem, the computation time has not decreased substantially, as Table 6 shows.

	Nodes	Iterations	Seconds
Original tree	91	242	819
Reduced tree	47	229	741

Table 6: Case 1: Comparison of computation time

This highlights the effectiveness of the Benders algorithm with adaptive oracles: Every time a subproblem is solved, the algorithm generates inexact but valid gradients for all subproblems, as opposed to a standard Benders algorithm, which solves all subproblems in each iteration. With adaptive oracles, the algorithm therefore does not benefit from a problem with substantially less subproblems. Essentially, the subproblems that were merged are those for which the adaptive oracles can already provide accurate cutting plane approximations. However, it is also worth noting that this problem is a very small example, in which solving the RMP and the oracles takes a negligible time relative to solving the subproblem. For a bigger problem, this will not be the case and reducing the size of the RMP and the number of oracles called in each iteration would be of more value.

3.3.2 Uncertainty in v_i^D and v_i^E

The second case considers two sources of uncertainty with five possible outcomes each: Uncertain relative level of energy demand (v_i^D) and uncertain yearly CO_2 limit (v_i^E). In order to focus on investment nodes, only one deterministic operational node follows the final investment node. The resulting scenario tree (see Appendix I) has 25 investment nodes, 50 operational nodes and a total of 51 nodes (the root node is an investment but not an operational node).

Just as in the previous case, the problem is solved with a convergence tolerance of 0.001%. Appendix K summarises the investment policy, which considers one decision in the first stage ($\omega 1$) and 25 different outcomes in the second stage ($\omega 2$ - $\omega 26$). An interesting pattern in both the investment policy and the gradients emerges[6]: With decreasing energy demand, the CO_2 limit becomes less relevant, and so the further down the tree, the more scenarios can be merged ($\omega 7$ - $\omega 8$ / $\omega 12$ - $\omega 14$ / $\omega 17$ - $\omega 20$ / $\omega 22$ - $\omega 26$).

The reduced scenario tree with 15 investment nodes and 30 operational nodes is depicted in Appendix J and the investment policy is summarised in Appendix L. The reduced problem produces a very similar policy with minor differences in the range of 0.22 GW that are likely due to the convergence tolerance. Surprisingly though, the smaller problem takes more iterations and more time to reach the same convergence tolerance of 0.001%, as Table 7 shows.

	Nodes	Iterations	Seconds
Original tree	51	254	760
Reduced tree	31	266	808

Table 7: Case 2: Comparison of computation time

3.4 Scenario Generation

The deterministic version of the investment planning problem is depicted in Figure 3. In this form, there are no uncertainties and all nodes have a probability of occurring of 100%.

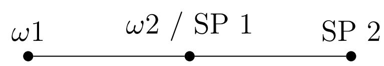


Figure 3: Deterministic scenario tree

When uncertainty is added to the model, the tree is split on some nodes in order to capture this uncertainty. In the context of this study, the uncertainty is represented by discretised scenarios, so the questions that arise are the

[6]The graphs showing the gradients are not included in the appendices, as they do not offer much benefit besides showing similar patterns in some nodes.

following: Where should additional nodes be added, how many should be added, and what probability of occurrence should be assigned to each node?

As previously discussed in section 2.1.2, none of these questions are easy to answer and the literature does not offer a universal solution. For example, in [11], the nodal probabilities are distributed according to an infeasibility measure, which is not available for the problem described in this thesis.

Having these concerns in mind, the goal of this section is to determine whether the method employed in the previous chapter can be transferred to scenario generation as well. However, it must be pointed out that the method presented here should not be considered a true scenario generation method. Rather, it is a method to select and assess scenarios, based on whether the scenario has an impact on the investment policy. Although this is not a generally used term within the literature, it could be referred to as a scenario expansion method.

3.4.1 Uncertainty in $c_i^{co_2}$ and v_i^E

Just as case 1 in section 3.3.1, the third case deals with two sources of uncertainty with three possible outcomes each: Uncertain CO_2 emission price ($c_i^{co_2}$) and uncertain yearly CO_2 limit (v_i^E). However, case 3 starts from its deterministic form and examines whether adding an additional source of uncertainty changes the investment policy. If adding uncertainty results in a scenario tree, which has the same gradients with respect to the investment decisions as the deterministic tree, then adding the uncertainty does not offer any benefit. On the other hand, if the expanded scenario tree has different gradients from the deterministic tree, then it is an indication that the uncertainty may have an impact on the solution.

With two sources of uncertainty and three possible outcomes each, there are essentially six ways of splitting the deterministic node at the second investment stage. When a deterministic path is split into three stochastic scenarios, each of the scenarios then has a probability of occurence of $\frac{1}{3}$. If all splits result in scenarios with different gradients, a total of nine successor nodes of the root node can be added, each with a probability of $\frac{1}{9}$. Each operational node can then be split once more in six possible ways. However, this case focuses on the expansion at the second stage for now. The general form of one such expansion is depicted in Figure 4. Appendix M summarises the six ways of splitting a deterministic tree into a tree with uncertainty in $c_i^{co_2}$ (with values of 100, 60 and 20) and uncertainty in v_i^E (with values of 0.9, 0.8 and 0.7).

By solving each of the problems in Appendix M, their gradients with respect to the investment decisions can be compared in each of the three investment

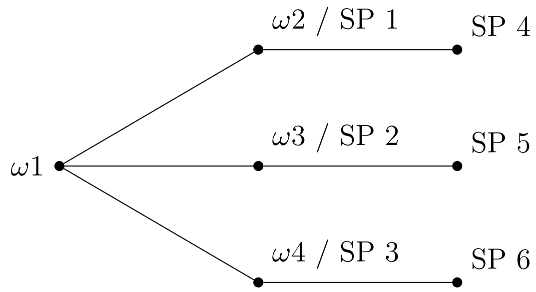


Figure 4: Expanding a deterministic scenario tree

nodes. If these gradients are the same, having three scenarios has no benefit compared to having a single deterministic scenario. The analysis shows that equivalent gradients could only be observed in the first problem, which has a deterministic CO_2 emission price of £ 100 and an uncertain CO_2 limit. In the other five problems, no gradients that coincide could be observed and, subsequently, each node also has a different investment policy. In order to illustrate this, Appendix N includes a selection of graphs that depict problems with coinciding gradients and with non-coinciding gradients. Based on this analysis, a tree can be constructed that includes one scenario with a CO_2 emission price of £ 100 and a relative CO_2 limit between 0.7 and 0.9[7], in addition to the other five scenarios that are included in Appendix M. With the exception of the operational nodes in stage 3, which are not considered in this case, the resulting scenario tree is the same as the reduced tree in case 1 (see Appendix F).

This goes to show showed that the same scenario tree can be constructed either by reducing or expanding the scenario tree with the method described above.

3.5 Review of Approach

This section addresses some issues that were encountered along the way. These might not have caused problems for the specific cases analysed, though it certainly rises the question whether the methodology can be generalised.

First, the results in the previous section 3.4.1 should be treated with caution, because the methodology employed has some drawbacks when the uncertainty is in the cost coefficient. This is because differences in costs, for example CO_2 emission price or fuel price, will affect the values of the gradients, regardless of whether a particular uncertainty in cost has an influence

[7]A limit of 1.0 equals the total CO_2 emissions in tonnes per year if the deterministic version of the problem is solved to optimality without an emission limit.

on the solution. For example, at a very low CO_2 emission limit, it might very well be that the CO_2 emission price does not have an effect on the solution. However, a higher price will always result in a different gradient with respect to the investment decision compared to a lower price. Although the method worked in the previous case 3, it is unclear whether it generalises to other cases. Since the investment planning problem is linear, one way to approach this problem would be to exploit the linearity with respect to price changes in the uncertain cost coefficient. If a change in price does not affect the operational policy, given some fixed investment decision, the gradient is just a linear function of the operational cost. Furthermore, it is worth noting that this problem will not occur if the uncertainties are in the right-hand-side, that is v_i^E and v_i^D.

Another dimension worth mentioning is the fact that the operational problem has always a feasible solution, as a result of the load shedding variable. Modelling the problem in such a way is very common when a decomposition algorithm is employed. However, an infeasible solution could be an indication of where to split nodes and how to distribute the nodal probabilities. For example, if there is an investment node, in which an investment decision is feasible for one successor node but not for another, this could then be an indication that the node needs to be split in order to allow for different investment decisions. Alternatively, instead of using infeasibility as an indication for splitting nodes, a large objective value of the operational subproblem could be used as indication that the node should be split. This is because the value is likely driven by the load shedding cost, which is a measure introduced to avoid infeasibility.

Furthermore, the high load shedding cost affects the number of operational leaf nodes that can be added. At this third stage, each set of nodes works with the same capacity. Because the cost associated with load shedding is so high, the model has a very high incentive to avoid it. Therefore, the investment decision is driven by the one operational node that is closest to shedding load. This is not generally wrong, however, it does put a huge emphasis on the worst-case scenarios. Case 1 demonstrated this, as a large number of operational nodes were merged from 3 into 2 nodes (the worst-case scenario plus another one). At this third stage, rather than adding or removing nodes, it might be more meaningful to adjust the probability of that worst-case scenario. For example, reducing the probability that the worst-case scenario occurs most likely has an impact on the investment decision, while an additional scenario does not.

Another issue arises when a technology reaches its capacity limit. For example, in case 1, onshore wind is at its capacity limit in all nodes, while nuclear is at its capacity limit in some nodes. The gradients with respect to these technologies have a value of zero in all investment nodes except in one (see Table 4). When case 1 is reduced, the assumption is made that the one non-zero gradient can be ignored because it should, in theory, have a value of zero as well. If this approach is to be automated, it would certainly be worth to further explore the issue of why some of these gradients have non-zero values.

4 Conclusions

The aim of this thesis has been to apply a novel Benders decomposition method to solve a large-scale investment planning problem for energy systems. By using the decomposition algorithm with adaptive oracles of [8],solving these large problems takes significantly less time compared to the original Benders decomposition algorithm. This decrease in solution time is due to the fact that the algorithm exploits the mathematical properties of the operational subproblems in order to generate valid upper and lower bounds for subproblems that are not solved in the current iteration. Although the algorithm does not rely on solving all subproblems in each iteration, it generates cutting planes for each subproblem in every iteration. As a consequence, solving the RMP becomes increasingly more difficult. Therefore, the aim of this thesis has also been to find new ways to adaptively expand and reduce the scenario tree during the solution process, so that scenarios that drive the investment decisions are identified and used to produce an investment policy.

The literature review has focused on research on scenario generation and scenario reduction methods. Although none of the papers reviewed propose a method that could be directly applied in the context of this thesis, these methods serve as an inspiration for the work that followed. For example, Dempster shows that scenario reduction can be applied iteratively during the solution process, while Casey highlights the potential of exploiting the dual variables. Another conclusion from the literature review is that optimal scenario generation or reduction is a very hard problem to solve and many algorithms essentially rely on some kind of heuristics.

The two months of work that went into exploring the problem are summarised in chapter 3. First, the investment planning problem and the method of solving it is described. Next, the idea of assessing the future value surface at different nodes of the scenario tree is outlined and various methods to extract the values needed to do so have been tested. This is more challenging than initially expected because the complexity of the model often makes understanding the values rather difficult. Eventually, one method has been identified that produces values that can be verified manually. This method has then been applied to assess the future value surface in different investment and operational nodes in the scenario tree. Based on this assessment, some nodes can be merged to create a reduced scenario tree. Case 1 and case 2 are two examples of a successful application of the method. The reduced tree in case 1 is only half the size of the original tree, while the resulting investment policy is the same as if the original tree was used. Furthermore, it can be observed that the investment policy itself is not an indication of where to merge or expand nodes, because even if the policy is similar in some nodes, the future value surface might not be.

