NTNU – Trondheim
Norwegian University of
Science and Technology

A Benders Decomposition Method for a Multi-stage Stochastic Energy Market Equilibrium Problem

Emily Siggerud

Master of Science in Physics and Mathematics
Submission date: June 2014
Supervisor: Anton Evgrafov, MATH
Co-supervisor: Rudolf Egging, IØT

Norwegian University of Science and Technology
Department of Mathematical Sciences

Preface

This master's thesis was written during the spring 2014, and is my completion of the studies in Applied Mathematics at the Norwegian University of Science and Technology. My sincere gratitudes go to my supervisors, Anton Evgrafov, for providing help and guidance throughout my work, and Ruud Egging, for aiding insight and understanding of the studied application and helping me learning and using the GAMS software. Lastly, I would like to thank the student's computer assistants at the Department of Mathematical Sciences for great IT support.

Trondheim, June 2014
Emily Siggerud

Abstract

In this thesis, a stochastic energy market equilibrium model is developed and implemented in GAMS. The model involves multi objective optimization and is solved as a mixed complementarity problem. To provide an efficient solution strategy, a Benders Decomposition method tailored for the modelled energy market problem is studied and implemented. The scalability of the implemented decomposition algorithm is investigated for different versions of the energy market problem, and the results are compared to the alternative of direct solution in GAMS.

An overall result is that the decomposition method succeeded in finding correct solutions, and proved to be the fastest solver option for the larger instances of the tested problems. In addition, the results did also facilitate a discussion of possibilities for further improvements in the efficiency of the algorithm.

Sammendrag

I denne masteroppgaven har en stokastisk likevektsmodell for energimarkeder blitt utviklet og implementert i GAMS. Modellen omfatter optimering med flere objektivfunksjoner og er løst som et blandet komplementaritetsproblem. En Benders dekomponeringsmetode spesielt tilpasset problemer som den utviklede modellen, er studert og implementert for å oppnå en effektiv løsningsstrategi. For ulike eksempler på energimarkeder er skaleringen av denne metoden undersøkt. Det er også testet at algoritmen gir korrekte resultater i forhold til tilsvarende løsning funnet ved direkte anvendelse av GAMS.

Overordnet er resultatet av dette arbeidet at dekomponeringsmetoden lykkes i å finne korrekte løsninger på problemet. Det viste seg også at denne metoden var det raskeste alternativet til for de største problemene som ble testet. I tillegg har test-resultatene gjort det mulig å påpeke ytterligere muligheter for forbedringer av effektiviteten til algoitmen.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background and Motivation

Energy market models as referred to in this report were first developed and applied in the 1970s. Most models are designed to consider only one energy source (such as natural gas, oil, electricity etc.), and a market located in some region of North America or Europe. Still, their structure and features are often similar regardless of location and type of energy. In this manner, the same models are applicable to a diversity of markets, enabling the modelling of a combination of markets and energy types, and the interaction between them. Combinations of interest are, for example, hydroelectricity and electricity generated by coal plants, as they to a certain extent are mutually exclusive commodities.

An energy market model can aid policy- and decision makers when considering changes in production, consumption and trade with energy. This is an important feature, as stability in the availability of many energy sources is both critical and anticipated in a modern society.

From a modeller's perspective, similarities and other characteristics in many energy markets give the opportunity to exploit the problem structure to develop good solution strategies. It may not be surprising that the determination of international patterns in energy markets is a highly demanding process in terms of data handling and computational resources. Some state of the art models even include stochastic elements to deal with uncertain elements. A collection of possible scenarios are then considered as part of the solution, requiring considerably more work than when seeking a solution to a deterministic model. This amplifies the importance of a fast solution strategy, and is also the motivation for this thesis.

In [16], a Benders Decomposition method tailored for stochastic energy market models is developed and tested. This method is also applied in [9] and [10] with promising results. Inspired by these works, a stochastic energy market model is developed as part of this thesis. The model involves several features that are in common with the aforementioned sources and other models found in the literature, but unlike the others, it does also have a few unique characteristics. These characteristics are: multiple fuel types, transformation facilities form one fuel type to another, and the possibility to invest in more transformation capacity if that should be demanded. To investigate the possibilities to reduce running time, the Benders Decomposition method from [16] is applied to the model.

Another contribution presented in this thesis is the thorough mathematical approach taken. This is in particular when analysing the model features for existence and uniqueness of a solution in general, and when analysing convergence and other technicalities regarding the application of the Benders Decomposition method.

1.2 The Structure of this Thesis

This thesis is structured by first, in Chapter 2, introducing background theory in economics and mathematics that is relevant for the energy marked model described in Chapter 3 and some parts of the later chapters. Next, in Chapter 4, relevant decomposition techniques are described, ending with a description of how the method in [16] applies to the developed model, and a discussion of convergence for the problem. In Chapter 5, the performance of the implemented algorithm is tested and, finally, the thesis is concluded in Chapter 6, summarizing the work and its results and indicating possibilities for further work.

Chapter 2

Background Theory

2.1 Economic Principles

A Market

In a market, producers and consumers are buying and selling products. A common assumption is that each of these actors is operating with only one ambition: to maximize the participant's own welfare. To provide a descriptive model of a market, all interfering or influencing factors must be taken into consideration.

Profit

A supplier's profit is what is left of income from sales when all related costs are subtracted. Income is the product of quantity sold and the selling price.

Supply and Demand

Market demand is the amounts of a product desired by all consumers at a given price. Conversely, supply can be interpreted as the quantity of a product that is offered for sale in a market at a given price.

Equilibrium

In a market, *equilibrium* occurs when supply equals demand. At this point, both suppliers and consumers are satisfied with the quantity and price of the traded product. Such a price is often called 'equilibrium price' or 'market clearing price'. The term 'Market clearing' relates to the fact that no non-supplied demand or excess supply exists.

Nash Equilibrium

Nash Equilibrium is defined in [22] as a state where no participant can obtain a more beneficial situation by changing his strategy, given all other participants strategies. In [22] it is also proven that at least one such equilibrium exists when the number of participants and possible strategies is finite.

Market Power

A market participant that can exert *market power*, is able to affect the market equilibrium significantly by his actions.

Perfectly Competitive Market

In a *perfectly competitive market*, the volume of consumers and producers precludes any exertion of market power.

Monopoly

In a monopoly, only one supplier is present, enabling this supplier - the *monopolist* - to adjust quantities supplied to a level at which the price and quantity maximizes his profit. The monopolists profit is only restricted by the market demand.

Oligopoly

An oligopoly is a market structure with a few actors, facilitating a limited amount of market power exertion.

Consumer Surplus

Consumer surplus (CS) is what consumers save when buying products at a price lower than the maximum price they are willing to pay [26]. Given a price function, $p(q^S)$, describing the price consumers are willing to pay for the product as a function of the quantity available for sale, q^S, the consumer surplus can be computed as:

$$CS = \int_0^{q_{eq}^S} (p - p_{eq}) \, \mathrm{d}q^S, \tag{2.1}$$

where the 'eq'-subscript denotes the variables values at equilibrium.

In a perfectly competitive market, both producers profits and consumer surplus is maximized. This can be modelled as an optimization problem with an objective function describing both profit and CS.

2.2 Optimization Theory

This section covers background theory in optimization that is essential to the work presented later in this report, both the modelling part and the decomposition.

Most of the optimization problems considered are problems with linear equality and inequality constraints:

$$\begin{aligned} \min_{z} \quad & f(z) \\ \text{st.} \quad & Cz \le a \quad (\lambda) \\ & Dz = b \quad (\mu), \end{aligned} \tag{2.2}$$

where $f(z) : \mathbb{R}^n \to \mathbb{R}$, C is an $m \times n$ matrix, D is a $p \times n$ matrix, and $\lambda \in \mathbb{R}^m_+$ and $\mu \in \mathbb{R}^p$ are vectors of Lagrangian multipliers related to the constraints $Cz \le a$ and $Dz = b$ respectively. The domain defined by $Cz \le a$ and $Dz = b$ is called the feasible region to the problem. For (2.2) the feasible region is a *polyhedron* when there is a finite number of constraints, and the constraints are all linear equations.

2.2.1 Optimization under Uncertainty

In this thesis, stochastic optimization problems will be studied. The uncertainty is modelled by considering a collection of possible future scenarios following a *scenario tree*. Hence, some relevant background theory is introduced in this section.

Scenario Trees

The scenario tree is a graph that consists of all possible outcomes together with their likelihood to occur on a discretized time line. The figure below gives a schematic illustration of a scenario tree with 4 time stages and 3 scenarios. At each of the 9 nodes, the unconditional probability, P, for the event(s) represented by the node, and the node number is printed. Unconditional probability means the probability for an event to occur regardless of the occurrence of its ancestor node. Hence, the probabilities at each time stage sum up to 1. The scenarios can be described by listing the associated nodes with respect to ascending time stage, for example; $\{1, 2, 4, 7\}$, $\{1, 3, 5, 8\}$ and $\{1, 3, 6, 9\}$.

Solving Stochastic Optimization Problems

In comparison to optimization problems where all data is deterministic, the uncertainties in a problem with stochastic variables entail a severe increase in complexity [19]. There are different ways to manage the stochastic elements in a model, leading to varieties in complexity and quality. A thorough review of this topic is found in [19]. Here, the most elementary concepts are described.

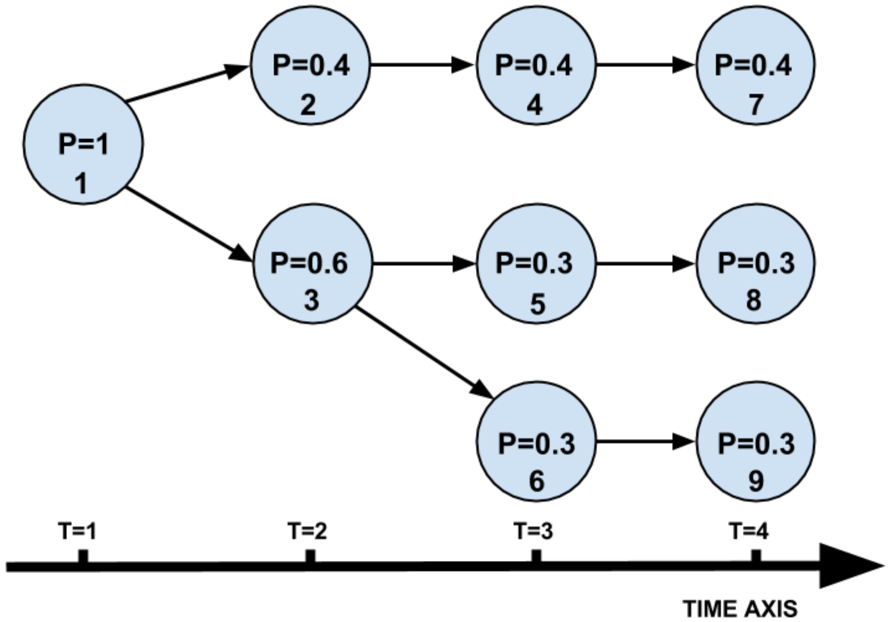


Figure 2.1: Illustration of a scenario tree.

Based on the information of all stochastic elements included in the problem one could compute the expected value of all stochastic variables, and solve the problem as if it was a deterministic problem. The result to this approach is called the *expected value solution* (EVS).

Conversely, the *stochastic solution* (SS) is obtained by considering the expected value of every objective value at every possible variation in the stochastic variables. This means that the problem is solved for all nodes of the scenario tree, and the solution is weighted by each node's probability. In comparison to the EVS, the SS is more demanding to obtain, but it may also be more indicative and true to the real world, because it involves a higher level of details [19].

2.2.2 Optimality Conditions

Consider a problem like (2.2). If $f(z)$ is continuously differentiable at a solution z^*, there is a pair of Lagrangian multiplier vectors, (λ^*, μ^*), such that the following conditions are satisfied [23]:

$$\nabla f(z^*) + C^T \lambda^* + D^T \mu^* = 0 \tag{2.3a}$$
$$a - Cz^* \geq 0 \tag{2.3b}$$
$$b - Dz^* = 0 \tag{2.3c}$$
$$\lambda^* \geq 0 \tag{2.3d}$$
$$\lambda^{*T}(a - Cz^*) = 0, \tag{2.3e}$$

The equations (2.3) are known as the Karush-Kuhn-Tucker conditions (KKT). When all constraints are linear and the number of constraints is finite, the KKT conditions are often referred to as *first order necessary conditions* because they must be satisfied for z^* to be a solution to (2.2). If $f(z)$ in addition is a convex function, the KKT conditions are also *sufficient* [23]. This means that a point z^* satisfying (2.3) is a solution to (2.2). Strict convexity of the objective function also assures that z^* is a unique solution.

2.2.3 Duality

For every optimization problem like (2.2), one can define a Lagrangian *dual* problem [7]:

$$\begin{aligned} \max_{\lambda \geq 0, \mu} \quad & q(\lambda, \mu), \\ \text{st.} \quad & \lambda \geq 0, \end{aligned} \tag{2.4}$$

where the function $q(\lambda, \mu)$ is

$$q(\lambda, \mu) = \inf_z \mathcal{L}(z, \lambda, \mu). \tag{2.5}$$

In this setting, (2.2) is often referred to as the *primal* problem. If $f(z)$ in (2.2) is a linear function, the relation between the primal and dual problem is symmetric [23], that is, deriving the dual of a dual problem always gives the primal. Despite the joint solution, the difficulty in solving either the primal or dual problem may vary [23]. Hence, having an alternative problem at hand is an often appreciated feature in algorithms to solve optimization problems.

When concerning algorithms utilizing the primal-dual relationship, the theorems below are useful.

Theorem 2.2.1. (Weak Duality Theorem) *(Theorem 4.3 in [7]).*

For any z feasible for (2.2) *and any λ and μ feasible for* (2.4)*, the following is always true*

$$q(\lambda, \mu) \leq f(z). \tag{2.6}$$

Theorem 2.2.2. (Strong Duality) *(After Theorem 12.13 in [23]). Suppose that $f(z)$ in (2.2) is convex and continuously differentiable on $\mathbb{R}^n$, and suppose that $\overline{z}$ is a solution of (2.2) at which LICQ*[1] *holds. Suppose that $\hat{\lambda}$ solves (2.4) and that the infimum in $\inf_z \mathcal{L}(z, \hat{\lambda})$ is attained at $\hat{z}$. Assume further that $\mathcal{L}(\cdot, \hat{\lambda})$ is a strictly convex function. Then $\overline{z} = \hat{z}$ (that is, $\hat{z}$ is the unique solution of (2.2)), and $f(\overline{z}) = \mathcal{L}(\hat{z}, \hat{\lambda})$.*

For a linear optimization problem, that is a problem like (2.2) but with $f(z)$ a linear function, it is not hard to show that the KKT conditions derived from the primal and dual problems are identical, see [23]. This is also a consequence of Theorem 2.2.2. Despite the previous focus on problems with a non-linear objective function, the equivalence between the KKT conditions for linear problems will become useful later in this report.

2.2.4 Shadow Prices and Lagrangian Multipliers

When optimization problems such as (2.2) describes prices or costs or a combination thereof, and strong duality holds for the problem, the term *shadow price* is a common synonym to the Lagrangian multipliers or dual variables λ and μ. To interpret shadow price, let $i = 1...m$ and $j = 1...p$, so that $a = [a_1, ..., a_m]^T$ etc. At the optimal solution z^*, the following relation holds:

$$\begin{aligned} f(z^*) = \mathcal{L}(z^*, \lambda^*, \mu^*) &= f(z^*) + \lambda^{*T}(a - Cz^*) + \mu^{*T}(b - Dz^*) \\ &= f(z^*) - \lambda^{*T} C z^* - \mu^{*T} D z^* + \sum_{i=1}^{m} \lambda_i{}^* a_i + \sum_{j=1}^{p} \mu_j{}^* b_j. \end{aligned} \tag{2.7}$$

Differentiation with respect to a_i and b_j gives [7]:

$$\left.\frac{\partial f(z)}{\partial a_i}\right|_{z^*} = \lambda_i, \qquad \left.\frac{\partial f(z)}{\partial b_j}\right|_{z^*} = \mu_j. \tag{2.8}$$

Hence, the shadow price is an approximation of the rate of change for the objective function at z^* respect to the right hand side of a constraint.

[1] LICQ (Linear Independence Constraint Qualification) [21] is satisfied when all gradients of all *active* constraints are linearly independent. The active constraints are all of the equality constraints and the inequality constraints for which $C_i(z_i) = 0$.

2.3 Complementarity Problems

2.3.1 The Non-linear Complementarity Problem - NCP

The solution, z, to a non-linear complementarity problem (NCP) is to find z satisfying the following conditions:

$$0 \leq z \perp F(z) \geq 0, \tag{2.9}$$

where $F(z) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is a given function [15]. The sign $\perp$ is part of common notation for NCPs, and equation (2.9) is equivalent with $z^T \cdot F(z) = 0$. For a linear function $F(z)$, the NCP is called a linear complementarity problem (LCP).

2.3.2 The Mixed Complementarity Problem - MCP

A mixed complementarity problem (MCP) extends the NCP problem class by allowing other bounds on z than 0 only. Hence, $z \in \mathbb{R}^n$, is a solution to a MCP if all of the following conditions are satisfied:

$$\begin{aligned} z_i \in (l_i, u_i) \quad &\Longrightarrow F_i(z_i) = 0 \\ z_i = l_i \quad &\Longrightarrow F_i(z_i) \geq 0 \\ z_i = u_i \quad &\Longrightarrow F_i(z_i) \leq 0, \end{aligned} \tag{2.10}$$

where $i = 1...n$ and l and u provide the lower and upper bounds on z respectively, [15]. With the set $K = [l, u]$, $MCP(K, F)$ is a convenient compact form for the problem (2.10). Optimality conditions can be formulated as an MCP. To do so for the problem (2.2), let:

$$l = \begin{bmatrix} -\infty \\ 0 \\ -\infty \end{bmatrix}, \quad u = \begin{bmatrix} \infty \\ \infty \\ \infty \end{bmatrix} \tag{2.11}$$

$$F(z, \lambda, \mu) = \begin{bmatrix} \nabla f(z) + C^T\lambda + D^T\mu \\ a - Cz \\ b - Dz \end{bmatrix}. \tag{2.12}$$

Then the optimality conditions (2.3) for the optimization problem (2.2) suits a mixed complementarity problem formulation:

$$\begin{aligned} &z \ free, && \nabla f(z) + C^T\lambda + D^T\mu = 0 \\ &0 \leq \ \lambda \perp && a - Cz \geq 0 \\ &\mu \ free, && b - Dz = 0. \end{aligned} \tag{2.13}$$

'z and μ *free*' is convenient notation, widely used in the literature. It refers to the situation when the upper and lower bounds for a variable are ∞ and $-\infty$

respectively. In (2.13), it is the lower bound on λ that distinguishes the problem from being an NCP. Nevertheless, non-negativity for multipliers arises in every optimization problem with inequality constrains, and hence the given example is a rather normal case.

2.4 Variational Inequality Problems - VIs

In [12], finite-dimensional variational inequality problems and complementarity problems are studied. The book gives an excellent overview of the topic, and defines a variational inequality problem as follows. For a function $G(z) : \mathbb{R}^n \to \mathbb{R}^n$ and a set $K \subset \mathbb{R}^n$ the Variational Inequality problem, $VI(K, G)$, is to find $z^* \in K$ such that:

$$G(z^*)^T(z - z^*) \geq 0, \quad \forall z \in K. \tag{2.14}$$

The following relation holds for the solution to (2.14):

$$G(z^*)^T(z) \geq G(z^*)^T(z^*), \quad \forall z \in K. \tag{2.15}$$

Hence, if z^* is treated as fixed, z^* is a solution to (2.14) if and only if $z = z^*$ solves the following (linear) optimization problem.

$$\begin{aligned} \min_z \quad & G(z^*)^T z \\ \text{st.} \quad & z \in K. \end{aligned} \tag{2.16}$$

If, additionally, the set K is polyhedral, the problem (2.16) is a linear optimization problem.

2.4.1 KKT Conditions for VI Problems

For a polyhedral set $K = \{z \in \mathbb{R}^n | Cz \leq a, Dz = b\}$, z^* is a solution of (2.14) if and only if the vectors $\lambda^* \in \mathbb{R}^m$ and $\mu^* \in \mathbb{R}^p$ exist so that the following conditions are satisfied (Proposition 1.2.1 in [12]):

$$\begin{aligned} G(z^*) + C^T\lambda^* + D^T\mu^* = &\ 0 \\ a - Cz^* \geq &\ 0 \\ b - Dz^* = &\ 0 \\ \lambda^* \geq &\ 0 \\ \lambda^{*T}(a - Cz^*) = &\ 0 \end{aligned} \tag{2.17}$$

The conditions (2.17) are the KKT conditions for (2.16). In Section 2.3.2 it was shown that the KKT system for optimization problems belong to a subclass of

MCPs. In the following the corresponding relationship between VI problems and MCPs are investigated. Recall the MCP function $F(z, \lambda, \mu)$ in (2.12). A similar function can be derived for the above KKT system (2.17).

$$\tilde{F}(z, \lambda, \mu) = \begin{bmatrix} G(z) + C^T\lambda + D^T\mu \\ a - Cz \\ b - Dz \end{bmatrix}. \tag{2.18}$$

In other words, $VI(K, G)$ and $MCP(K, \tilde{F})$ are equivalent when $K = \{z \in \mathbb{R}^n | Cz \leq a, Dz = b\}$, and the solution of $VI(K, G)$ is equivalent to the solution of its KKT system. Hence, MCPs are a subclass of VI problems.

Similarly, the KKT conditions for an optimization problem suits a special class of VI problems. Consider the problem (2.2) and note that the feasible region in (2.2) is the polyhedral set $K = \{z \in \mathbb{R}^n | Cz \leq a, Dz = b\}$. If $G(z^*)$ and $f(z^*)$ are such that $G(z^*) = \nabla f(z^*)$, then the conditions (2.17) are the same as the KKT conditions to (2.2).

The converse is not necessarily true; not all VI problems correspond to an optimization problem. Among others, [15] and [12] describe the *Principle of Symmetry* and an *Integrability Condition* which are tools to determine if such an optimization problem exists.

As a summary to this discussion, the relation among the problem classes of importance for the remaining of this report are summarized in Figure 2.2 below.

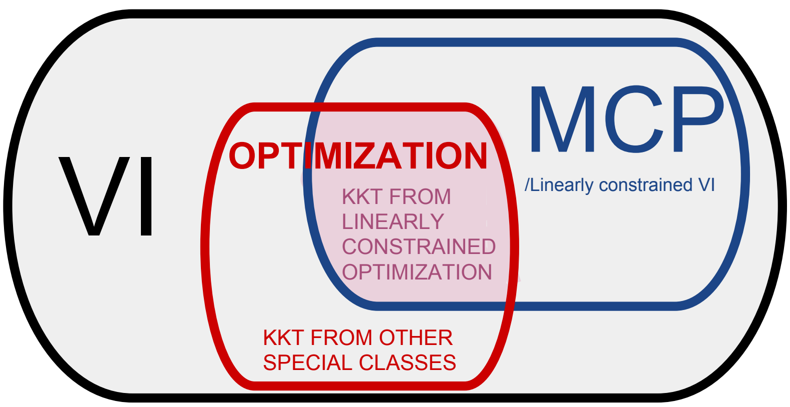


Figure 2.2: Relationship among VI, MCP and KKT conditions of optimization problems.

2.4.2 Duality for VI Problems

Later in this report, when concerning decomposition techniques, the derivation of a dual VI problem will be required. A good piece of deep theory on the topic is found in [18], and [12] describes saddle point theory for a vector valued Lagrangian function corresponding to VI problems. A much easier option that suits the specialized application problem is however found in [16]. In this approach, the linear programming problem (2.16) related to the VI (2.14) is used as a starting point. The dual is deduced by investigating the KKT conditions and using linear duality theory to derive a problem that has dual variables as decision variables and with the same KKT conditions as the primal problem. The details of this approach are given later, in Section 3.4, after the introduction of generalized notation for the energy market model.

2.4.3 Existence and Uniqueness of Solutions to VI Problems

Theory of existence and uniqueness of solutions to VI problems is found in [15] and [12]. While [15] provides a brief overview with focus on applications, [12] takes a more detailed and mathematical approach. The theory in this section is based on both. First a useful theorem on existence:

Theorem 2.4.1. *(Corollary 2.2.5 in [12]) Let $K \subseteq \mathbb{R}^n$ be compact and convex and let $G : K \to \mathbb{R}^n$ be continuous. Then the set of solutions to $VI(K, F)$ is non-empty and compact.*

Before stating another useful theorem on uniqueness of solutions, strict and strong monotonicity is defined:

Definition 2.4.2. Strict Monotonicity *A mapping $G : K \subseteq \mathbb{R}^n \to \mathbb{R}^n$ is strictly monotone on K if*

$$(G(z) - G(y))^T (z - y) > 0 \quad \forall z, y \in K \ \text{ and } z \neq y. \tag{2.19}$$

Definition 2.4.3. Strong Monotonicity *A mapping $G : K \subseteq \mathbb{R}^n \to \mathbb{R}^n$ is strongly monotone on K if there exists a constant $c > 0$ such that*

$$(G(z) - G(y))^T (z - y) \geq c \left\| z - y \right\|^2 \quad \forall z, y \in K. \tag{2.20}$$

The definitions 2.4.2 and 2.4.3 are from [12] together with the following theorem.