The same method that is used to reduce the scenario tree is also used to generate scenarios. Although this should not be considered a true scenario generation method, it shows that some of the same ideas might be transferable to the generation of scenarios as well. In the last section of chapter 3, some concerns related to the developed method are summarised. One of the problems is related to the fact that uncertainty in the cost coefficient, as opposed to uncertainty in the right-hand-side, affects the value surface regardless of whether the uncertainty has an effect on the investment policy.

This thesis leaves room for future research, as the method and issues described above deserve further exploration. First, it remains unexplored if there is a more direct way to assess the future value surface in particular nodes of the tree. While it is easy to blame the complexity of the model for the failure of some methods, it would be appropriate to explore these underlying issues in more depth. Second, it is worth spending more time on assessing the actual value surfaces. In case 1 and 2, investment and operational nodes are merged because the value surfaces were exactly coinciding. No case, in which the value surfaces of different nodes are symmetric around their mean node, has been found. Especially with uncertainties in the cost coefficients, such cases should exist and examining them could offer further insights. Third, more research is required to test whether the method can be implemented in an iterative algorithm that dynamically merges and expands the scenario tree. The various graphs in the appendices show that a very tight convergence is not necessary in order to compare the gradients with respect to the investment decisions. This suggests that an iterative algorithm, which solves the problem and adjusts the scenario tree dynamically, could be possible. Finally, the concept of scenario generation remains largely unexplored. This would become of particular interest when the method is embedded in a Nested Benders algorithm - basically a decomposed tree of trees - and being able to add new scenario (trees) where needed would become highly useful.

Solving large-scale stochastic investment planning problems remains a challenging task. Since planning problems can become more flexible and, at the same time, more complex by adding more uncertainties and more stages, there will always be a problem that cannot be solved with the currently available methods. This thesis has contributed to the development of a method that dynamically explores a scenario tree in order to generate scenarios where needed and to merge scenarios that do not make a difference.

References

[1] International Energy Agency, *Report on World Energy Investment 2019*, 2019.

[2] International Energy Agency, *World Energy Model Documentation*, 2018.

[3] P. Massé, *Les Réserves et la Régulation de l'Avenir dans la vie Économique*. Hermann, Paris, 1946.

[4] S. Wallace and S.-E. Fleten, "Stochastic programming models in energy," *Handbooks in Operations Research and Management Science*, vol. 10, pp. 637–677, 12 2003.

[5] J. R. Birge and F. Louveaux, *Introduction to Stochastic Programming*. Springer New York, 2011.

[6] M. Kaut and S. Wallace, "Evaluation of scenario-generation methods for stochastic programming," *Pacific Journal of Optimization*, vol. 3, 06 2003.

[7] J. F. Benders, "Partitioning procedures for solving mixed-variables programming problems," *Numerische Mathematik*, vol. 4, pp. 238–252, Dec 1962.

[8] N. Mazzi, A. Grothey, K. McKinnon, and N. Sugishita, "Benders decomposition with adaptive oracles for large scale optimization," *Optimization Online*, 2019.

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

[10] S. Mitra and N. D. Domenica, "A review of scenario generation methods," *International Journal of Computing Science and Mathematics*, vol. 3, no. 3, p. 226, 2010.

[11] M. S. Casey and S. Sen, "The scenario generation algorithm for multistage stochastic linear programming," *Mathematics of Operations Research*, vol. 30, no. 3, pp. 615–631, 2005.

[12] J. Dupacová, Gröwe-Kuska, and W. Römisch, "Scenario reduction in stochastic programming," *Mathematical Programming*, vol. 95, pp. 493–511, Mar. 2003.

[13] N. Growe-Kuska, H. Heitsch, and W. Romisch, "Scenario reduction and scenario tree construction for power management problems," in *2003 IEEE Bologna Power Tech Conference Proceedings,*, vol. 3, pp. 7 pp. Vol.3–, June 2003.

[14] H. Heitsch and W. Römisch, “Scenario reduction algorithms in stochastic programming,” *Computational Optimization and Applications*, vol. 24, pp. 187–206, Feb 2003.

[15] H. Heitsch and W. Römisch, “A note on scenario reduction for two-stage stochastic programs,” *Oper. Res. Lett.*, vol. 35, pp. 731–738, Nov. 2007.

[16] H. Heitsch and W. Römisch, “Scenario tree reduction for multistage stochastic programs,” *Computational Management Science*, vol. 6, pp. 117–133, May 2009.

[17] `https://www.gams.com/24.8/docs/tools/scenred2/index.html`.

[18] `https://github.com/trigaut/Scenred2.jl`.

[19] M. Dempster and R. Thompson, “Evpi-based importance sampling solution proceduresfor multistage stochastic linear programmeson parallel mimd architectures,” *Annals of Operations Research*, vol. 90, pp. 161–184, Jan 1999.

[20] F. Munoz, B. Hobbs, and J.-P. Watson, “New bounding and decomposition approaches for MILP investment problems: Multi-area transmission and generation planning under policy constraints,” *European Journal of Operational Research*, vol. 248, no. 3, pp. 888–898, 2016.

[21] Q. Ploussard, L. Olmos, and A. Ramos, “An operational state aggregation technique for transmission expansion planning based on line benefits,” *IEEE Transactions on Power Systems*, vol. 32, pp. 2744–2755, July 2017.

[22] M. Sun, F. Teng, I. Konstantelos, and G. Strbac, “An objective-based scenario selection method for transmission network expansion planning with multivariate stochasticity in load and renewable energy sources,” *Energy*, vol. 145, pp. 871 – 885, 2018.

Appendices

A Operational Model: Mathematical Formulation

Sets

$\mathcal{P}$ = Set of p technologies
$\mathcal{G}$ = Set of g conventional technologies
$\mathcal{B}$ = Set of b storage technologies
$\mathcal{R}$ = Set of r renewable technologies
$\mathcal{S}$ = Set of s seasons
$\mathcal{H}$ = Set of h hours within a season

Variables

x_p = installed capacity of technology p (MW)
y^G_{bsh} = power generation from conventional unit g at hour h of season s (MW)
y^I_{bsh} = charging power of storage unit b at hour h of season s (MW)
y^O_{bsh} = discharging power of storage unit b at hour h of season s (MW)
y^L_{bsh} = energy level of storage unit b at hour h of season s (MWh)
y^S_{sh} = shedded demand at hour h of season s (MW)

Objective function

$$\min \quad \sum_{s\in\mathcal{S}}\Big(\sum_{h\in\mathcal{H}}\Big(\sum_{g\in\mathcal{G}}\Big(\Big(C^{OMvar}_g + \frac{C^{emCO_2}_g + C^{fuel}_g}{\eta^G_g}\Big)\cdot y^G_{gsh}\Big) + C^{shed}_{sh} y^S_{sh}\Big)\cdot\alpha\Big)$$

Inventory constraint for storages

$$y^L_{bsh} - y^L_{bsh\text{-}1} = \eta^B_b \cdot y^I_{bsh} - y^O_{bsh} \quad \forall b\in\mathcal{B}, s\in\mathcal{S}, h\in\mathcal{H}$$

Ramp up/down constraint

$$y^G_{gsh} - y^G_{gsh\text{-}1} \le ramp^G_g \cdot x_g \quad \forall g\in\mathcal{G}, s\in\mathcal{S}, h\in\mathcal{H}$$
$$y^G_{gsh\text{-}1} - y^G_{gsh} \le ramp^G_g \cdot x_g \quad \forall g\in\mathcal{G}, s\in\mathcal{S}, h\in\mathcal{H}$$

Capacity constraints

$$
\begin{aligned}
y_{gsh}^{G} &\le x_g \quad \forall g \in \mathcal{G}, s \in \mathcal{S}, h \in \mathcal{H} \\
y_{bsh}^{L} &\le x_b \quad \forall b \in \mathcal{B}, s \in \mathcal{S}, h \in \mathcal{H}
\end{aligned}
$$

Charging discharging capacity constraints

$$
\begin{aligned}
y_{bsh}^{I} &\le P_b^B \cdot x_b \quad \forall b \in \mathcal{B}, s \in \mathcal{S}, h \in \mathcal{H} \\
y_{bsh}^{O} &\le P_b^B \cdot x_b \quad \forall b \in \mathcal{B}, s \in \mathcal{S}, h \in \mathcal{H}
\end{aligned}
$$

Emission limit

$$
\sum_{g \in \mathcal{G}} \sum_{s \in \mathcal{S}} \sum_{h \in \mathcal{H}} \frac{y_{gsh}^{G} \cdot EM_g^{CO_2}}{\eta_g^G} \le CO_2^{lim}
$$

Energy balance

$$
\sum_{g \in \mathcal{G}} y_{gsh}^{G} + \sum_{b \in \mathcal{B}} y_{bsh}^{O} + y_{sh}^{S} \ge P_{sh}^{D} - \sum_{r \in \mathcal{R}} P_{rsh}^{R} \quad \forall s \in \mathcal{S}, h \in \mathcal{H}
$$

Parameters

C_g^{OMvar} = Variable operating cost of conventional technology g (£/MWh)

C_g^{fuel} = Fuel cost of conventional technology g (£/MWh)

$EM_g^{CO_2}$ = CO_2 emissions of conventional technology g (tCO_2/MWh)

η_g^G = Efficiency of conventional technology g

η_b^B = Efficiency of storage b

$ramp_g^G$ = Ramping limitations of conventional technology g (MW)

P_b^B = Charging / discharging power of storage b (MW)

P_r^R = Renewable energy production of renewable technology r (MW)

P^D = Energy demand (MW)

C^{shed} = Cost of shedding demand (£/MW)

α = Seasonal weight

$C_g^{emCO_2}$ = CO_2 cost of conventional technology g (£/tCO_2)

CO_2^{lim} = Yearly CO_2 limit (tCO_2)

B Operational Model: Technologies

Technology	η	$C_g^{emCO_2}$	C_g^{OMvar}	C_g^{fuel}	$C^{OMfix\ 1)}$	$C^{inv\ 2)}$
Coal generator	0.39	0.318	3.17	5.74	26,840	595,142
Coal generator with CCS	0.28	0.032	6.75	5.74	47,170	958,285
Open Cycle Gas Turbine (OCGT)	0.35	0.181	7.36	12.11	5,000	160,000
Combined Cycle Gas Turbine (CCGT)	0.53	0.181	2.32	12.11	10,910	242,000
Diesel generator	0.34	0.250	0.00	28.10	3,000	128,000
Nuclear power plant	0.33	0.000	1.52	40.00	66,230	981,500
Pumped hydroelectric storage (low capacity)	0.80	-	-	-	10,500	640,000
Pumped hydroelectric storage (high capacity)	0.80	-	-	-	12,780	750,800
Lithium-ion battery storage	0.92	-	-	-	1,500	500,000
Onshore Wind	-	-	-	-	28,080	314,200
Offshore Wind	-	-	-	-	52,540	884,600
Solar photovoltaics	-	-	-	-	17,530	1,832,666

1) Fix operational cost (£/GWyr)
2) Investment cost at the first investment stage (£/GW)

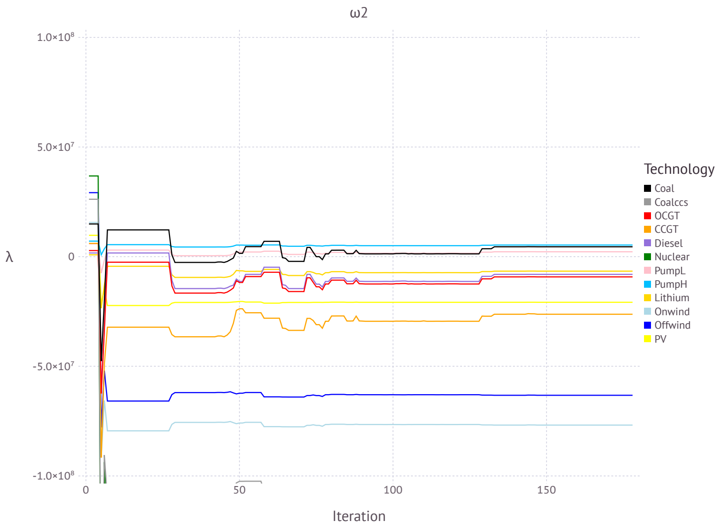
C Gradients of Original Inventory Constraint (Equation 2)

Iteration 0

	ω2	ω3	ω4	ω5	ω6	ω7	ω8	ω9	ω10
Coal	1.491e7	1.491e7	1.491e7	1.491e7	1.491e7	1.491e7	1.491e7	1.491e7	1.657e6
Coal&CCS	2.621e7	2.621e7	2.621e7	2.621e7	2.621e7	2.621e7	2.621e7	2.621e7	2.912e6
OCGT	2.778e6	2.778e6	2.778e6	2.778e6	2.778e6	2.778e6	2.778e6	2.778e6	308600.0
CCGT	6.061e6	6.061e6	6.061e6	6.061e6	6.061e6	6.061e6	6.061e6	6.061e6	673500.0
Diesel	1.667e6	1.667e6	1.667e6	1.667e6	1.667e6	1.667e6	1.667e6	1.667e6	185200.0
Nuclear	3.679e7	3.679e7	3.679e7	3.679e7	3.679e7	3.679e7	3.679e7	3.679e7	4.088e6
PumpL	5.833e6	5.833e6	5.833e6	5.833e6	5.833e6	5.833e6	5.833e6	5.833e6	648100.0
PumpH	7.1e6	7.1e6	7.1e6	7.1e6	7.1e6	7.1e6	7.1e6	7.1e6	788900.0
Lithium	833300.0	833300.0	833300.0	833300.0	833300.0	833300.0	833300.0	833300.0	92590.0
Onwind	1.56e7	1.56e7	1.56e7	1.56e7	1.56e7	1.56e7	1.56e7	1.56e7	1.733e6
Offwind	2.919e7	2.919e7	2.919e7	2.919e7	2.919e7	2.919e7	2.919e7	2.919e7	3.243e6
PV	9.739e6	9.739e6	9.739e6	9.739e6	9.739e6	9.739e6	9.739e6	9.739e6	1.082e6

Final iteration

	ω2	ω3	ω4	ω5	ω6	ω7	ω8	ω9	ω10
Coal	4.509e6	4.509e6	814300.0	822300.0	838600.0	2.335e6	-82100.0	-82100.0	1.653e6
Coal&CCS	-1.049e8	-1.049e8	-5.539e7	-5.532e7	-5.53e7	378300.0	-2.043e6	-2.043e6	-7.599e6
OCGT	-9.225e6	-9.225e6	-1.086e7	-1.085e7	-1.084e7	-8.43e6	-1.084e7	-1.084e7	225700.0
CCGT	-2.627e7	-2.627e7	-2.038e7	-2.034e7	-2.028e7	-7.835e6	-1.026e7	-1.026e7	55640.0
Diesel	-8.073e6	-8.073e6	-1.138e7	-1.138e7	-1.138e7	-9.059e6	-1.145e7	-1.145e7	185200.0
Nuclear	-1.273e8	-1.273e8	-5.578e7	-5.572e7	-5.569e7	2.206e7	1.964e7	1.964e7	-1.114e7
PumpL	2.224e6	2.224e6	2.553e6	2.555e6	2.561e6	3.702e6	3.268e6	3.268e6	344500.0
PumpH	5.299e6	5.299e6	5.54e6	5.541e6	5.544e6	6.133e6	5.94e6	5.94e6	619500.0
Lithium	-6.604e6	-6.604e6	-5.781e6	-5.776e6	-5.762e6	-3.668e6	-4.154e6	-4.154e6	-1.288e7
Onwind	-7.682e7	-7.682e7	-5.536e7	-5.533e7	-5.534e7	-3.365e7	-3.377e7	-3.377e7	3.686e7
Offwind	-6.323e7	-6.323e7	-4.177e7	-4.174e7	-4.175e7	-2.006e7	-2.018e7	-2.018e7	-6.136e6
PV	-2.08e7	-2.08e7	-1.359e7	-1.359e7	-1.359e7	-6.25e6	-6.278e6	-6.278e6	-2.341e6



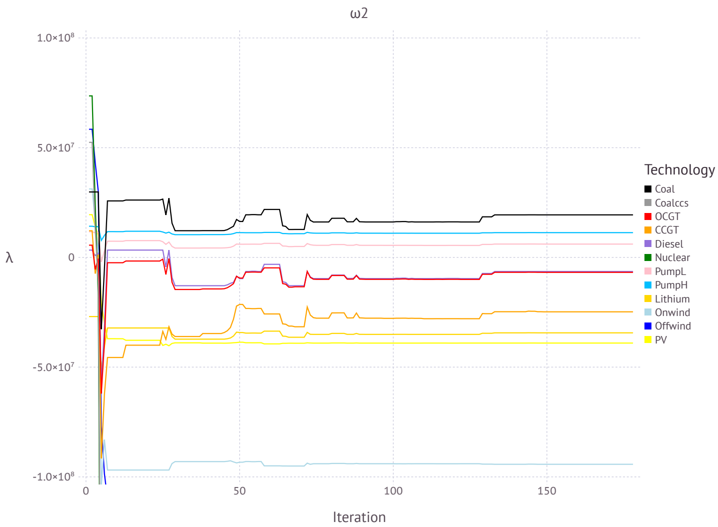
D Gradients of new Inventory Constraints (Equations 5 and 6)

Equation 5 iteration 0

	$\omega 2$	$\omega 3$	$\omega 4$	$\omega 5$	$\omega 6$	$\omega 7$	$\omega 8$	$\omega 9$	$\omega 10$
Coal	2.982e7	2.982e7	2.982e7	2.982e7	2.982e7	2.982e7	2.982e7	2.982e7	-8.337e8
Coal&CCS	5.241e7	5.241e7	5.241e7	5.241e7	5.241e7	5.241e7	5.241e7	5.241e7	-1.378e9
OCGT	5.556e6	5.556e6	5.556e6	5.556e6	5.556e6	5.556e6	5.556e6	5.556e6	-2.044e8
CCGT	1.212e7	1.212e7	1.212e7	1.212e7	1.212e7	1.212e7	1.212e7	1.212e7	-3.39e8
Diesel	3.333e6	3.333e6	3.333e6	3.333e6	3.333e6	3.333e6	3.333e6	3.333e6	-1.547e8
Nuclear	7.359e7	7.359e7	7.359e7	7.359e7	7.359e7	7.359e7	7.359e7	7.359e7	-1.57e9
PumpL	1.167e7	1.167e7	1.167e7	1.167e7	1.167e7	1.167e7	1.167e7	1.167e7	-7.333e8
PumpH	1.42e7	1.42e7	1.42e7	1.42e7	1.42e7	1.42e7	1.42e7	1.42e7	-8.644e8
Lithium	-2.694e7	-2.694e7	-2.694e7	-2.694e7	-2.694e7	-2.694e7	-2.694e7	-2.694e7	-2.694e7
Onwind	3.12e7	3.12e7	3.12e7	3.12e7	3.12e7	3.12e7	3.12e7	3.12e7	-5.638e8
Offwind	5.838e7	5.838e7	5.838e7	5.838e7	5.838e7	5.838e7	5.838e7	5.838e7	-1.352e9
PV	1.948e7	1.948e7	1.948e7	1.948e7	1.948e7	1.948e7	1.948e7	1.948e7	-1.988e9

Equation 5 final iteration

	$\omega 2$	$\omega 3$	$\omega 4$	$\omega 5$	$\omega 6$	$\omega 7$	$\omega 8$	$\omega 9$	$\omega 10$
Coal	1.939e7	1.939e7	1.939e7	5.134e6	5.285e6	5.859e6	3.727e6	3.747e6	-6.771e8
Coal&CCS	-1.581e8	-1.581e8	-1.581e8	-1.075e8	-1.069e8	-1.085e8	-5.422e7	-5.384e7	-5.286e7
OCGT	-6.82e6	-6.82e6	-6.82e6	-1.824e7	-1.815e7	-1.792e7	-1.949e7	-1.954e7	-1.663e7
CCGT	-2.48e7	-2.48e7	-2.48e7	-3.39e7	-3.372e7	-3.274e7	-2.37e7	-2.37e7	-1.983e7
Diesel	-6.406e6	-6.406e6	-6.406e6	-1.85e7	-1.85e7	-1.85e7	-1.856e7	-1.856e7	-1.617e7
Nuclear	-1.819e8	-1.819e8	-1.819e8	-1.104e8	-1.102e8	-1.102e8	-3.489e7	-3.489e7	-3.247e7
PumpL	6.079e6	6.079e6	6.079e6	5.882e6	5.727e6	5.494e6	6.528e6	6.337e6	-6.882e8
PumpH	1.13e7	1.13e7	1.13e7	1.142e7	1.134e7	1.121e7	1.178e7	1.167e7	-8.421e8
Lithium	-3.438e7	-3.438e7	-3.438e7	-3.357e7	-3.354e7	-3.354e7	-3.193e7	-3.193e7	-3.145e7
Onwind	-9.427e7	-9.427e7	-9.427e7	-7.282e7	2.678e8	-7.278e7	-5.123e7	-5.123e7	-5.11e7
Offwind	-1.116e8	-1.116e8	-1.116e8	-8.442e7	-8.382e7	-8.453e7	-6.283e7	-6.229e7	-1.719e8
PV	-3.902e7	-3.902e7	-3.902e7	-2.756e7	-2.769e7	-2.844e7	-2.022e7	-2.035e7	-1.591e9