Theorem 2.4.4. *(Theorem 2.3.3 in [12]) Let $K \subseteq \mathbb{R}^n$ be a closed and convex set, and let $G : K \to \mathbb{R}^n$ be continuous. If G is strictly monotone on K, $VI(K, G)$ has at most one solution.*

Note that Theorem 2.4.4 does not guarantee that a solution exists, there might as well not be any solution. Uniqueness of a solution is stated in the theorem below.

Theorem 2.4.5. *(Theorem 4.7 in [15]) Let $K \subseteq \mathbb{R}^n$ be a non-empty closed and convex set, and let $G : K \to \mathbb{R}^n$ be continuous. If G is strongly monotone on K, $VI(K, G)$ has a unique solution.*

If the strong monotonicity requirement in Theorem 2.4.5 cannot be satisfied, the next theorem may help, broadening the class of mappings for which a unique solution exists:

Theorem 2.4.6. *(Theorem 9 in [13] Let $K \subseteq \mathbb{R}^n$ be a non-empty closed and convex set, and let $G : K \to \mathbb{R}^n$ be a mapping of the form $G(z) = \binom{F(q)}{\nabla c(x)}$. If F is strictly monotone on K, then the solution to $VI(K, G)$ is unique in q and in the scalar value $c(x)$. If $c(x)$ is a strictly convex function on K, the solution to $VI(K, G)$ is unique in q and x.*

The Theorems 2.4.5 and 2.4.6 are useful in many applications together with the following results; the mapping $G(z) = Az + b : \mathbb{R}^n \to \mathbb{R}^n$ is both strictly and strongly monotone if A is symmetric positive definite. To see why, insert $G(z) = Az + b$ and $G(y) = Ay + b$ into equation (2.19):

$$
\begin{gathered}
(Az + b - Ay - b)^T (z - y) > 0 \\
\Downarrow \\
(Az - Ay)^T (z - y) > 0 \\
\Downarrow \\
(z - y)^T A^T (z - y) > 0
\end{gathered}
\tag{2.21}
$$

If A^T is symmetric positive definite, then $x^T Ax$ is positive for any non-zero vector x. When $z \neq y$, $\forall\ z, y \in \mathbb{R}^n$, the vector $x = z - y$ is non-zero, and thus, $G(z) = Az + b$ is strictly monotone on $\mathbb{R}^n$.

Furthermore, consider equation (2.20), insert $G(z) = Az+b$ and $G(y) = Ay+b$ as before, let the right hand side norm be the ℓ_2-norm, and let I be the identity matrix of same dimensions as A. This gives:

$$
(Az + b - Ay - b)^T (z - y) \geq c \, \|z - y\|^2
$$

$$\Downarrow$$

$$(Az - Ay)^T(z - y) \geq c\,(z - y)^T(z - y)$$

$$\Downarrow$$

$$(z - y)^T A^T (z - y) \geq c\,(z - y)^T I (z - y)$$

$$\Downarrow$$

$$(z - y)^T A^T (z - y) \geq (z - y)^T (cI)(z - y) \qquad (2.22)$$

Now, if A^T is positive definite, the smallest eigenvalue, λ_{min}, of A is positive. If the constant $c > 0$ is $c = \lambda_{min}$, the above equation (2.21) is satisfied $\forall\ z, y \in \mathbb{R}^n$ (equality holds at $z = y$), and $G(z) = Az + b$ is strongly monotone on $\mathbb{R}^n$.

Chapter 3

Energy Market Modelling

In this chapter, the developed energy market model is described. The model has several features in common with other examples in the literature, [9], [10] and [15] are among the main sources of inspiration here. After the model presentation, a generalized problem formulation is stated, and existence and uniqueness is discussed for this problem. The chapter ends with a small example that illustrates the functionality of the model.

3.1 Modelling Approach - Overview of General Features

This section describes a few key concepts of energy market models. Some of the theory given in Chapter 2 is applied here. More intuitive concepts and details are not covered, but introduced together with the actual model formulation.

A general approach is to describe the situation where all market participants offering products (energy) aim to maximize their profits [15]. As participants are interacting and relying on others in a market, the optimization problems are all interdependent. The dependency is specified through common variables and constraints including other supplier's variables. While the objective functions in an energy market model are formulated to maximize profits, the constraints to the corresponding optimization problems describe the physical frame work as well as economical relations between actors in the market.

3.1.1 Networks

Energy is often produced and consumed in different localities. Hence, geographical restrictions must be a part of the model. A common practice is to model regions as nodes. Supply and demand or a combination thereof, can take place in a node. In this way, a single node has the potential to contain a market itself. A global market is spanned by the graph containing all nodes and their connections. Connections between nodes are arcs in the graph and represent transportation possibilities between regions. Such transportation facilities are often operated by a governmental agency or sometimes companies separated from production and trade with energy.

3.1.2 Stochastic Elements

Several elements of an energy market model can be associated with uncertainties. Examples can be factors such as weather, detection of new reservoirs, investments in infrastructure, new technology and economical fluctuations. Politics may also affect the supply and demand. As discussed in Section 2.2.1, discrete variations in such stochastic elements can be stored in a scenario tree. To solve the problem for the energy market model, the stochastic solution (SS) is aimed for, because it is more informative. Hence, the solution is found by weighting all parameters that vary in the scenario tree by the probability of the event(s) in the respective scenario tree nodes to occur.

3.1.3 Participants

The Supply Side

In many energy sectors it is common that the trade includes more types of participants than a set of suppliers and consumers only. Most energy market models found in the up-to-date literature do at least include trading actors as a linking function between producers and consumers. A trader buys and sells energy and is in this fashion a demander of the producer's goods and a supplier of the final consumers' demanded goods. Hence, a trader's objective to maximize his own profit is included in many models. Other examples of participants that can be modelled individually (with its own objective) are transformers and storage operators. A transformer facilitates the transformation to and from different types of energy, for example a coal plant. Storage operators facilitates the storage of energy sources, enabling sale at a later time than production.

Some participants or groups of participants are often assumed to behave similarly, so that their optimization problem formulations can have the same form. In a market with producers, traders and consumers it would for instance be convenient to formulate two standard problems, one for producers and one for traders. The participants' individual problem can later be customized by including participant-specific parameters.

The Demand Side

To find an equilibrium solution, the demand side must also be included in the model. This is usually done by imposing a demand function for all regions where consumers are present. In this way, it is ensured that the energy price and quantities supplied reflects the consumers demand. If the regional demand functions are linear and with slope *slp*, the unit price, u, that consumers are willing to pay at a given quantity, $\sum_{p \in P} q_p^S$, can be computed as:

$$u = int - slp \cdot \sum_{p \in P} q_p^S. \tag{3.1}$$

Where q_p^S is the supplied quantity by producer p, P is the set of producers, and int is the unit price at $\sum_{p \in P} q_p^S = 0$.

3.1.4 Market Clearing

At equilibrium, the market must be cleared. That means, as discussed in Section 2.1, that all supplied quantities are consumed. To a supply chain with more participants than producers and consumers only, the joint interactions among the intermediaries of participants must also be subject to clearing restrictions.

3.1.5 Other Physical Restrictions

Some physical restrictions can be generalized to apply to almost any energy supply chain. One of those restrictions is the level of production, which can be reliant on the resources available, the capacity of the production facilities or both. Another general restriction describes the transition of a suppliers energy from one instance of the supply chain to another. The energy produced must be consistent with the amounts transported and consumed throughout the modelled system.

3.1.6 Competition

Competitiveness of a market is an essential and critical part of any energy market model. Modern markets in the Western world do often have a few monopolistic actors, but the general interpretation seems to be that most markets are somewhere in between oligopolistic and perfectly competitive. Hence, it is of interest to aid the inclusion of all these elements. Sometimes, varieties in competition is included as an element of the model. In [11] such a variation is incorporated by a parameter that assigns weights to each of the distinct competition elements. Competition is also a factor that complicates both the implementation and solution algorithms significantly.

Monopoly

Maximization of the sum of all objective functions in a model can be interpreted as the the situation where all participant's roles are taken by a single player. In other words, this approach will give a single optimization problem with a solution that describes a monopoly.

Perfect Competition

As described in Section 2.1, the market price in a perfectly competitive market can be found when the sum of suppliers profits and consumer surplus is maximized. Hence, a monopoly model can easily be extended to describe a perfectly competitive market by adding a term for consumers surplus in the objective function. By inserting the inverse demand function (3.1) to the definition of consumer surplus given in equation (2.1), the consumer surplus term for this model becomes

$$CS = \frac{1}{2} slp \cdot \sum_{p \in P} (q_p^S)^2. \tag{3.2}$$

In this way, equilibrium states for a perfectly competitive market can also be found by solving a single optimization problem.

Oligopoly

The inclusion of oligopolistic behaviour among participants restricts the solution possibilities of the model substantially. It is no longer possible to solve a single optimization problem to model such a situation. Instead, a collection of dependent optimization problems, one for each supplier, must be solved. A solution that is a Nash equilibrium (see section 2.1) is optimal to all actors given the other actors' solutions. Hence this solution approach is often referred to as *equilibrium modelling.*

The dependency necessitates the optimization problems to be solved simultaneously, yielding a multi-objective problem [15]. To find a solution that is optimal with respect to multiple objectives, the KKT conditions to all optimization problems involved together with the market clearing conditions are are collected, resulting in an MCP. This approach is taken for the energy market model to be introduced next. As described in Chapter 2, a single optimization problem can also be solved as an MCP through the KKT conditions. Therefore, the model formulation is flexible in the sense that other competition elements can be included without causing massive changes.

3.2 A Stochastic Complementarity Model of an Energy Market

3.2.1 Notational Matters

Table 3.1: Sets, variables, functions and parameters.

Sets	
P	Producers
N	Nodes in network
$N(p)$	Nodes operated by producer $p \in P$, $N(p) \subseteq N$
$P(n)$	Producers producing and/or supplying at node $n \in N$, $P(n) \subseteq P$
M	Scenario tree nodes
S	Scenarios
$S(m)$	Scenario in which $m \in M$ is a member, $S(m) \subseteq M$
$S^A(m)$	Scenario tree nodes in $S(m)$ ancestors to $m \in M$
$S^O(m)$	Scenario tree nodes in $S(m)$ offspring of $m \in M$
D	Demand sectors
E	Types of energy
Variables	
$q^p_{m,n,p,e}$	Quantity produced
$q^S_{m,n,p,d,e}$	Quantity sold
$q^T_{m,n,n',p,e}$	Quantity transported
$q^C_{m,n,p,e,e'}$	Quantity transformed
$f_{m,n,n',e}$	Flow by transporter
$f^I_{m,n,n',e}$	Investment in transportation capacity
$x_{m,n,e,e'}$	Quantity transformed by transformer
$x^I_{m,n,e}$	Investment in transformation output capacity
Dual Variables	

Continued on next page...

Table 3.1 – Continued

$\epsilon_{m,n,p,e}$	Lagrange multiplier to production capacity constraint (3.5)
$\zeta_{m,n,p,e}$	Lagrange multiplier to mass balance constraint (3.6)
$\iota_{m,n,n',e}$	Lagrange multiplier to flow capacity constraint (3.11)
$\kappa_{m,n,n',e}$	Lagrange multiplier to flow capacity investment constraint (3.12)
$\nu_{m,n,e}$	Lagrange multiplier transformation capacity constraint (3.14)
$\xi_{m,n,e}$	Lagrange multiplier to transformation capacity investment constraint (3.15)
$\upsilon_{m,n,n',e}$	Lagrange multiplier to transportation market clearing equation (3.16)
$\phi_{m,n,e,e'}$	Lagrange multiplier to transformation market clearing equation (3.17)
Functions	
$u_{m,n,d,e}\Big(\sum_{p\in P(n)} q^S_{m,n,p,d,e}\Big)$	Unit price of energy
$c^P_{m,n,p,e}(q^P_{m,n,p,e})$	Production cost
z_p	Producer p's profit
z_t	Transporter's profit
z_x	Transformer's profit
Parameters	
$\overline{q^P}_{m,n,p,e}$	Max production capacity
$int_{m,n,d,e}$	Demand at $u_{m,n,d,e} = 0$
$slp_{m,n,d,e}$	Slope of inverse demand curve
$\overline{f}_{m,n,n',e}$	Max flow capacity before extensions
$\overline{f^I}_{m,n,n',e}$	Max flow capacity investment at a scenario tree node, m
$\overline{x}_{m,n,e}$	Max transformation capacity before extensions
$\overline{x^I}_{m,n,e}$	Max transformation capacity investment at a scenario tree node, m
$disc_m$	Discount factor
$prob_m$	Probability of scenario tree node m

Continued on next page…

Table 3.1 – Continued

k_1	Positive coefficient in quadratic term of production cost function
k_2	Positive coefficient in linear term of production cost function
k_3	Positive coefficient in quadratic term of flow cost
k_4	Positive coefficient in linear term of flow capacity expansion cost function
k_5	Positive coefficient in quadratic term of transformation cost
k_6	Positive coefficient in linear term of transformation capacity expansion cost function
$l_{m,n,e,e'}$	Efficiency rate for transformation

3.2.2 The Market Participants' Optimization Problems

In the following, the optimization problems with the objective to maximize profits are described for all types of market participants. Afterwards the market clearing conditions are stated. The KKT conditions which gives the energy market equilibrium problem in the form of an MCP are given in Appendix A.