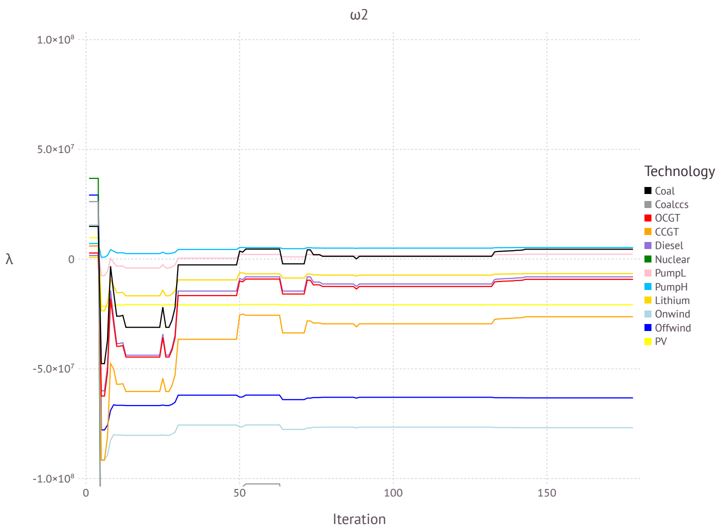


Equation 6 iteration 0

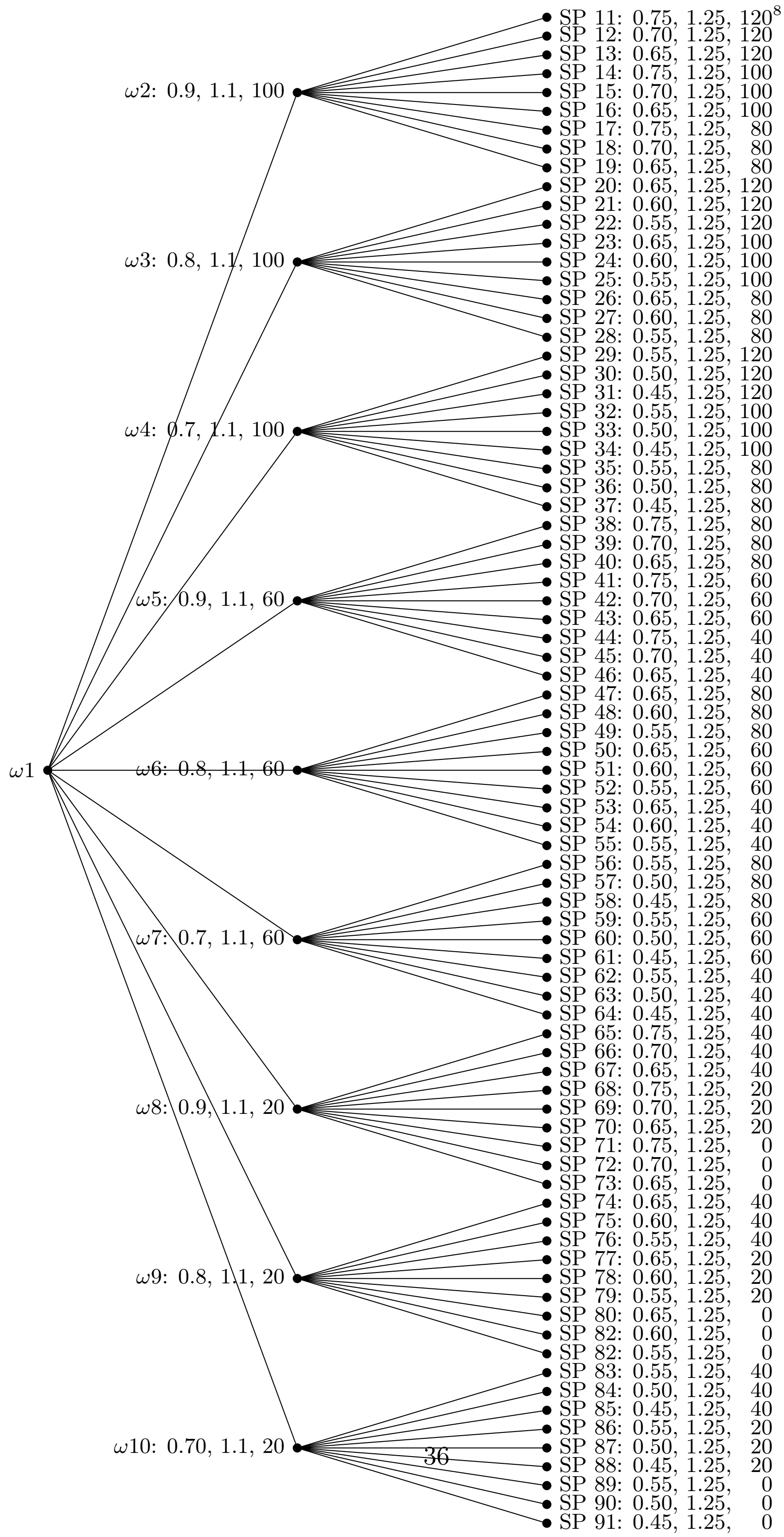
	$\omega 2$	$\omega 3$	$\omega 4$	$\omega 5$	$\omega 6$	$\omega 7$	$\omega 8$	$\omega 9$	$\omega 10$
Coal	1.491e7	1.491e7	1.491e7	1.491e7	1.491e7	1.491e7	1.491e7	1.491e7	1.657e6
Coal&CCS	2.621e7	2.621e7	2.621e7	2.621e7	2.621e7	2.621e7	2.621e7	2.621e7	2.912e6
OCGT	2.778e6	2.778e6	2.778e6	2.778e6	2.778e6	2.778e6	2.778e6	2.778e6	308600.0
CCGT	6.061e6	6.061e6	6.061e6	6.061e6	6.061e6	6.061e6	6.061e6	6.061e6	673500.0
Diesel	1.667e6	1.667e6	1.667e6	1.667e6	1.667e6	1.667e6	1.667e6	1.667e6	185200.0
Nuclear	3.679e7	3.679e7	3.679e7	3.679e7	3.679e7	3.679e7	3.679e7	3.679e7	4.088e6
PumpL	5.833e6	5.833e6	5.833e6	5.833e6	5.833e6	5.833e6	5.833e6	5.833e6	648100.0
PumpH	7.1e6	7.1e6	7.1e6	7.1e6	7.1e6	7.1e6	7.1e6	7.1e6	788900.0
Lithium	833300.0	833300.0	833300.0	833300.0	833300.0	833300.0	833300.0	833300.0	92590.0
Onwind	1.56e7	1.56e7	1.56e7	1.56e7	1.56e7	1.56e7	1.56e7	1.56e7	1.733e6
Offwind	2.919e7	2.919e7	2.919e7	2.919e7	2.919e7	2.919e7	2.919e7	2.919e7	3.243e6
PV	9.739e6	9.739e6	9.739e6	9.739e6	9.739e6	9.739e6	9.739e6	9.739e6	1.082e6

Equation 6 final iteration

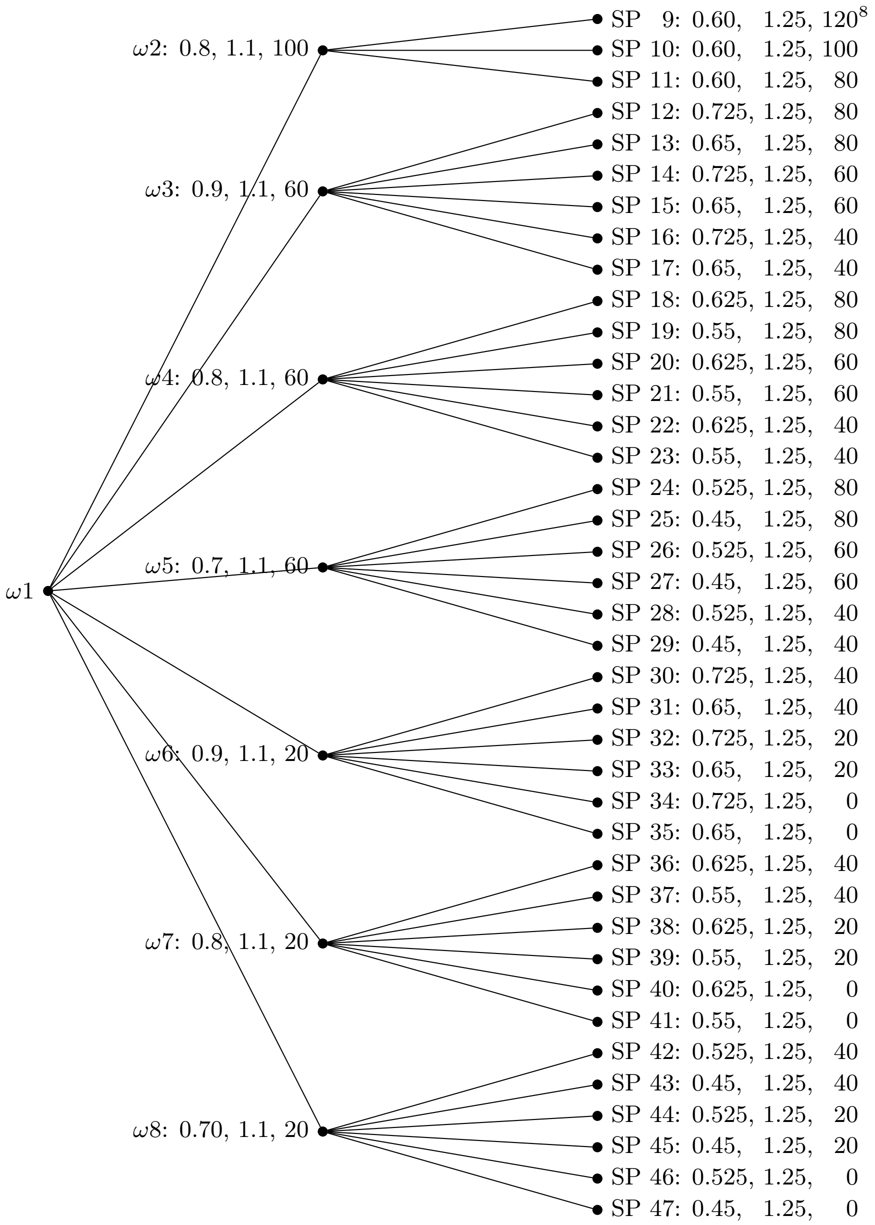
	$\omega 2$	$\omega 3$	$\omega 4$	$\omega 5$	$\omega 6$	$\omega 7$	$\omega 8$	$\omega 9$	$\omega 10$
Coal	4.509e6	4.509e6	798900.0	838600.0	837700.0	-82100.0	-82100.0	2.335e6	1.653e6
Coal&CCS	-1.049e8	-1.049e8	-5.545e7	-5.53e7	-5.526e7	-2.043e6	-2.043e6	378300.0	-7.599e6
OCGT	-9.225e6	-9.225e6	-1.087e7	-1.084e7	-1.084e7	-1.084e7	-1.084e7	-8.43e6	225700.0
CCGT	-2.627e7	-2.627e7	-2.045e7	-2.028e7	-2.028e7	-1.026e7	-1.026e7	-7.835e6	55640.0
Diesel	-8.073e6	-8.073e6	-1.138e7	-1.138e7	-1.138e7	-1.145e7	-1.145e7	-9.059e6	185200.0
Nuclear	-1.273e8	-1.273e8	-5.584e7	-5.569e7	-5.566e7	1.964e7	1.964e7	2.206e7	-1.114e7
PumpL	2.224e6	2.224e6	2.548e6	2.561e6	2.559e6	3.268e6	3.268e6	3.702e6	344500.0
PumpH	5.299e6	5.299e6	5.538e6	5.544e6	5.543e6	5.94e6	5.94e6	6.133e6	619500.0
Lithium	-6.604e6	-6.604e6	-5.792e6	-5.762e6	-5.764e6	-4.154e6	-4.154e6	-3.668e6	-1.288e7
Onwind	-7.682e7	-7.682e7	-5.537e7	-5.534e7	-5.532e7	-3.377e7	-3.377e7	-3.365e7	-7.645e6
Offwind	-6.323e7	-6.323e7	-4.178e7	-4.175e7	-4.173e7	-2.018e7	-2.018e7	-2.006e7	-6.136e6
PV	-2.08e7	-2.08e7	-1.36e7	-1.359e7	-1.359e7	-6.278e6	-6.278e6	-6.25e6	-2.341e6



E Case 1: Scenario Tree

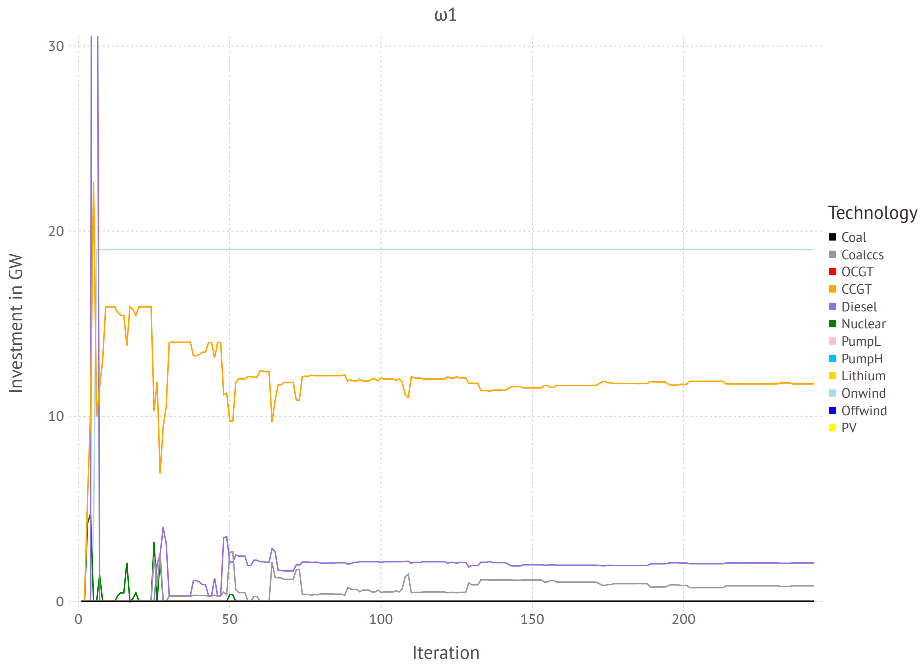


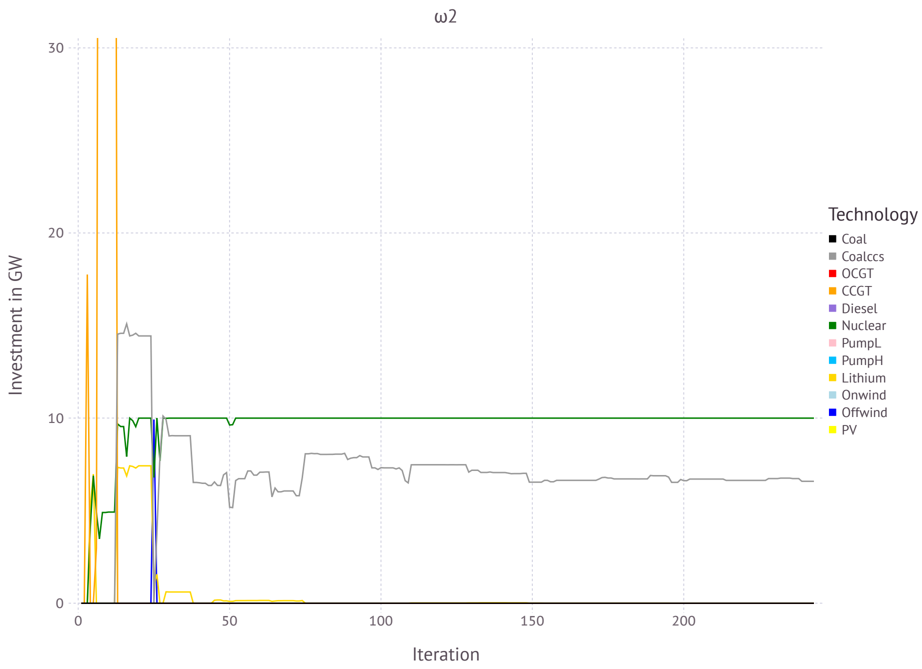
F Case 1: Reduced Scenario Tree

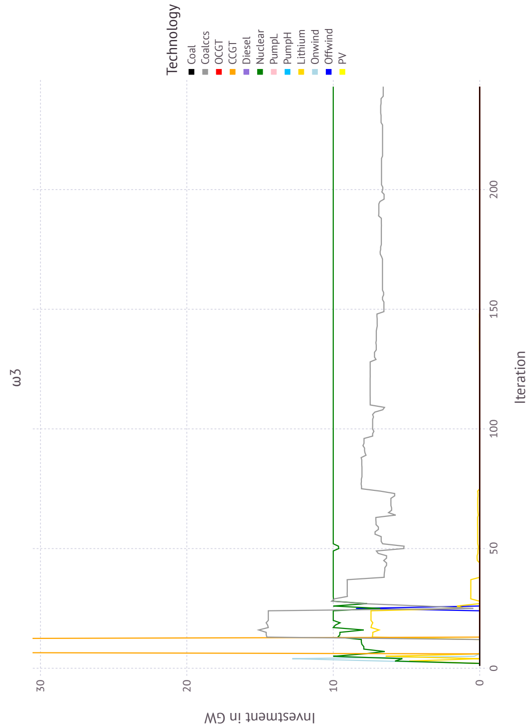


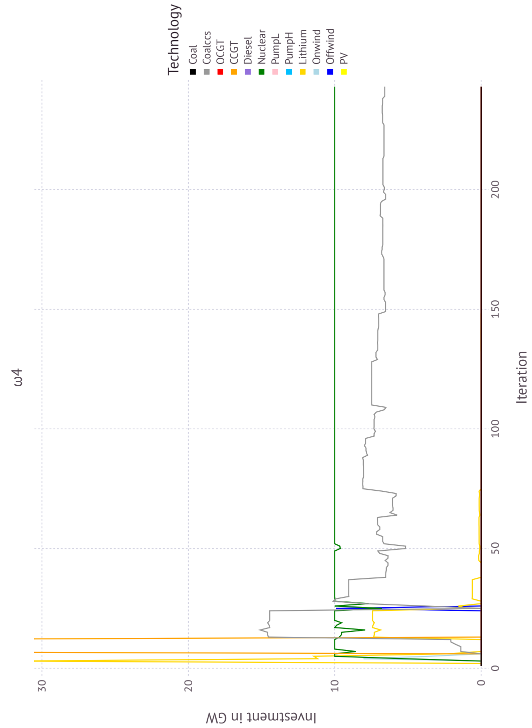
[8] Uncertainty parameters: v_i^E, v_i^D, $c_i^{co_2}$

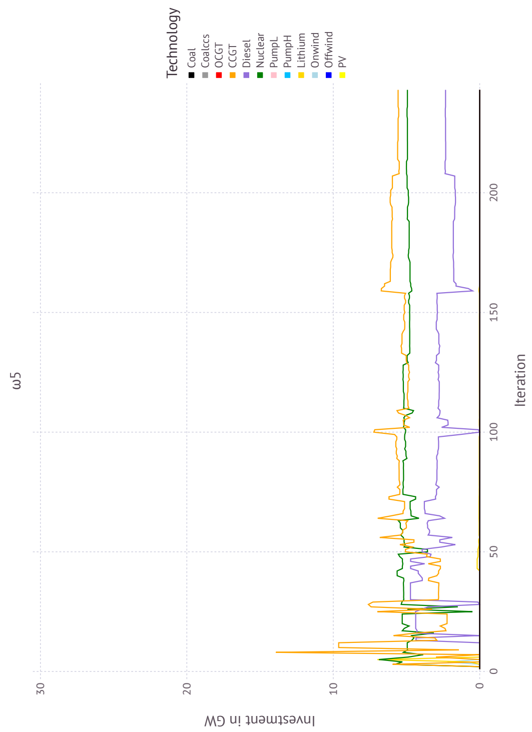
G Case 1: Investment Decisions

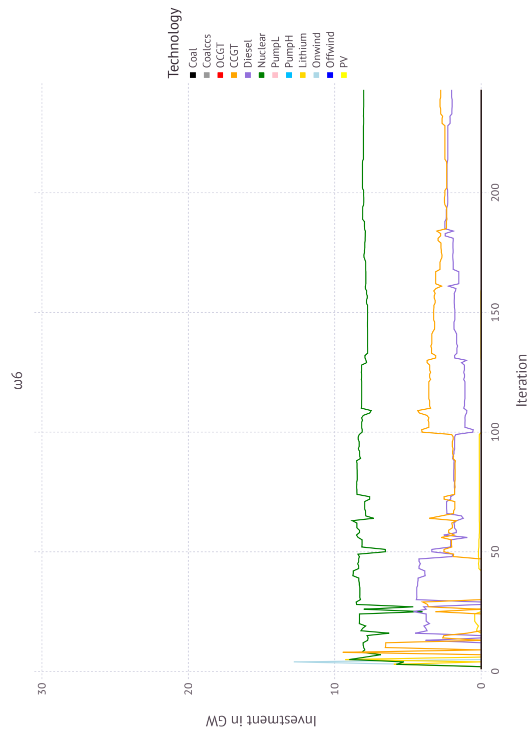


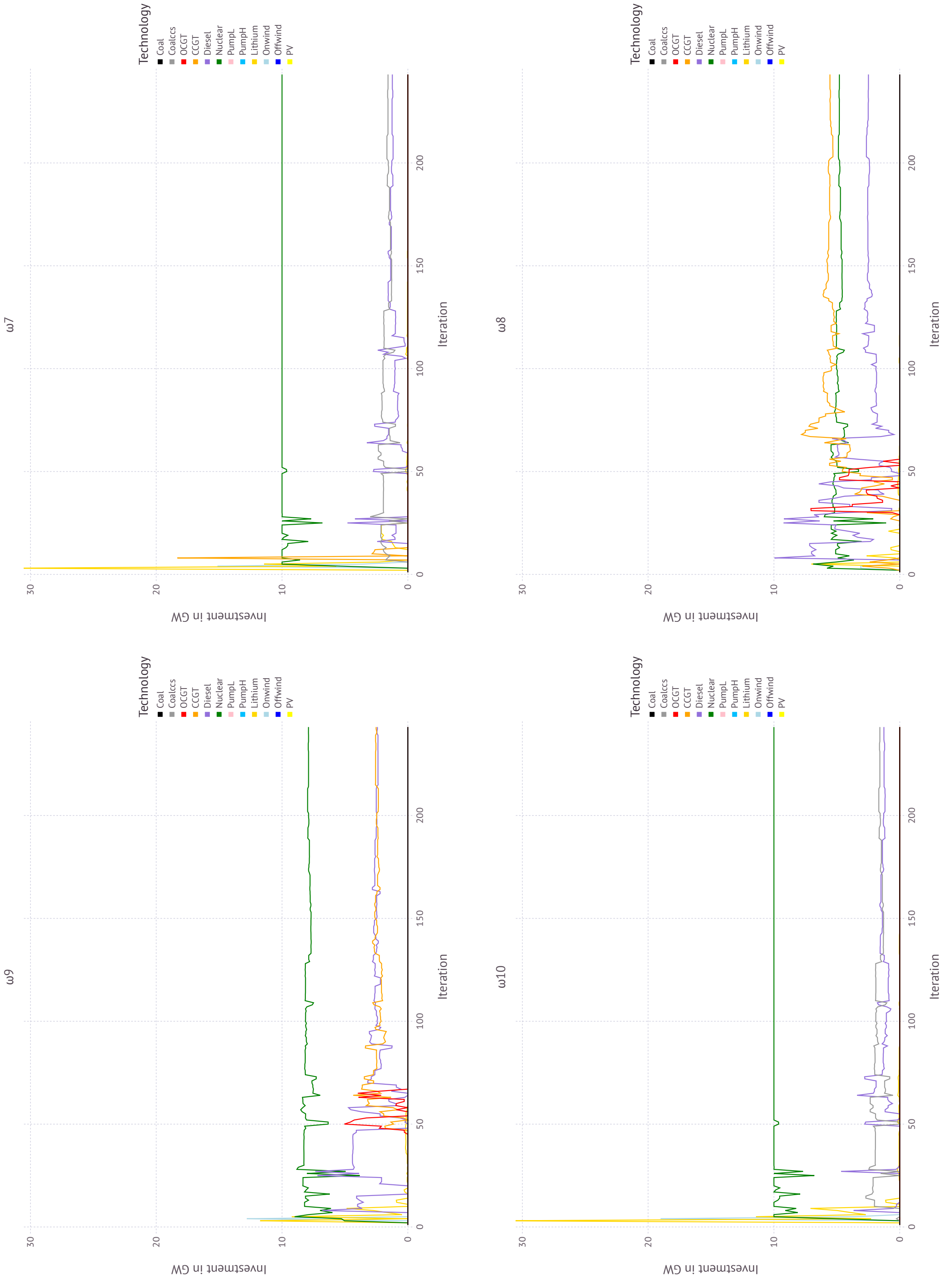



ω3
Technology
Coal
Coalccs
OCGT
CCGT
Diesel
Nuclear
PumpL
PumpH
Lithium
Onwind
Offwind
PV
Investment in GW
Iteration