Producers

Producers, $p \in P$, of energy aim to maximize expected profits, z_p, described as the following optimization problem:

$$\max_{\substack{q^P,q^S \\ q^C,q^T}} z_p = \sum_{m \in M} \sum_{d \in D} \sum_{e \in E} \sum_{n \in N(p)} prob_m \cdot disc_m \cdot \Bigg(u_{m,n,d,e}\Big(\sum_{p' \in P(n)} q^S_{m,n,p',d,e} \Big) \cdot q^S_{m,n,p,d,e} - c^P_{m,n,p,e}(q^P_{m,n,p,e}) - \Big(\sum_{n' \in N(p)} \upsilon_{m,n,n',e} \cdot q^T_{m,n,n',p,d,e} \Big) - \Big(\sum_{e' \in E} \phi_{m,n,e,e'} \cdot q^C_{m,n,p,e,e'} \Big) \Bigg) \tag{3.3}$$

$$
\begin{aligned}
\text{st.} \quad q^S_{m,n,p,d,e},\ q^P_{m,n,p,e},\ q^T_{m,n,n',p,e},\ q^C_{m,n,p,e,e'} &\geq 0 && (3.4)\\
q^P_{m,n,p,e} &\leq \overline{q^P}_{m,n,p,e} \quad \forall\, m,n,p,e (\epsilon_{m,n,p,e}) && (3.5)\\
q^P_{m,n,p,e} + \sum_{n' \in N(p)} q^T_{m,n',n,p,e} &+ \sum_{e' \in E} l_{m,n,e',e} \cdot q^C_{m,n,p,e',e} - \sum_{d \in D} q^S_{m,n,p,d,e} \\
- \sum_{n' \in N(p)} q^T_{m,n,n',p,e} - \sum_{e' \in E} q^C_{m,n,p,e,e'} &= 0 \quad \forall\, m,n,p,e \qquad (\zeta_{m,n,p,e}) && (3.6)
\end{aligned}
$$

Greek letters denote the Lagrange multipliers associated with a constraint. The non-negativity conditions are not assigned any multiplier as such variables will not be used explicitly in this work. The expected profit is calculated by summing the profits over all scenario tree nodes, $m \in M$, weighted by the probability $prob_m$ of each scenario. Profits of a producer are the incomes from sales with costs of production, transportation and transformation subtracted. By summing over all nodes of the market network ($n \in N$), all types of energy ($e \in E$) and sectors with different demands ($d \in D$), the total profit, z_p, of a producer's operations is found. Income from sales of energy is each producers supplied quantities, $q^S_{m,n,p,d,e}$ times the unit price $u_{m,n,d,e}$. To find the unit price, an inverse demand function is involved, based on the theory in Section 2.1:

$$
u_{m,n,d,e}\Big(\sum_{p \in P(n)} q^S_{m,n,p,d,e}\Big) = int_{m,n,d,e} - slp_{m,n,d,e} \cdot \Big(\sum_{p \in P(n)} q^S_{m,n,p,d,e}\Big). \qquad (3.7)
$$

Production costs are described by the quadratic function:

$$
c^P_{m,n,p,e}\big(q^P_{m,n,p,e}\big) = k_1 \cdot \big(q^P_{m,n,p,e}\big)^2 + k_2 \cdot \big(q^P_{m,n,p,e}\big). \qquad (3.8)
$$

Costs associated with transportation from node $n \in N$ is the sum of the cost of transportation to any other node operated by producer p (in $n' \in N(p)$). Hence, the summation over all $n' \in N$ is included in (3.3). The cost of transportation on the arc (n, n') is computed by taking the unit price of transportation, $\upsilon_{m,n,n',e}$, times the quantity transported, $q^T_{m,n,n',p,e}$.

Similarly, the costs of transformation from type e to any other type, $e' \in E$, are included in the producers objective function as the unit cost, $phi_{m,n,e,e'}$ times the quantity transformed, $q^C_{m,n,p,e,e'}$.

The constraints collected in (3.4) ensure that all decision variables are positive. In addition, the produced quantities must not exceed the production capacity available. Equation (3.5) enforces this. Lastly constraint (3.6), states that the produced, transported and supplied quantities must be in compliance so that mass balance is preserved for every producer and every type of energy at every node at all times.

Transporters

A transporter provides transport to producers at a unit price. In addition, its operations include extensions of the transportation capacity if that is demanded. The transporters objective function is:

$$\max_{f,f^I} \; z_t = \sum_{m \in M} \sum_{e \in E} \sum_{n \in N} \sum_{n' \in N} prob_m \cdot disc_m \cdot \Big(\upsilon_{m,n,n',e} \cdot f_{m,n,n',e} - k_3 \cdot (f_{m,n,n',e})^2 - k_4 \cdot f^I_{m,n,n',e} \Big) \tag{3.9}$$

$$\begin{aligned} \text{st.} \quad f_{m,n,n',e},\ f^I_{m,n,n',e} &\geq 0 && (3.10) \\ f_{m,n,n',e} &\leq \overline{f}_{m,n,n',e} + \sum_{m' \in S^A(m)} f^I_{m',n,n',e} \quad \forall m,n,n',e \;\; (\iota_{m,n,n',e}) && (3.11) \\ f^I_{m,n,n',e} &\leq \overline{f^I}_{m,n,n',e} \quad \forall m,n,n',e \;\; (\kappa_{m,n,n',e}) && (3.12) \end{aligned}$$

Again, the expected profit is maximized. The transporters profit, z_t, is the income from providing transportation with cost of transportation and expanding the transportation capacities subtracted. Income is the unit price of transporting energy, $\upsilon_{m,n,n',e}$, times the total flow, $f_{m,n,n',e}$, on each arc (n, n'). The cost of transporting a unit is $k_3 \cdot (f_{m,n,n',e})^2$, and the costs of extensions $k_4 \cdot f^I_{m,n,n',e}$. As for producers, the constraint (3.10) ensures that the decision variables in (3.9) are positive. Expansions in the flow capacity are prevented form exceeding some upper limitation by constraint (3.12). The flows are in a similar fashion constrained by the initial maximal capacity, $\overline{f}_{m,n,n',e}$, together with earlier investments in expansions, $\sum_{m' \in S^A(m)} f^I_{m',n,n',e}$. Hence, the cost of increasing the transportation capacity applies to the time stage before the expansion at first can be utilized. The expansion is available at any later time stage.

Transformer

The transformer facilitates the transformation from one type of energy, e, to another, e' $(e, e' \in E)$. For the transformation of a quantity $x_{m,n,e,e'}$, an energy specific unit price, $\phi_{m,n,e,e'}$, is charged and a unit cost k_5 applies. The transformation also causes a certain waste specified by the efficiency rate parameter, $l_{m,n,e,e'}$. As for the transporter, total transformations must not give outputs exceeding the capacity limit, $\overline{x}_{m,n,e'}$ and it can be invested in future expansions in transformation output capacity, $x^I_{m,n,e'}$. Thus the transformer's optimization problem reads:

$$\max_{x,x^I} z_x = \sum_{m\in M}\sum_{e\in E}\sum_{n\in N} prob_m \cdot disc_m \cdot \Bigg(\Big(\sum_{e'\in E}\big(\phi_{m,n,e',e}\cdot x_{m,n,e',e} - k_5\cdot(x_{m,n,e',e})^2\big)\Big) - k_6\cdot x^I_{m,n,e}\Bigg)$$

$$\text{st.} \quad x_{m,n,e',e},\ x^I_{m,n,e} \geq 0 \tag{3.13}$$

$$\sum_{e\in E} l_{m,n,e',e}\cdot x_{m,n,e',e} \leq \overline{x}_{m,n,e} + \sum_{m'\in S^A(m)} x^I_{m',n,e} \quad \forall m,n,e\ (\nu_{m,n,e}) \tag{3.14}$$

$$x^I_{m,n,e} \leq \overline{x^I}_{m,n,e} \qquad \forall m,n,e\ (\xi_{m,n,e}) \tag{3.15}$$

Market Clearing

The producers and transporters are linked through the equation below. This ensures that the flow handled by the transporter equals the sum of all suppliers transported quantities on all arcs.

$$f_{m,n,n',e} = \sum_{p\in P(n)} q^T_{m,n,n',p,e} \qquad (\upsilon_{m,n,n',e}). \tag{3.16}$$

The equilibrium unit equilibrium price of transportation, $\upsilon_{m,n,n',e}$, is the Lagrange multiplier for constraint (3.16). In the same way, producers are linked to the transformer as follows:

$$x_{m,n,e,e'} = \sum_{p\in P(n)} q^C_{m,n,p,e,e'} \qquad (\phi_{m,n,e,e'}), \tag{3.17}$$

where $\phi_{m,n,e,e'}$ is the Lagrange multiplier for constraint (3.17) and gives the unit price of energy transformation.

3.3 Important Properties

For later, it is noted that all constraints together with the market clearing equations are linear equalities and inequalities. Hence the feasible region for each problem is convex. Furthermore, the three objective functions are quadratic and concave, and thus, each of the three optimization problems are convex.

It is easy to see that the feasible region to each separate problem is non-empty, because all the included variables being equal to zero is a feasible point. The same applies to the problem arising when the market clearing conditions are imposed so that the three problems are interrelated.

3.4 Generalized Notation

Due to the high level of details in the above notation, in the corresponding KKT conditions given in Appendix A and in the resulting MCP, it is for the purpose of further work, desirable to extract a generalized formulation of both the KKT conditions forming the MCP and the equivalent VI problem. The generalized notation is addressed in this section, leading to a form that is comparable with the theory and later algorithms described in [16].

First, the primal variables are considered. These are grouped in two categories. Investments in transformation and transportation capacities, $x^I_{m,n,e}$ and $f^I_{m,n,n',e}$, are the only variables that (through the constraints (3.11) and (3.14)) connects the solutions for more than one scenario tree node. Thus $x^I_{m,n,e}$ and $f^I_{m,n,n',e}$ are collected in the set y. All other variables are contained in the other set, which again is divided in two components; (q, x). q is distinguished from x by causing non-constant terms in the gradient of one of the objective functions (3.3), (3.9) and (3.13). Hence $q^P_{m,n,p,e}$, $q^S_{m,n,p,d,e}$, $f_{m,n,n',e}$ and $x_{m,n,e,e'}$ are in q while $q^T_{m,n,n',p,e}$ and $q^C_{m,n,p,e,e'}$ are in x. y, x and q are all non-negative variables because their elements are so.

To express the MCP and corresponding VI problem, let each of the three objective functions be $z_i(q, x, y)$. Then it is convenient to formulate the gradient of each of the three objective functions as:

$$\nabla z_i = F_i(q)^T + d_i^T + l_i^T \qquad \forall i \in \{1, 2, 3\}. \tag{3.18}$$

As already mentioned, the variables in q are identified as the only variables with a non-constant gradient. Thus, the function $F(q)$ describes the gradient with respect to q, while the other variables have constant gradients, d and l. Note that none of the elements in q, x and y appears in more than one of the three problems explicitly. That is, $q^S_{m,n,p,d,e}$ is only part of the producers problem etc.

The constraints and market clearing equations are associated with three different categories; inequalities involving the variable y only, the remaining inequalities, and lastly all equalities (mass balance and market clearing equations). The first group comprises the constraints (3.12) and (3.15), and has the general form:

$$\widetilde{D}y \geq \widetilde{c} \quad (v_1), \tag{3.19}$$

where

$$\widetilde{D} = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}, \quad y = \begin{bmatrix} f^I{}_{m,n,n',e} \\ x^I{}_{m,n,e} \end{bmatrix}, \quad \widetilde{c} = \begin{bmatrix} -\overline{f^I}_{m,n,n',e} \\ -\overline{x^I}_{m,n,e} \end{bmatrix}, \text{ and } v_1 = \begin{bmatrix} \kappa_{m,n,n',e} \\ \xi_{m,n,e} \end{bmatrix},$$

and the variable v_1 is the associated vector of Lagrange multipliers. This explicit relation between the original and generalized equations are left out in the description of the two remaining categories. Group two summarizes the constraints (3.5), (3.11) and (3.14):

$$\overline{A}q + \overline{D}y \geq \overline{c} \quad (v_2). \tag{3.20}$$

The last group, contains the equality constrains (3.6), (3.16) and (3.17). None of these constraints involves constant terms, and hence, the generalized form is:

$$\hat{A}q + \hat{B}x = 0 \quad (v_3). \tag{3.21}$$

Altogether, the constraints to the three optimization problems are now reduced to:

$$\begin{aligned} &\widetilde{D}y \geq \widetilde{c} && (v_1) \\ &\overline{A}q + \overline{D}y \geq \overline{c} && (v_2) \\ &\hat{A}q + \hat{B}x = 0 && (v_3) \\ &q, x, y \geq 0 \end{aligned} \tag{3.22}$$

By letting:

$$F(q) = \begin{bmatrix} F_1(q) \\ F_2(q) \\ F_3(q) \end{bmatrix}, \quad d = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix} \quad \text{and} \quad l = \begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix}.$$

The KKT conditions to (3.22) are:

$$0 \leq q \perp F(q) - \overline{A}^T v_2 - \hat{A}^T v_3 \geq 0 \tag{3.23a}$$

$$0 \leq x \perp d - \hat{B}^T v_3 \geq 0 \tag{3.23b}$$

$$0 \leq y \perp l - \widetilde{D}^T v_1 - \overline{D}^T v_2 \geq 0 \tag{3.23c}$$

$$0 \leq v_1 \perp \widetilde{D}y - \widetilde{c} \geq 0 \tag{3.23d}$$

$$0 \leq v_2 \perp \overline{A}q + \overline{D}y - \overline{c} \geq 0 \tag{3.23e}$$

$$v_3 \ free, \quad \hat{A}q + \hat{B}x = 0 \tag{3.23f}$$

The equations (3.23), yields an MCP, and has the following corresponding VI problem:

For $K = \{(q, x, y) \mid \ \widetilde{D}y \geq \widetilde{c},\ \overline{A}q + \overline{D}y \geq \overline{c},\ \hat{A}q + \hat{B}x = 0, q \geq 0,\ x \geq 0,\ y \geq 0\}$,
find $(q^*, x^*, y^*) \in K$ s.t.
$F^T(q^*)(q - q^*) + d^T(x - x^*) + l^T(y - y^*) \geq 0, \ \ \forall (q, x, y) \in K.$ (3.24)

To obtain an even more compact form, let the mapping G be

$$G\begin{pmatrix} q \\ x \\ y \end{pmatrix} = \begin{pmatrix} F(q) \\ d \\ l \end{pmatrix}, \tag{3.25}$$

then the problem in (3.24) is $VI(K, G)$.

Table 3.2: Overview of Generalized Notation

Generalized Primal Vectors	**Contained Variables**
$y =$	$[f^I_{m,n,n',e}, x^I_{m,n,e}]^T$
$q =$	$[q^P_{m,n,p,e}, q^S_{m,n,p,d,e}, f_{m,n,n',e}, x_{m,n,e,e'}]^T$
$x =$	$[q^T_{m,n,n',p,e}, q^C_{m,n,p,e,e'}]^T$
Generalized Objective Function Elements	**Contained Elements**
$F_1(q) =$	$prob_m \cdot disc_m \cdot \begin{pmatrix} k_1 \cdot q^P_{m,n,p,e} + k_2 \\ -\left(int_{m,n,d,e} - slp_{m,n,d,e} \cdot \left(q^S_{m,n,p,d,e} + \sum_{p' \in P(n)} q^S_{m,n,p',d,e}\right)\right) \\ 0 \\ 0 \end{pmatrix}$
$F_2(q) =$	$prob_m \cdot disc_m \cdot \begin{pmatrix} 0 \\ 0 \\ 2 \cdot k_3 \cdot f_{m,n,n',e} - \upsilon_{m,n,n',e} \\ 0 \end{pmatrix}$
$F_3(q) =$	$prob_m \cdot disc_m \cdot \begin{pmatrix} 0 \\ 0 \\ 0 \\ 2 \cdot k_5 \cdot x_{m,n,e',e} - \phi_{m,n,e,e'} \end{pmatrix}$
$d_1 =$	$prob_m \cdot disc_m \cdot \begin{pmatrix} \upsilon_{m,n,n'e} \\ \phi_{m,n,e,e'} \end{pmatrix}$
$d_2 =$	$prob_m \cdot disc_m \cdot \begin{pmatrix} 0 \\ 0 \end{pmatrix}$
$d_3 =$	$prob_m \cdot disc_m \cdot \begin{pmatrix} 0 \\ 0 \end{pmatrix}$
$l_1 =$	$prob_m \cdot disc_m \cdot \begin{pmatrix} 0 \\ 0 \end{pmatrix}$
$l_2 =$	$prob_m \cdot disc_m \cdot \begin{pmatrix} k_4 \\ 0 \end{pmatrix}$
$l_3 =$	$prob_m \cdot disc_m \cdot \begin{pmatrix} 0 \\ k_6 \end{pmatrix}$

Continued on next page...

Table 3.2 – Continued

Generalized Dual Vectors	**Contained Dual Elements**
$v_1 =$	$[\kappa_{m,n,n',e}, \xi_{m,n,e}]^T$
$v_2 =$	$[\epsilon_{m,n,p,e}, \iota_{m,n,n',e}, \nu_{m,n,e}]^T$
$v_3 =$	$[\zeta_{m,n,p,e}, \upsilon_{m,n,n',e}, \phi_{m,n,e,e'}]^T$

3.4.1 The generalized Dual VI Problem

As mentioned in Section 2.4.2, [16] is followed to obtain a dual form of the problem (3.24). The first step in this approach is to express the original VI problem as an LP problem where the solution to q^* is assumed to be known. Following the theory in Section 2.4, gives:

$$\begin{aligned} \min_{q,x,y} \quad & F(q^*)^T q + d^T x + l^T y \\ \text{st.} \quad & \widetilde{D} y \geq \widetilde{c} \quad (v_1) \\ & \overline{A} q + \overline{D} y \geq \overline{c} \quad (v_2) \\ & \hat{A} q + \hat{B} x = 0 \quad (v_3) \\ & q, x, y \geq 0 \end{aligned} \tag{3.26}$$

Next, linear duality theory (see [23]) can be applied to derive the dual VI problem. As in (3.26), the solution to variables that occur in non-linear terms must be fixed in the LP version of a VI problem. Hence, a new dual variable, f, is introduced, together with the properties $F(q) = f$ and $F^{-1}(f^*) = q$. Details regarding the invertibility of $F(q)$ are shown in Section 3.5. When f^* is fixed in $F^{-1}(f)$, the dual problem becomes:

$$\begin{aligned} \min_{v_1,v_2,v_3,f} \quad & -\widetilde{c}^T v_1 - \overline{c}^T v_2 + F^{-1}(f^*)^T f \\ \text{st.} \quad & \widetilde{D}^T v_1 + \overline{D}^T v_2 \leq l \quad (y) \\ & \overline{A}^T v_2 + \hat{A}^T v_3 \leq f \quad (q) \\ & \hat{B}^T v_3 \leq d \quad (x) \\ & v_1, v_2 \geq 0 \end{aligned} \tag{3.27}$$

which is the linear optimization problem corresponding to the following VI problem:

$$\begin{aligned} &\text{Find } (v_1^*, v_2^*, v_3^*, f^*) \in K_D \text{ s.t.} \\ &-\widetilde{c}^T(v_1 - v_1^*) - \overline{c}^T(v_2 - v_2^*) + F^{-1}(f^*)^T(f - f^*) \geq 0, \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad \forall (v_1, v_2, v_3, f) \in K_D. \end{aligned} \tag{3.28}$$

with

$$K_D = \left\{ \begin{array}{c} (v_1, v_2, v_3, f) \mid \\ \widetilde{D}^T v_1 + \overline{D}^T v_2 \leq l \quad (y) \\ \overline{A}^T v_2 + \hat{A}^T v_3 \leq f \quad (q) \\ \hat{B}^T v_3 \leq d \quad (x) \\ v_1 \geq 0, v_2 \geq 0 \end{array} \right\}. \tag{3.29}$$

Similarly to the notation in (3.27), the dual variables to the constraints in K_D, y, q and x, are entered in brackets just after the respective constraint.

The equivalence between the primal and dual form is defined in the following theorem.

Theorem 3.4.1. *(Theorem 1 in [14]). The vectors* (q, x, y) *and* (v_1, v_2, v_3) *solve the primal VI problem* (3.24) *if and only if* (f, v_1, v_2, v_3) *with* $f = F(q)$ *together with the constraints multiplier vector* (q, x, y)*, solves the dual VI problem* (3.28).

3.5 Existence and Uniqueness

Fundamental theory of existence and uniqueness of an equilibrium solution to the energy market model was introduced in Section 2.4.3. By use of Theorem 2.4.1, it can be shown that a solution exists. The theorem applies to the VI problem $VI(K, G)$, and requires the feasible region K to be a compact and convex subset of $\mathbb{R}^n$. $K \subseteq \mathbb{R}^n$ is true because all involved variables are real-valued. The set K is also convex and closed because it is defined by linear equality and inequality constraints. K is thereby a convex polyhedron. In general, a set in $\mathbb{R}^n$ is compact if it is bounded and closed, and thus it remains to investigate the boundedness.

All variables are ≥ 0, and hence bounded from below. Upper bounds on the variables $q^P_{m,n,p,e}$, $f_{m,n,n',e}$, $f^I_{m,n,n'e}$, $x_{m,n,e,e'}$ and $x^I_{m,n,e}$ are defined in the equations (3.5), (3.11), (3.12), (3.14) and (3.15) respectively. This means that $q^T_{m,n,n',p,e}$ and $q^C_{m,n,p,e',e}$ also have an upper bound as these are related to $f_{m,n,n',e}$ and $x_{m,n,e,e'}$ through the market clearing equations (3.16) and (3.17) respectively. Lastly, the same logic applies to $q^S_{m,n,p,d,e}$, which is related to the other producer variables through the mass balance constraint (3.6). Therefore, it can be concluded that the set K is bounded. The last requirement from Theorem 2.4.1 is that the mapping

F is continuous. This can be observed in (3.24), where $F(q)$ is a linear function, and l and d are coefficients. Thus, there exists at least one solution, (q^*, x^*, y^*), to the VI problem (3.24) and therefore also to the MCP (3.23).

In many similar applications, the Theorems 2.4.4, 2.4.5 and 2.4.6 are applied to verify the existence of a unique solution, requiring strict and strong monotonicity of the mapping, or parts of the mapping for the latter theorem. Theorem 2.4.6 is a natural choice for the present application. Recall that this theorem involves a mapping of the form $G(z) = \binom{F(q)}{\nabla c(x')}$ (x' is used here to avoid conflicts with the generalized notation to the present application). From equation (3.25), it can be seen that this form is obtained if $\nabla c(x') = \binom{d}{l}$. With $c(x') = \binom{d^T x}{l^T y}$, the property $\nabla c(x') = \binom{d}{l}$ is satisfied, and therefore $c(x')$ is convex and the mapping $G(z)$ can be used to describe the problem specific VI in (3.24). Furthermore, since $F(q)$ is a linear mapping of the form $F(q) = Cq + b$, it is relevant to investigate if C is symmetric positive definite. If that is the case, uniqueness of the solution follows from Theorem 2.4.6.

In the three objective functions for the model, the variables that appear in quadratic terms are members of q. Thus, matrix C consists of the coefficients to all quadratic terms in the objective functions. Except from $q^S_{m,n,p,d,e}$, all variables in q are only multiplied by themselves and a coefficient in the quadratic terms, and therefore giving non-zero entries to the diagonal of C only. The coefficients are all positive; $prob_m \cdot disc_m \cdot 2 \cdot k_1$, $prob_m \cdot disc_m \cdot 2 \cdot k_3$ and $prob_m \cdot disc_m \cdot 2 \cdot k_5$. However, the unit price function (3.7) for sold quantities, does lead to off-diagonal terms in C. The summation over $p' \in P(n)$ causes interaction terms among $q^S_{m,n,p,d,e}$ and every $q^S_{m,n,p',d,e}$ for $p' \in P(n)$, leaving the coefficient $prob_m \cdot disc_m \cdot slp_{m,n,d,e}$ in positions that are not on the diagonal of C. Nevertheless, all the diagonal terms corresponding to $q^S_{m,n,p,d,e}$-variables are of the form $prob_m \cdot disc_m \cdot 2 \cdot slp_{m,n,d,e}$, leading to a sub matrix in C that has positive eigenvalues only. As all other entries in C are positive and located on the diagonal, it can be concluded that the matrix is symmetric positive definite, and thus, by Theorem 2.4.6 and the result in equation (2.21), the problem has a solution that is unique in q, x and y.

Another important consequence of the fact that C is symmetric positive definite, is that $F(q)$ is invertible. This is used in the derivation of the dual VI problem as shown in Section 3.4.1.

At this point it is worth mentioning, that if probabilities or discount rates or possibly both are small, so that $prob_m \cdot disc_m \to 0$, then C may become singular. To use data that has scenarios with 0 probability, may seem to be a neat short-cut for a modeller testing various problems. But due to the above reasoning, such cases should in stead be handled by constructing a new scenario tree where the unattainable nodes are excluded. If the discount rate reaches zero, this is because the total period of time stages considered is large relatively to the discount rate.

In such cases, special attention should be paid to the respective factors in the mapping $F(q)$.