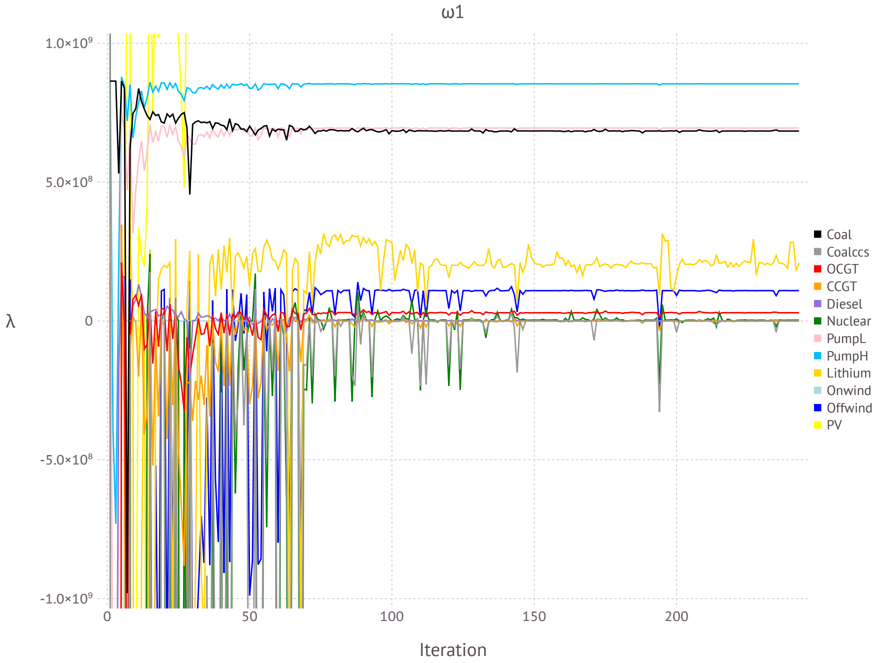

ω4
Technology
Coal
Coalccs
OCGT
CCGT
Diesel
Nuclear
PumpL
PumpH
Lithium
Onwind
Offwind
PV
Investment in GW
Iteration

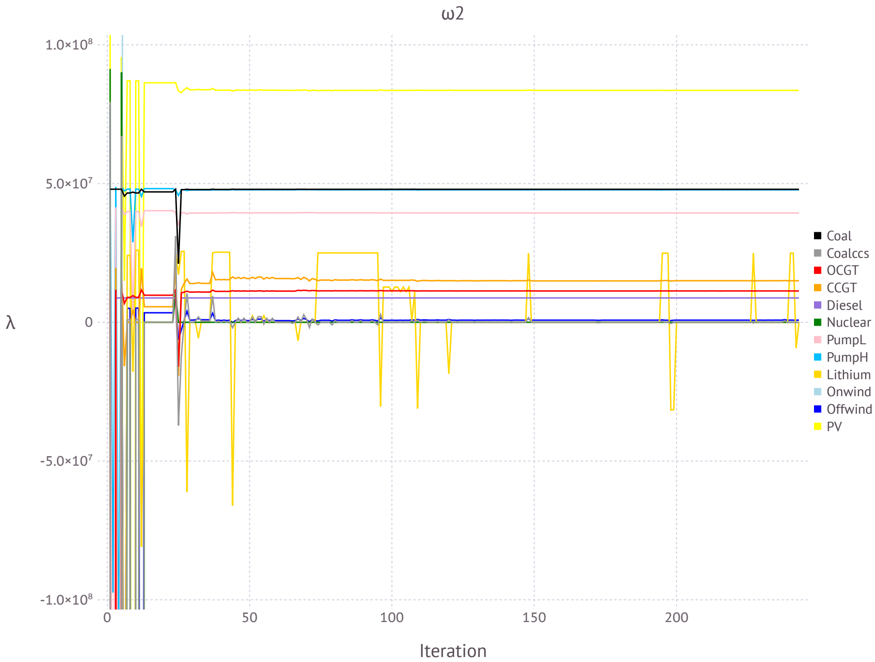

ω5
Technology
Coal
Coalccs
OCGT
CCGT
Diesel
Nuclear
PumpL
PumpH
Lithium
Onwind
Offwind
PV
Investment in GW
Iteration

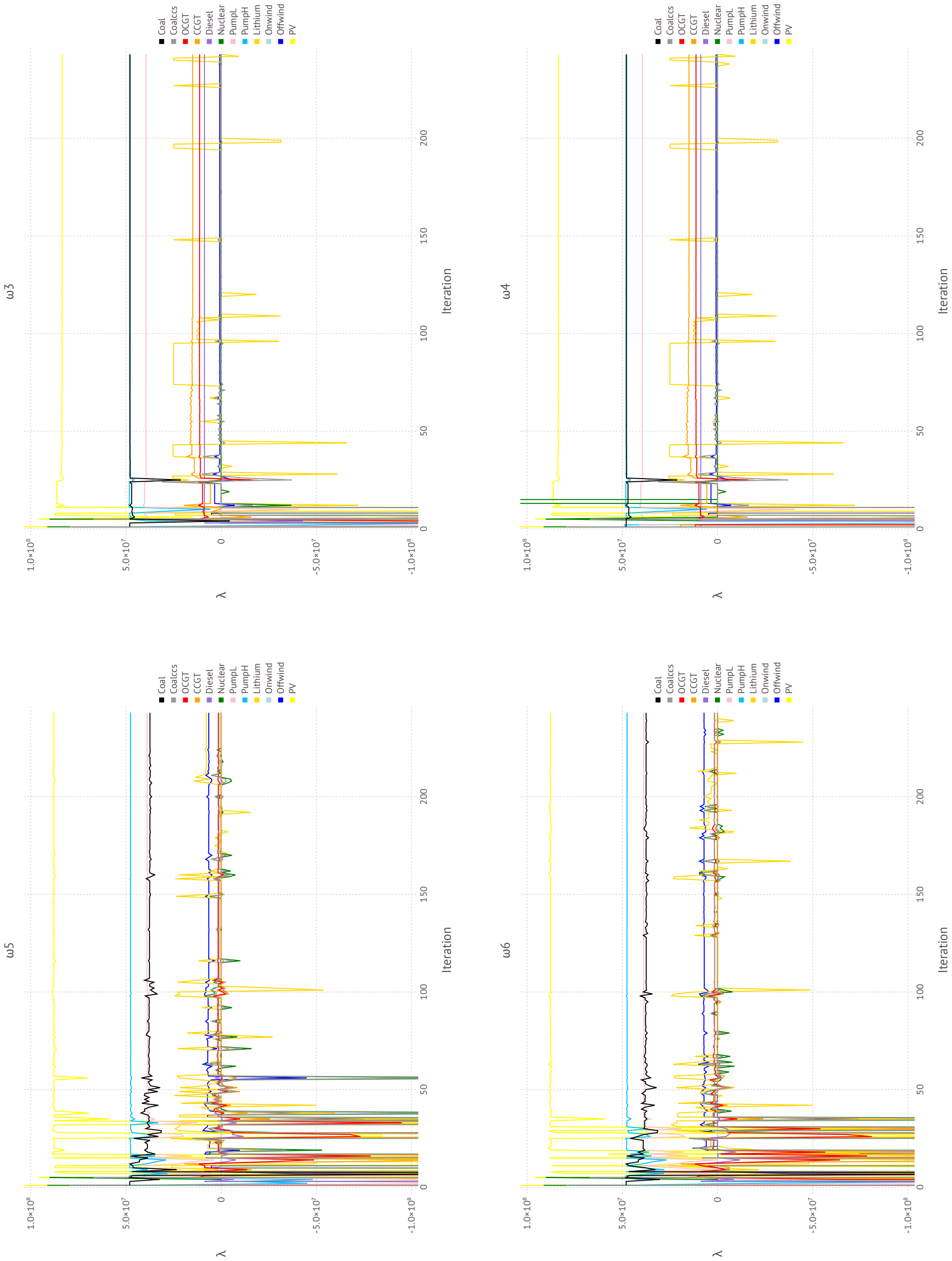

ω6
Technology
Coal
Coalccs
OCGT
CCGT
Diesel
Nuclear
PumpL
PumpH
Lithium
Onwind
Offwind
PV
Investment in GW
Iteration

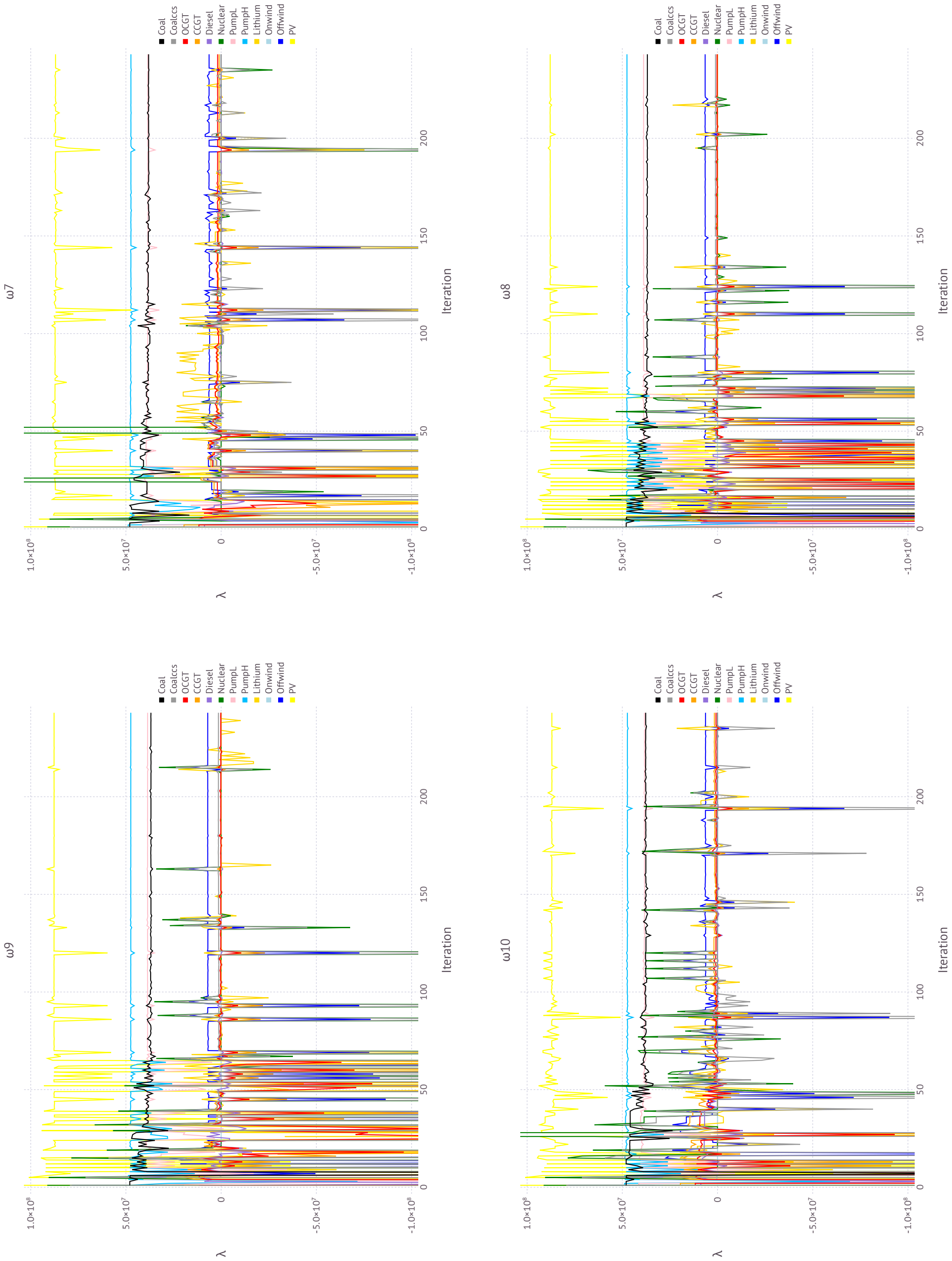

ω9
ω7
ω10
ω8
Investment in GW
Iteration
Technology
Coal
Coalccs
OCGT
CCGT
Diesel
Nuclear
PumpL
PumpH
Lithium
Onwind
Offwind
PV