3.6 Implementation and Software

MCPs can be solved GAMS[1] [24], using the PATH solver [3]. The solver is based on a generalization of the Newton method to MCPs. For the present application, the GAMS environment is well suited, facilitating a high-level language closely related to the problem formulation. Thus, in terms of implementation of the present application, the MCP formulation of the KKT conditions derived form optimization problems and market clearing equations is a good starting point. As mentioned earlier, the non-generalized KKT conditions are listed in Appendix A.

The GAMS code used to solve the entire model as a single MCP is available on GitHub [25].

3.7 Test Experiments

To illustrate some of the functions in the energy market model, a small test example is shown here.

3.7.1 Example Problem Description

The following tables (Table 3.3 and 3.4) shows the deterministic input data used.

Table 3.3: Input data, sets and parameters

Sets	
P	$\{1,2\}$
N	$\{1,2\}$
$N(1)$	$\{1,2\}$
$N(2)$	$\{2\}$
$P(1)$	$\{1\}$
$P(2)$	$\{2\}$
M	$\{1,2,3,4\}$
S	$\{1,2,3\}$
$S(1)$	$\{1,2,3\}$

Continued on next page...

[1]platform: win64, version: 24.0.2

Table 3.3 – Continued

$S(2)$	$\{2\}$
$S(3)$	$\{3\}$
$S(4)$	$\{4\}$
$S^A(2)$	$\{1\}$
$S^A(3)$	$\{1\}$
$S^A(4)$	$\{1\}$
$S^O(1)$	$\{2,3,4\}$
D	$\{1,2\}$
E	$\{1,2\}$
Parameters	
$\overline{q^P}_{m,n,p,e}$	$30\ \forall\, m \in M, n \in N, p \in P, e \in E$
$int_{m,n,d,e}$	See Table 3.4 below
$slp_{m,n,d,e}$	$1\ \forall\, m \in M, n \in N, p \in P, e \in E$
$\overline{f}_{m,1,2,e}$	$0.1\ \forall\, m \in M, e \in E$
$\overline{f^I}_{m,1,2,e}$	$5\ \forall\, m \in M, e \in E$
$\overline{x}_{m,2,e}$	$0.1\ \forall\, m \in M, e \in E$
$\overline{x^I}_{m,2,e}$	$10\ \forall\, m \in M, e \in E$
$disc_1$	1
$disc_m$	0.98 for $m = 2,3,4$
$prob_1$	1
$prob_m$	0.3 for $m = 2,4$
$prob_3$	0.4
k_1	1
k_2	1
k_3	1
k_4	1
k_5	1
k_6	1
$l_{m,2,1,2}$	$0.4\ \forall\, m \in M$
$l_{m,n,e,e'}$	1 for $n = 1 \cap (n = 2, e = 2, e' = 1),\ \forall\, m \in M$

The data in Table 3.3 and 3.4 describes a market with two nodes, a transportation facility from node 1 to 2 and a transformation facility in node 2. There are two types of energy, and these can both be transported, but transformation is only possible from type 1 to type 2 at a 40% efficiency rate. Both the transformation and transportation can be expanded by making an investment one time stage ahead. As there are only two time stages, this means that any investment should

Table 3.4: Input data for the parameter $int_{m,n,d,e}$ $\forall\ m \in M$

$\{n, d\}$	Energy type 1	Energy type 2
$\{1, 1\}$	10	10
$\{1, 2\}$	25	30
$\{2, 1\}$	20	10
$\{2, 2\}$	65	80

take place in stage one. In stage 2 there is no gain associated with an investment as this is the last stage considered. From Table 3.4 it can be seen that the demand for both energy types tends to be higher in node 2 than node 1. The network is illustrated in Figure 3.1.

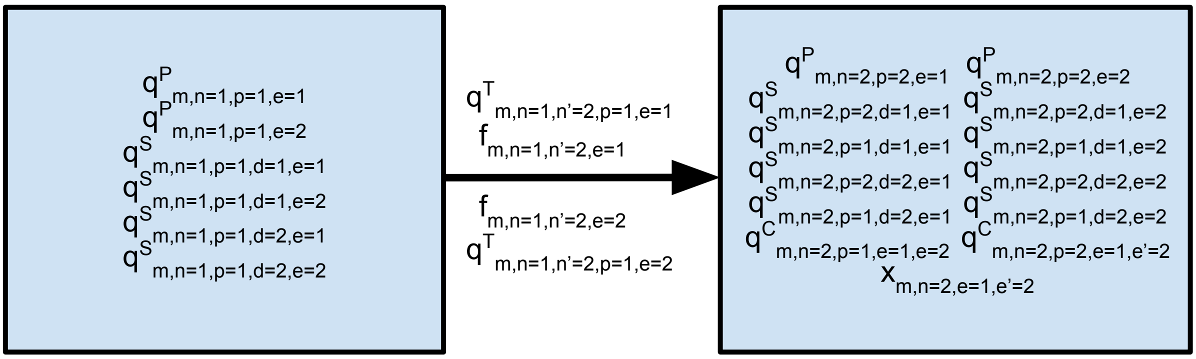


Figure 3.1: Network structure of the example market with primal variables.

The example considered involves variations in the demand for the two types of energy. From the initial state where $int_{1,n,d,e}$ is from Table 3.4, there are three different cases that may occur in the next time stage; a doubling in all values of $int_{m,n,d,e}$ from stage 1 (scenario 1, $m = 2$), all values in stage 1 remain equal to stage 1 (scenario 2, $m = 3$) or a 10% reduction in all values of $int_{m,n,d,e}$ from stage 1 (scenario 3, $m = 4$). The data is illustrated in the below figure of the scenario tree, including probabilities.

3.7.2 Results

The results for this experiment are listed in Table 3.6 below, and a selection of results corresponding to Figure 3.1 are shown in Figure 3.3-3.6.

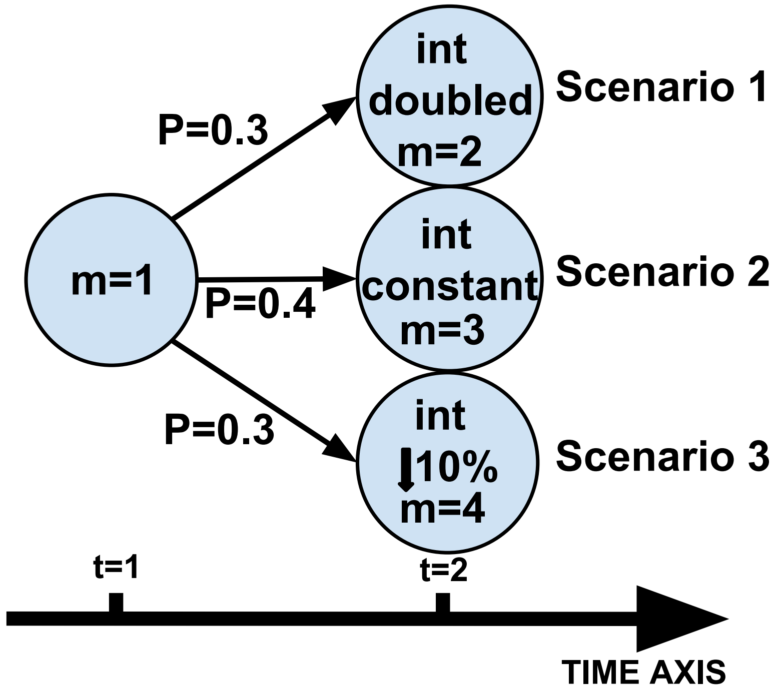


Figure 3.2: Different scenarios when $int_{m,n,d,e}$ varies, for $n \in N, d \in D, e \in E$.

Table 3.6: All non-zero results when varying $int_{m,n,d,e}$

m, n, p, e	*Value of* $q^P_{m,n,p,e}$	m, n, p, d, e	*Value of* $q^S_{m,n,p,d,e}$
$\{1, 1, 1, 1\}$	8.033	$\{1, 1, 1, 2, 1\}$	7.933
$\{1, 1, 1, 2\}$	9.700	$\{1, 1, 1, 2, 2\}$	9.600
$\{1, 2, 2, 1\}$	21.333	$\{1, 2, 1, 2, 1\}$	0.100
$\{1, 2, 2, 2\}$	26.333	$\{1, 2, 1, 2, 2\}$	0.100
$\{2, 1, 1, 1\}$	18.033	$\{1, 2, 2, 2, 1\}$	21.333
$\{2, 1, 1, 2\}$	21.367	$\{1, 2, 2, 2, 2\}$	26.333
$\{2, 2, 2, 1\}$	30.000	$\{2, 1, 1, 2, 1\}$	12.933
$\{2, 2, 2, 2\}$	30.000	$\{2, 1, 1, 2, 2\}$	16.267
$\{3, 1, 1, 1\}$	9.700	$\{2, 2, 1, 2, 1\}$	5.100
$\{3, 1, 1, 2\}$	11.367	$\{2, 2, 1, 2, 2\}$	5.100
$\{3, 2, 2, 1\}$	21.333	$\{2, 2, 2, 2, 1\}$	30.000
$\{3, 2, 2, 2\}$	26.333	$\{2, 2, 2, 2, 2\}$	30.000
$\{4, 1, 1, 1\}$	7.971	$\{3, 1, 1, 2, 1\}$	4.600
$\{4, 1, 1, 2\}$	10.367	$\{3, 1, 1, 2, 2\}$	6.267

Continued on next page...

Table 3.6 – Continued

$\{4,2,2,1\}$	7.700	$\{3,2,1,2,1\}$	5.100
$\{4,2,2,2\}$	23.667	$\{3,2,1,2,2\}$	5.100
		$\{3,2,2,2,1\}$	21.333
m,n,n',e	*Value of* $f^I_{m,n,n',e}$	$\{3,2,2,2,2\}$	26.333
		$\{4,1,1,2,1\}$	5.558
$\{1,1,2,1\}$	5.000	$\{4,1,1,2,2\}$	5.267
$\{1,1,2,2\}$	5.000	$\{4,2,1,2,1\}$	0.732
		$\{4,2,1,2,2\}$	5.772
m,n,e	*Value of* $x^I_{m,n,e}$	$\{4,2,2,1,1\}$	1.600
		$\{4,2,2,2,1\}$	6.100
$\{1.2.2\}$	0.572	$\{4,2,2,2,2\}$	23.667
m,n,n',p,e	*Value of* $q^T_{m,n,n',p,e}$	m,n,n',e	*Value of* $f_{m,n,n',e}$
$\{1,1,2,1,1\}$	0.100	$\{1,1,2,1\}$	0.100
$\{1,1,2,1,2\}$	0.100	$\{1,1,2,2\}$	0.100
$\{2,1,2,1,1\}$	5.100	$\{2,1,2,1\}$	5.100
$\{2,1,2,1,2\}$	5.100	$\{2,1,2,2\}$	5.100
$\{3,1,2,1,1\}$	5.100	$\{3,1,2,1\}$	5.100
$\{3,1,2,1,2\}$	5.100	$\{3,1,2,2\}$	5.100
$\{4,1,2,1,1\}$	2.413	$\{4,1,2,1\}$	2.413
$\{4,1,2,1,2\}$	5.100	$\{4,1,2,2\}$	5.100
m,n,p,e,e'	*Value of* $q^C_{m,n,p,e,e'}$	m,n,e,e'	*Value of* $x_{m,n,e,e'}$
$\{4,2,1,1,2\}$	1.681	$\{4,2,1,2\}$	1.681

3.7.3 Discussion of Results

From the results table, the following observation can be made, supporting the differences in demand in scenario 1, 2 and 3. In general, one can see that the total production and supply are highest at $m = 2$, and lowest at $m = 4$. To see why, summarize all produced quantities at $m = 2, 3, 4$ respectively, which gives: $q^{Ptot}_{m=2} = 18.033 + 21.367 + 30 + 30 = 99.400$, $q^{Ptot}_{m=3} = 9.7 + 11.367 + 21.333 + 26.333 = 68.733$ and $q^{Ptot}_{m4} = 7.971 + 10.367 + 7.7 + 23.667 = 49.7050$. This is realistic due to the corresponding decline in demand.

At stage 1, there are made maximal investments in transportation facilities for both energy types, and a relatively small investment in transformation output

Table 3.5: Input data for the parameter $int_{m,n,d,e}$ in all scenarios.

$\{m,n,d\}$	Energy type 1	Energy type 2
$\{1,1,1\}$	10	10
$\{1,1,2\}$	25	30
$\{1,2,1\}$	20	10
$\{1,2,2\}$	65	80
$\{2,1,1\}$	20	20
$\{2,1,2\}$	50	60
$\{2,2,1\}$	40	20
$\{2,2,2\}$	130	160
$\{3,1,1\}$	10	10
$\{3,1,2\}$	25	30
$\{3,2,1\}$	20	10
$\{3,2,2\}$	65	80
$\{4,1,1\}$	9	9
$\{4,1,2\}$	22.5	27
$\{4,2,1\}$	18	9
$\{4,2,2\}$	22.5	72

capacity ($x^I_{1,2,1,2} = 0.572$) for transformations from energy type 1 to 2.

These decisions are reasonable in comparison to the amounts transported and transformed at $m = 2, 3$ and 4. In scenario 1 and 2, the investments in transportation are fully utilized for both energy types, and at $m = 4$, the type 2 energy is utilizing maximum possible flow, while there is almost 50% idle capacity for type 1 ($f_{4,1,2,1,1} = 2.413$). The transformation capacity is only utilized at $m = 4$. After the investments, the upper limit of transformation output is $0.100 + 0.572 = 0.672$, which corresponds to a maximum input quantity of 1.681 when the efficiency rate for transformations is 40%. This equals the total transformed quantities at $m = 4$, $x_{4,2,1,2} = 1.681$, and hence it can be concluded that the maximum transformation constraint (3.14) is binding. As this is the only scenario for which the transformation facility is used, it is however reasonable that no more investments were made in the first stage.

The market has two demand sectors, but the only case where the demands in sector 1 are met is for $m = 4$. In the other scenarios, quadratic production costs and maximal transportation form network node 1 does probably make the difference in demand in sector 1 and production costs negative. Hence, no sales are made to demand sector 1. For $m = 4$, the quantities produced are lower, and hence, costs are not growing that rapidly, and it is possible to make profits on sales that not pays that much as well.

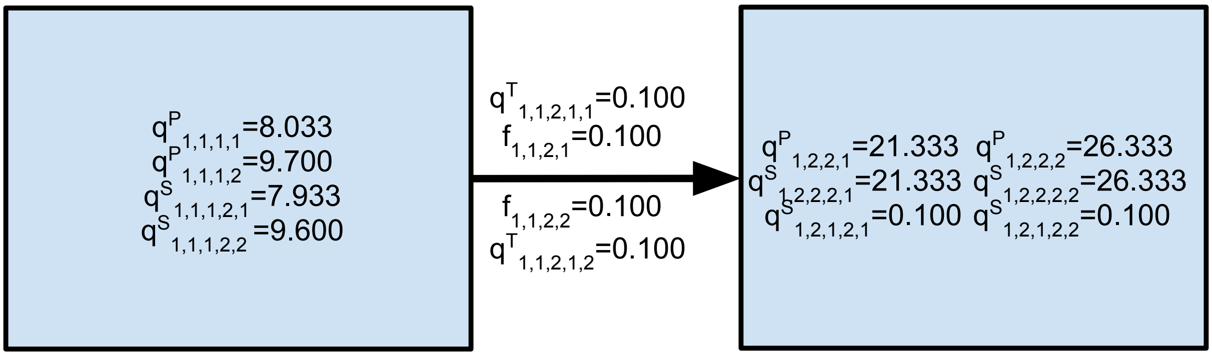


Figure 3.3: Selected non-zero results for $m = 1$.

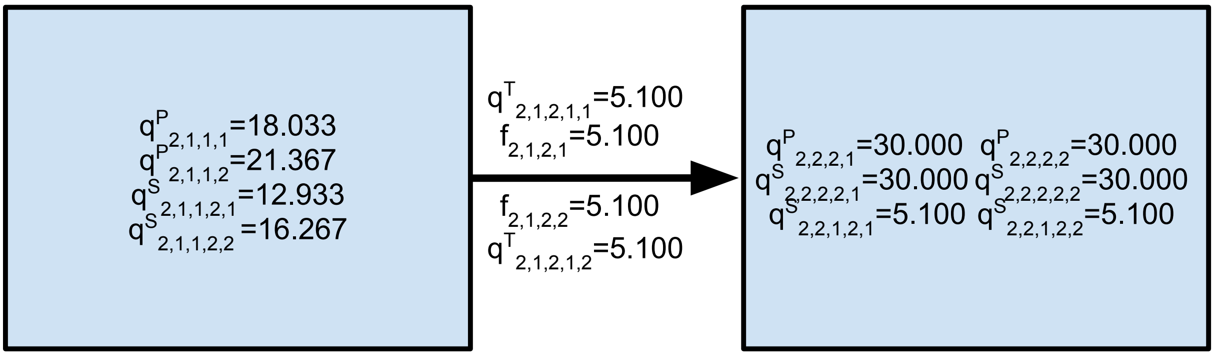


Figure 3.4: Selected non-zero results for $m = 2$.

The same reasoning can explain why transformation is suddenly utilized in scenario 3. Quadratic costs on both transportation and transformation makes it advantageous to transport and transform smaller quantities, rather than one big. It is still only a small quantity that is transformed though, which makes sense with respect to the relatively low efficiency rate (40%) associated with the transformation.

Other opportunities to verify the correctness of the model are limited, but this discussion does at least show that the results seem meaningful.

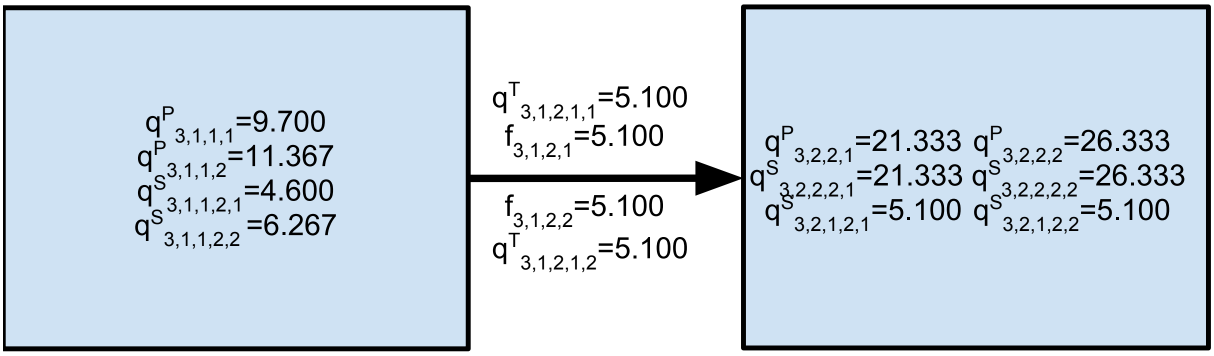


Figure 3.5: Selected non-zero results for $m = 3$.

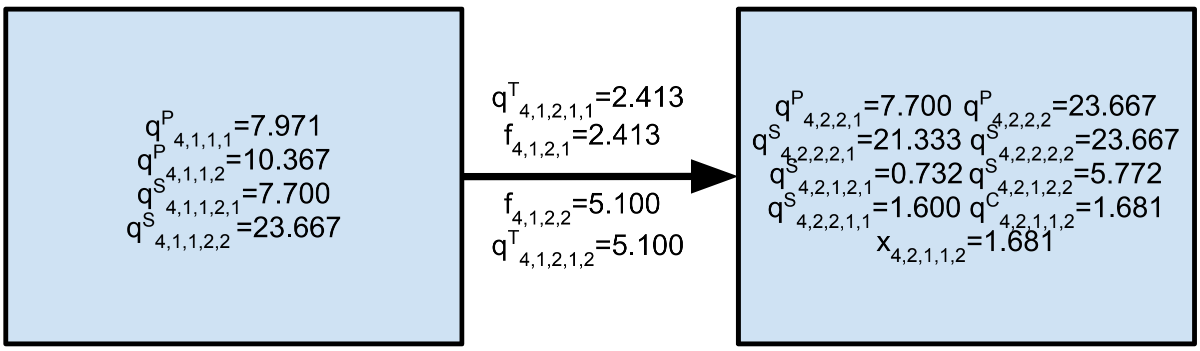


Figure 3.6: Selected non-zero results for $m = 4$.

Chapter 4

Decomposition Techniques

Decomposition techniques are methods where an original problem is reformulated in such a way that several smaller separate problems can be solved iteratively instead.

For suitable problem structures, there are many elements that may reduce total running times and memory demand. First of all, the specific structure of the original problem formulation can be exploited to obtain partial problems that are significantly easier to solve. Sometimes the decomposition can give convex or concave problems even though the original problem has neither of these properties [17]. Another advantage can occur when some or all of the separate problems have a formulation well suitable for especially effective algorithms [17]. Also, the decomposition enables the algorithm for solving decomposed problems to make some computations in parallel. Typically, the required communication between these parallel processes is small relative to the problem sizes, amplifying the possibility to increase speed-up.

Before describing the specific details of deriving a decomposition method for the energy market model from Chapter 3, some common concepts and details that apply to decomposition in general are described.

4.1 Introductory Concepts in Decomposition Techniques

4.1.1 Suitable Problems

There are two types of structures that are especially well suited for basic decomposition techniques; when a problem has either a small number of complicating variables or constraints relative to the total number of variables or constraints respectively. In this context, the word *complicating* means that the problem becomes remarkably easier to solve if these constraints were relaxed, or the variables were fixed to a constant value. Ideally, the relaxation or fixation of variables breaks the problem down to smaller, independent problems, but also in cases where dependent problems arises, decomposition techniques can be beneficial to apply. In an optimization problem, either equality- or inequality constraints, or both can be complicating or include complicating variables.

As an illustration, consider the optimization problem:

$$\begin{aligned} \min_{z} \quad & f(z) \\ \text{st.} \quad & Cz \le a \quad (\lambda). \end{aligned} \tag{4.1}$$

If there were a few variables complicating the problem, and these were organized at the end of the vector z, the matrix C would have a shape similar to (a) in Figure 4.1. Similarly, if the problem (4.1) had a small number of constraints linking a larger number of variables in z that otherwise are not connected, the matrix C could have a form like (b) in the figure.

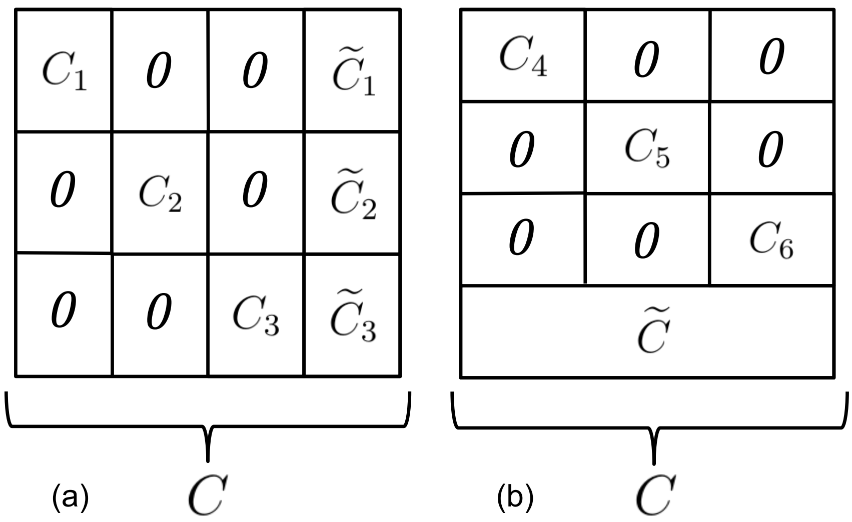


Figure 4.1: Constraint matrices with the presence of complicating variables in (a) and complicating constraints in (b).

Assume that the objective function, $f(z)$, is separable in the complicating and non-complicating variables, and in the variables corresponding to C_1, C_2 and C_3 in case (a) and C_4, C_5 and C_6 in case (b). Then the problem (4.1) would consist of three separate problems that could be solved independently if either the complicating variables in case (a) were fixed to some value, or the complicating constraints in case (b) were relaxed. Such a separation can lead to a substantial reduction in the difficulty of solving the problem, which is a fact that decomposition methods can benefit from.

There are two main types of decomposition techniques that are suitable for problems with decomposable structure as described above. Benders Decomposition can be used for problems containing complicating variables (case (a) in Figure 4.1), and a method called Dantzig-Wolfe Decomposition is appropriate for problems with complicating constraints (case (b) in Figure 4.1). Lagrangian Relaxation is a closely related alternative to Dantzig-Wolfe [7], but is not considered in this report.

Dantzig-Wolfe and Benders Decomposition are often abbreviated DWD and BD respectively, and they are both widely applied to many models of energy markets and in other applications involving optimization theory. In case of an even more complex problem structure, a sort of nested algorithm combining several techniques could also be constructed.

4.1.2 Benders Decomposition (BD) for Linear Optimization Problems

The principles of Benders Decomposition (BD) were first formulated by Benders himself in [6]. In [17], a more generalized method was proposed, often known as *Generalized Benders Decomposition*. In BD, the complicating variables invoking the application of BD are handled in a *master problem* (MP). The corresponding *subproblem* (SP) consist of all terms of the objective function and all constraints from the original problem that do not involve the complicating variables only. Still, where complicating variables appear in the SP, these are fixed to a certain value determined in the previous MP. The SP solution and the Lagrange multipliers to the SP constraints where the MP parameters are present are returned to the MP. Next, a new constraint is added to the MP, based on the information from the SP. The constraint excludes non-optimal parts of the MP feasible region, and is known as a *Benders cut*. As a result of this cut, the MP feasible region is reduced for every iteration. The BD procedure is illustrated in Figure 4.2.