H Case 1: Gradients w.r.t. Investment Decisions

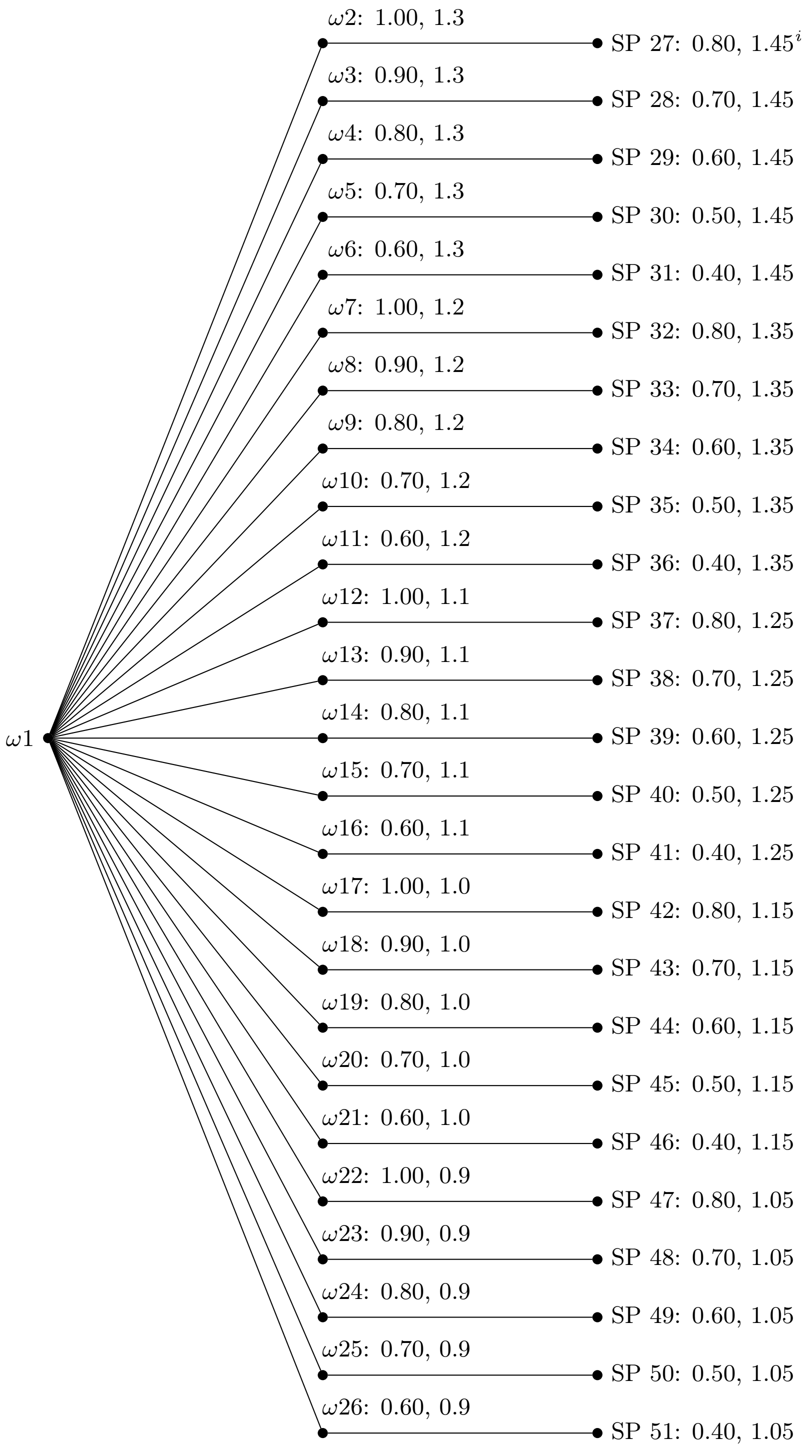





ω3
ω4
ω5
ω6
Iteration
λ
Coal
Coalccs
OCGT
CCGT
Diesel
Nuclear
PumpL
PumpH
Lithium
Onwind
Offwind
PV

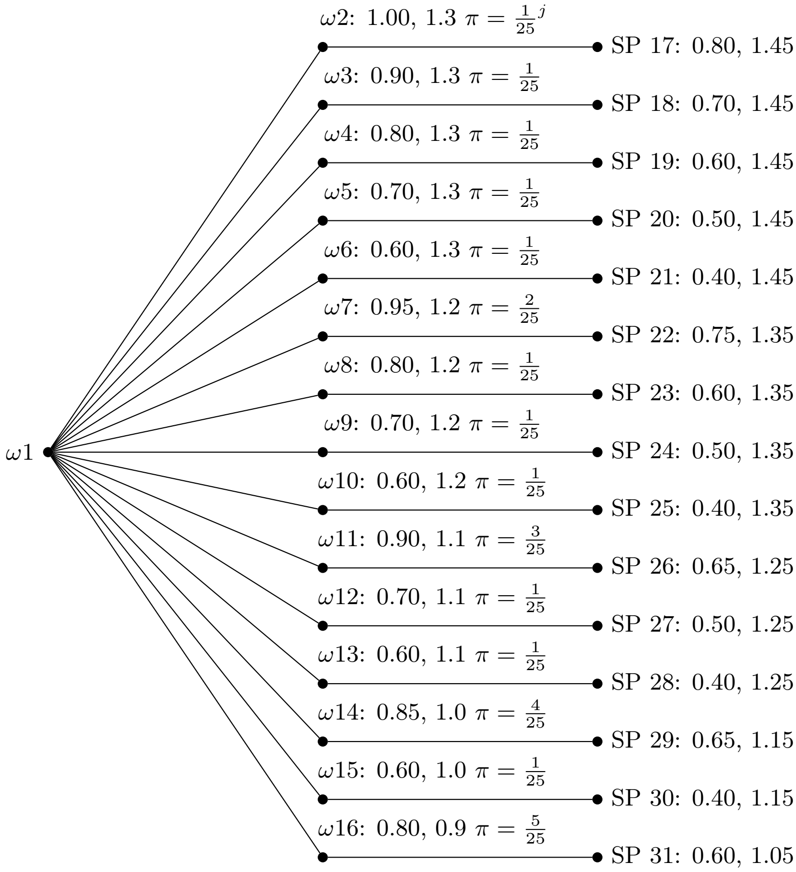

ω7
ω8
ω9
ω10
Iteration
λ
1.0×10⁸
5.0×10⁷
0
-5.0×10⁷
-1.0×10⁸
0
50
100
150
200
Coal
Coalccs
OCGT
CCGT
Diesel
Nuclear
PumpL
PumpH
Lithium
Onwind
Offwind
PV

I Case 2: Scenario Tree



[i] Uncertainty parameters: v_i^E, v_i^D

J Case 2: Reduced Scenario Tree



[j] Uncertainty parameters: v_i^E, v_i^D, π = Probability of scenario occurrence

K Case 2: Installed Capacity

	$\omega1$	$\omega2$	$\omega3$	$\omega4$	$\omega5$	$\omega6$	$\omega7$	$\omega8$	$\omega9$	$\omega10$	$\omega11$	$\omega12$	$\omega13$	$\omega14$	$\omega15$	$\omega16$
Coal																
Coal&CCS			0.57	5.3	10.4	16.08			0.35	5.25	10.38				0.25	5.07
OCGT																
CCGT	9.03	13.66	12.49	7.92	3.32		9.3	9.3	8.44	3.64		4.85	4.85	4.85	4.09	
Diesel	4.12	1.87	1.38	1.22	0.71		1.17	1.17	0.62	0.51		0.65	0.65	0.65		
Nuclear	8.89		1.11	1.11	1.11	1.11			1.11	1.11	1.11				1.11	1.11
PumpL																
PumpH																
Lithium																
Onwind	19.0															
Offwind						0.01										
PV																

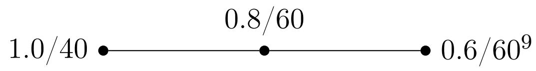
continued:

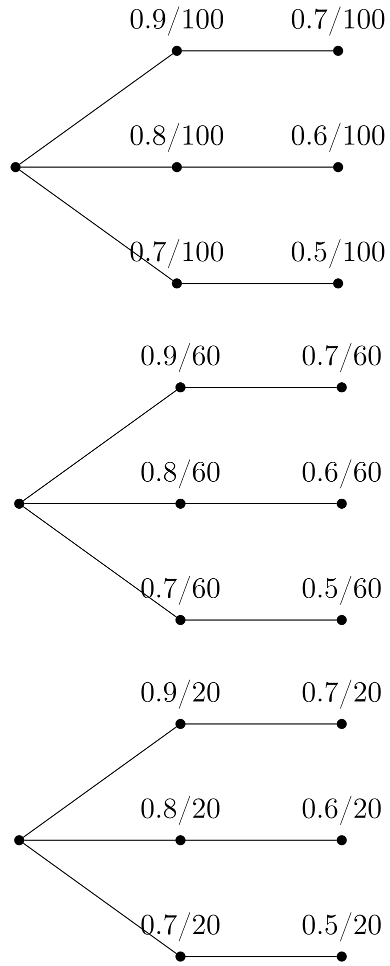
	$\omega17$	$\omega18$	$\omega19$	$\omega20$	$\omega21$	$\omega22$	$\omega23$	$\omega24$	$\omega25$	$\omega26$
Coal										
Coal&CCS					0.24					
OCGT										
CCGT	0.34	0.34	0.34	0.34						
Diesel										
Nuclear					1.11					
PumpL										
PumpH										
Lithium										
Onwind										
Offwind										
PV										

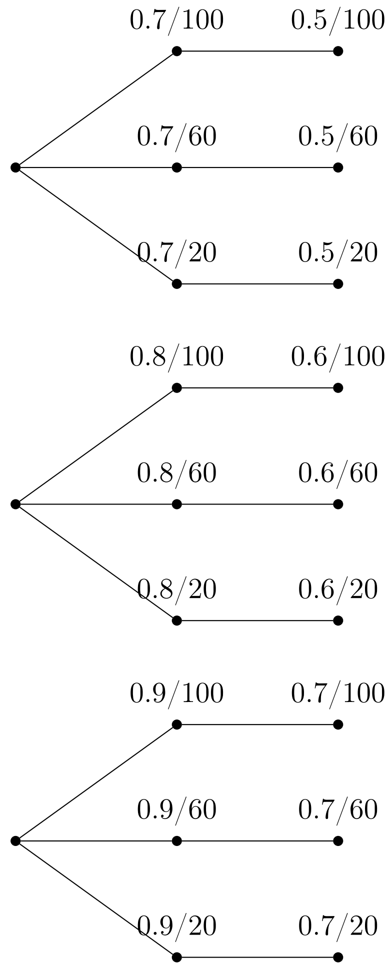
L Case 2: Installed Capacity in Reduced Tree

	$\omega1$	$\omega2$	$\omega3$	$\omega4$	$\omega5$	$\omega6$	$\omega7$	$\omega8$	$\omega9$	$\omega10$	$\omega11$	$\omega12$	$\omega13$	$\omega14$	$\omega15$	$\omega16$
Coal																
Coal&CCS			0.56	5.32	10.39	16.07		0.35	5.22	10.38		0.25	5.07		0.24	
OCGT																
CCGT	9.04	13.42	12.55	7.77	3.35		9.2	8.44	3.89		5.07	4.08		0.33		
Diesel	4.11	2.13	1.34	1.34	0.68		1.27	0.6	0.28		0.37					
Nuclear	8.89		1.11	1.11	1.11	1.11		1.11	1.11	1.11		1.11	1.11		1.11	
PumpL																
PumpH																
Lithium																
Onwind	19.0															
Offwind						0.03										
PV																

M Case 3: Expanding the Deterministic Tree



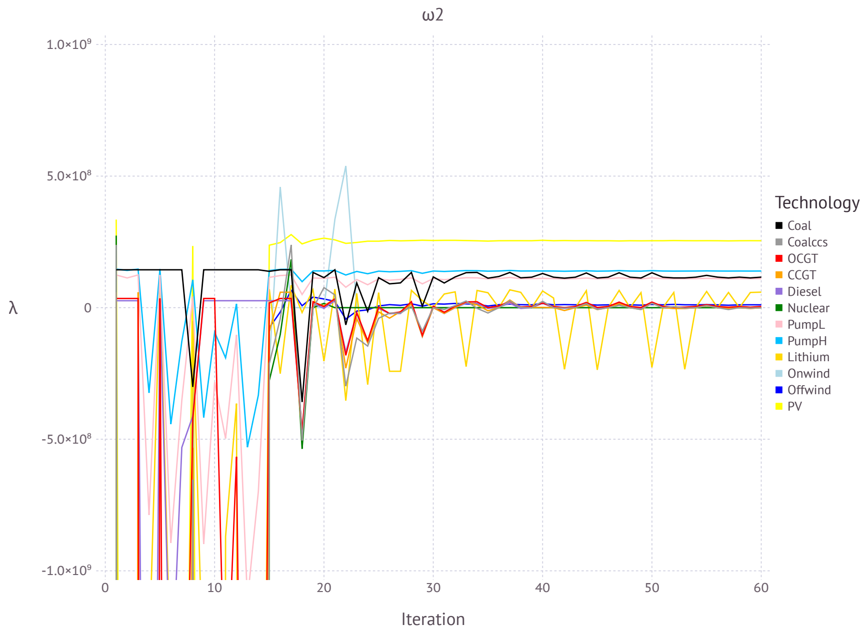


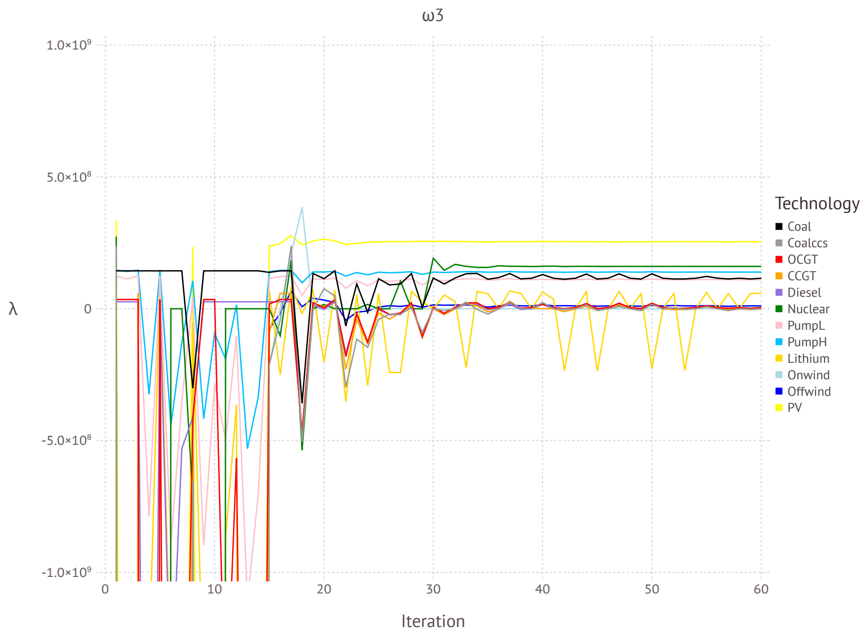


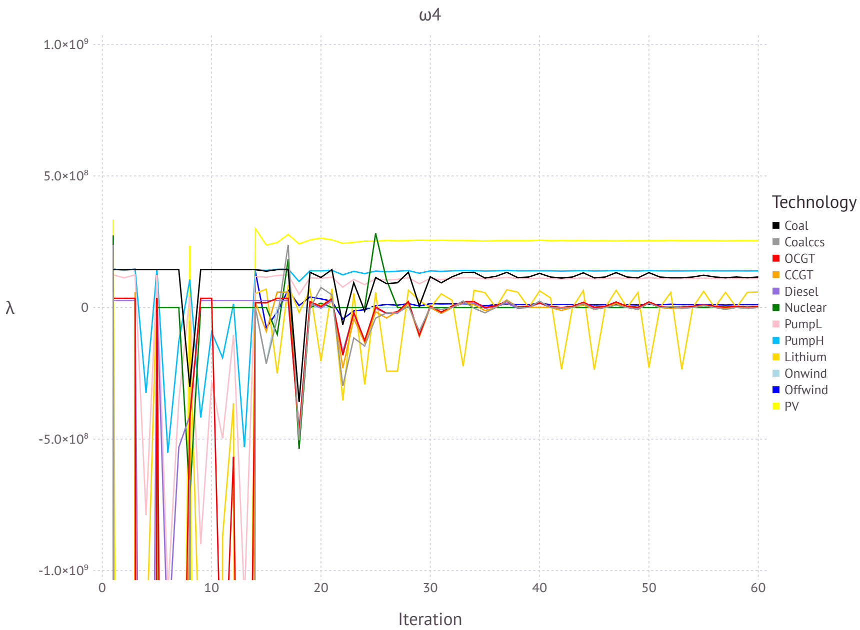
[9] Uncertainty parameters: v_i^E, $c_i^{co_2}$

N Case 3: Gradients w.r.t Investment Decisions

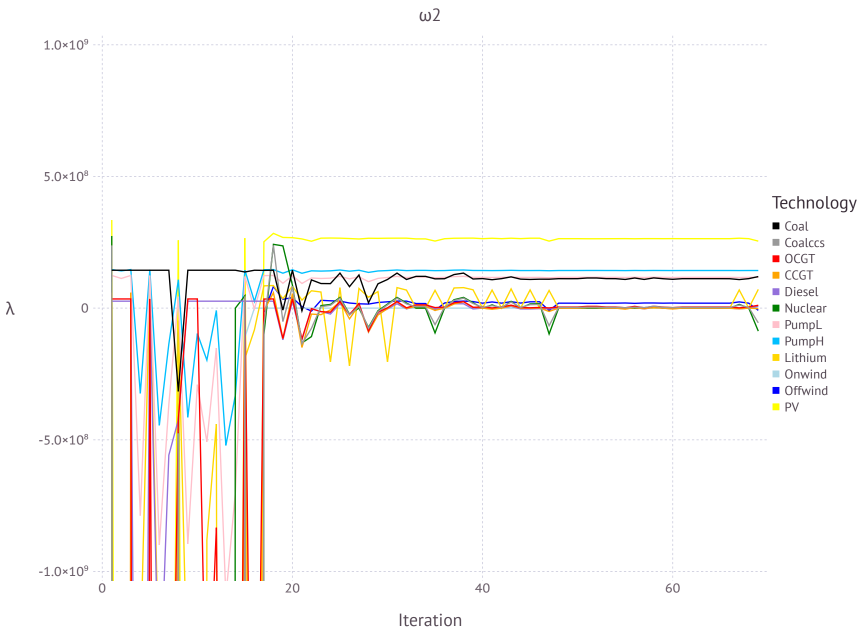
The following three graphs depict the gradients for the first expansion with a deterministic CO_2 emission price of 100. With the exception of nuclear, which is at its capacity limit, the gradients coincide.

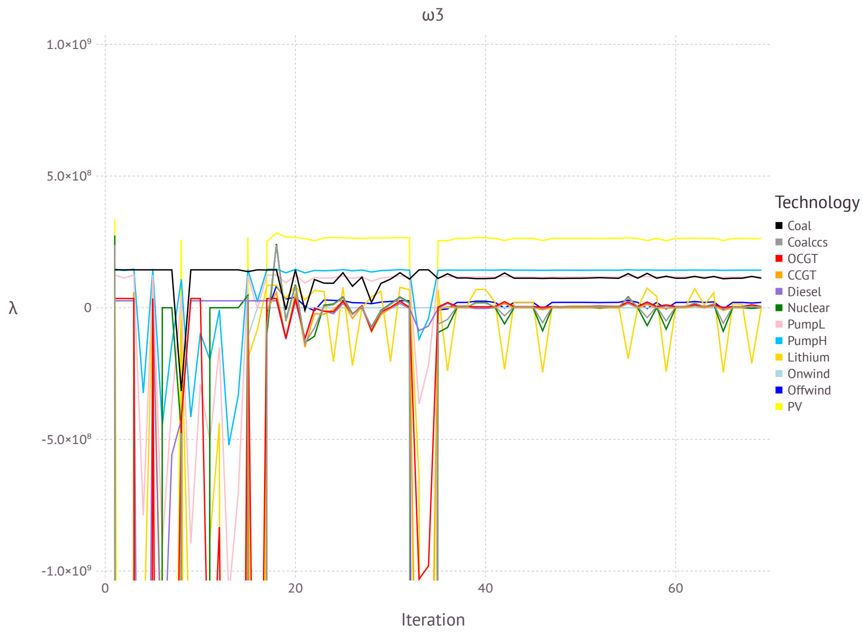




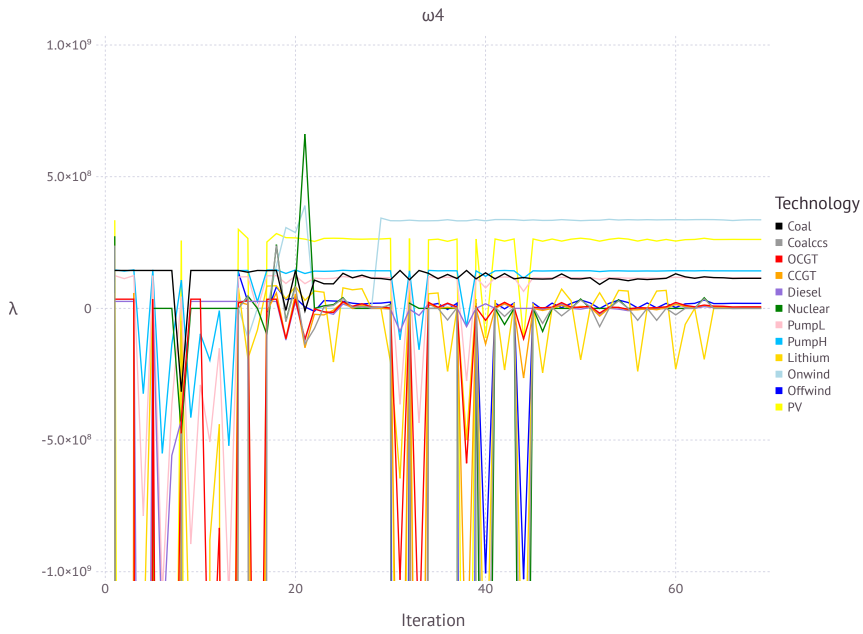


The next three graphs depict the second expansion with a deterministic CO_2 emission price of 60. The gradients do not coincide. Graphs for the remaining expansions, in which the gradients also do not coincide, are not included here.





ω3
1.0×10⁹
5.0×10⁸
0
-5.0×10⁸
-1.0×10⁹
λ
0
20
40
60
Iteration
Technology
Coal
Coalccs
OCGT
CCGT
Diesel
Nuclear
PumpL
PumpH
Lithium
Onwind
Offwind
PV



ω4
1.0×10⁹
5.0×10⁸
0
-5.0×10⁸
-1.0×10⁹
λ
0
20
40
60
Iteration
Technology
Coal
Coalccs
OCGT
CCGT
Diesel
Nuclear
PumpL
PumpH
Lithium
Onwind
Offwind
PV