To show details in the MP and SP, reconsider the problem in (4.1) and Figure 4.1 (a), and let the variables in z be organized as follows:

$$z = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \\ \widetilde{z} \end{bmatrix},$$

where $\widetilde{z}$ are the complicating variables. Accordingly, the vectors a and λ and the

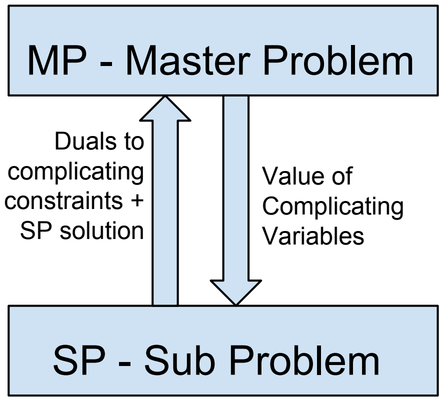


Figure 4.2: Information flow in Benders Decomposition.

function $f(z)$ are;

$$a = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \widetilde{a} \end{bmatrix}, \quad \lambda = \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \widetilde{\lambda} \end{bmatrix} \quad \text{and} \quad f(z) = \begin{bmatrix} f_1(z_1) \\ f_2(z_2) \\ f_3(z_3) \\ \widetilde{f}(\widetilde{z}) \end{bmatrix}.$$

If the variables $\widetilde{z}$ are fixed so that $\widetilde{z} = \widetilde{z}^k$, the problem in (4.1) can be written:

$$\begin{aligned} \min_{z_1,z_2,z_3} \quad & f_1(z_1) + f_2(z_2) + f_3(z_3) \\ \text{st.} \quad & C_1 z_1 \le a_1 - \widetilde{C}_1 \widetilde{z}^k \quad (\lambda_1) \\ & C_2 z_2 \le a_2 - \widetilde{C}_2 \widetilde{z}^k \quad (\lambda_2) \\ & C_3 z_3 \le a_3 - \widetilde{C}_3 \widetilde{z}^k \quad (\lambda_3). \end{aligned} \tag{4.2}$$

This is clearly separable in three independent problems for $i = 1, 2, 3$:

$$\begin{aligned} \min_{z_i} \quad & f_i(z_i) \\ \text{st.} \quad & C_i z_i \le a_i - \widetilde{C}_i \widetilde{z}^k \quad (\lambda_i), \end{aligned} \tag{4.3}$$

which are the SPs. When each of the SPs are solved, the objective function value of all SPs, $\sum_{i=1}^{3} f_i(z_i)$, and the Lagrange multipliers of the constraints in the SPs, λ_i, are passed to the MP. The MP consists of the complicating elements of (4.1), which without any adjustments from the SP would read:

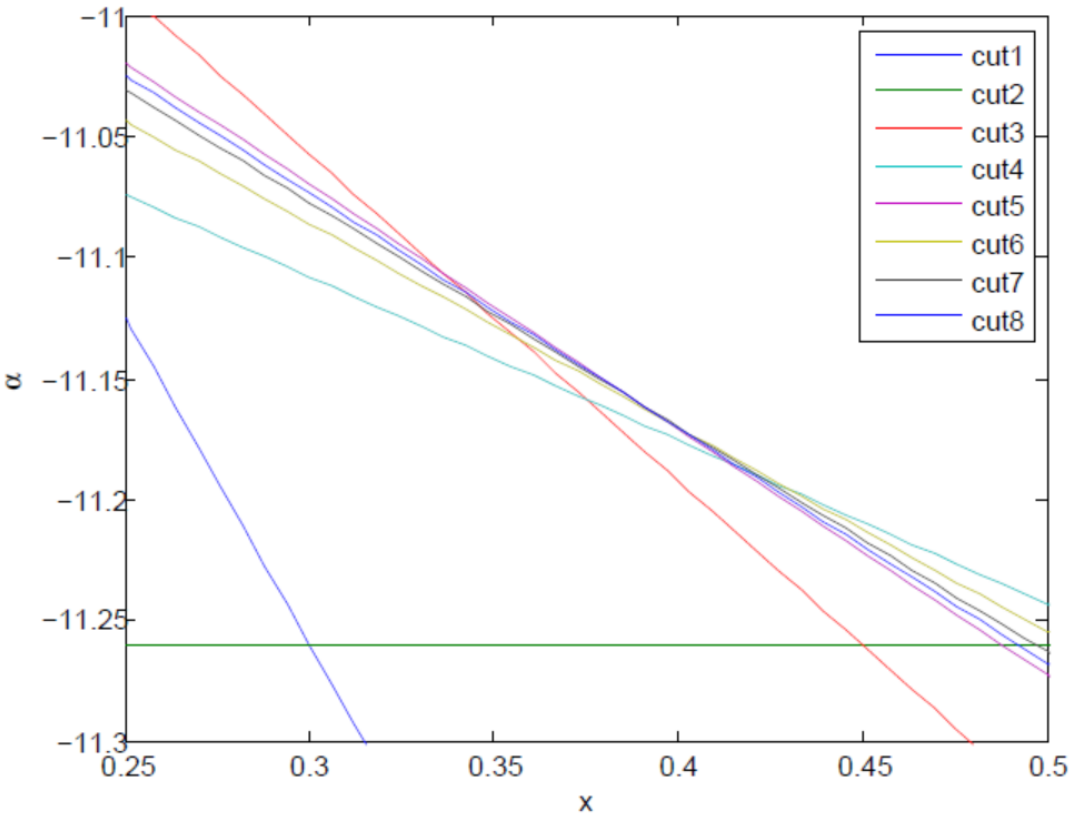
$$\begin{aligned} \min_{\widetilde{z}} \quad & \widetilde{f}(\widetilde{z}) \\ \text{st.} \quad & \widetilde{C}\widetilde{z} \le \widetilde{a}. \end{aligned} \tag{4.4}$$

The SP information is included in a new variable α, which approximates the SP solution. To obtain a $\widetilde{z}^{k+1}$ that is better than the previous in terms of optimizing the SP, α is limited in the MP by the so called Benders cut. This cut is a plane tangent to the previous SP solution, and it prevents the MP from choosing a new $\widetilde{z}^{k+1}$ that would exacerbate the previous SP solution. Hence the MP becomes:

$$\begin{aligned} \min_{\widetilde{z},\alpha} \quad & \widetilde{f}(\widetilde{z}) + \alpha \\ \text{st.} \quad & \widetilde{C}\widetilde{z} \leq \widetilde{a} \\ & \alpha \geq \sum_{i=1}^{3} f_i(z_i) + \sum_{i=1}^{3} \lambda_i(\widetilde{z}^k - \widetilde{z}). \end{aligned} \tag{4.5}$$

At every iteration, a new cut is added, consecutively narrowing the feasible region of the MP. An example plot of such cuts, is shown below. Observe how the cuts become more and more equal throughout the iterations.

Figure 4.3: Illustration of how the cuts are added in a BD method running in 8 iterations.



To summarize this brief description of BD for a generic optimization problem, the variable names are added to Figure 4.2 as shown below.

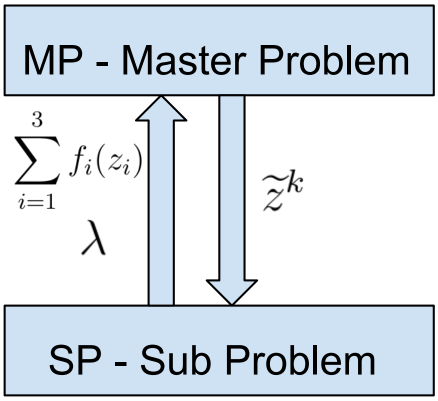


Figure 4.4: Information flow with variables in Benders Decomposition.

Dantzig-Wolfe Decomposition (DWD) for Linear Optimization Problems

Dantzig-Wolfe Decomposition (DWD) by Dantzig and Wolfe [8] was first developed for linear optimization problems.

In brief, DWD has the relaxed problem (according to situation (b) in the previous section) in a the SP. The solution to the SP is passed to the MP where the complicating constraints are represented. When solving the MP, the SP solutions are considered, taking the complicating constraints into account. The MP solution is a convex combination of SP solutions, and this combination has the possibility to include a new element at every iteration as new SP solutions are found. Next, the solution of the dual variables representing the complicating constraints in the MP are passed to the SP so that a better solution (i.e. closer to global optimum) can be found in the next iteration. In this fashion, the SP is considering the complicating constraints as constants, reducing the complexity of the problem. As for BD, the DWD procedure is illustrated in Figure 4.5.

Among extensions to this method, [13] proposed a method for variational inequalities based on the DWD principles, and showed its application to a model of competitive Canadian energy markets. A detailed description of this extension is given later in this chapter.

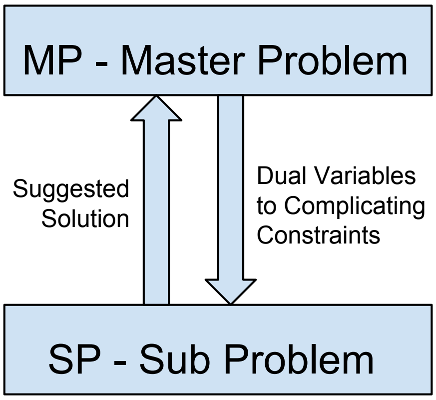


Figure 4.5: Information flow in Dantzig-Wolfe Decomposition.

4.1.3 Relation Between BD and DWD

By investigating the primal and dual problems in Section 2.2.3 (problem (2.2) and (2.4) respectively), one can notice a certain relationship among the variables of one problem and the constraints of the opposite. The variables in the dual problem represent the constraints in the primal and vice versa. Hence the principles of BD and DWD can be interpreted as interconnected in the sense that they are the preferred methods to one version of a problem each. This fact can be exploited to combine the DWD and BD elements, and to develop new methods, which is the key to the following theory.

4.2 Decomposition Methods for VI Problems

The principles of BD and DWD can be used to derive decomposition methods for VI problems. Among papers of relevance to this thesis, [13] describes a DWD procedure for a general class of VI problems, as mentioned earlier. This work was later extended in [14], to a BD method by using duality theory. More specifically, the DWD method was applied to the dual version of a VI problem suited for BD. Furthermore, [16], extended the work in [13] and [14] by formulating a BD procedure that is suitable to stochastic VI problems with complicating variables. Application of this method is appropriate to the energy market model in this report. Details are shown later in this chapter, but to provide a good understanding of the underlying concepts, the details in [13] [14] are described first.

4.2.1 DWD for VI Problems

This section gives a summary of the DWD method for VI problems by [13].

With the intent to decompose a VI problem with complicating constraints, a VI problem notation that includes a separation of complicating constraints, $h(z) \geq 0$, and non-complicating constraints, $g(z) \geq 0$, is convenient. For the set $K = \{z \in \mathbb{R}^n \mid g(z) \geq 0, h(z) \geq 0\}$ and $G : \mathbb{R}^n \to \mathbb{R}^n$, the problem considered, $VI(K, G)$, is:

$$\text{Find } z^* \in K \text{ s.t. } G(z^*)^T(z - z^*) \geq 0, \;\; \forall z \in K. \tag{4.6}$$

Before stating the algorithm, a few assumptions are made for the problem (4.6);

- A solution exists.
- $g(z) \geq 0$ and $h(z) \geq 0$ are concave and continuously differentiable.

The above assumptions are true in many applications and will be accepted without any further discussion. In addition, a few more assumptions are made regarding the robustness of the algorithm.

- The SP has a feasible solution.
- The first MP has a feasible solution, and therefore all MPs are feasible.

The above assumptions are discussed later in this section.

Subproblem

The first instance investigated is the SP at iteration k (k will denote iteration counter throughout the report). This problem is like the original (4.6), but the complicating constraint, $h(z) \geq 0$, is evaded. Hence, the feasible region to the SP is:

$$\overline{K} = \{z \in \mathbb{R}^n \mid g(z) \geq 0\}. \tag{4.7}$$

The set $\overline{K}$ remains constant throughout the entire process. The mapping in the SP, G_S, takes the dual variables, β^{k-1}, form the most recent solution of the MP for $k-1$ into account, multiplied with the gradient of $h(z)$ at the most recent MP solution, z_M^{k-1}. In this way, the SP can take the complicating constraints into account implicitly through the information in $\nabla h(z_M^{k-1})^T \beta^{k-1}$. The SP mapping is:

$$G_S = G - \nabla h(z_M^{k-1})^T \beta^{k-1}. \tag{4.8}$$

And thereby, the SP at iteration k is:

$$\text{Find } z_S^k \in \overline{K} \text{ s.t. } \left(G(z_S^k) - \nabla h(z_M^{k-1})^T \beta^{k-1}\right)^T (z - z_S^k) \geq 0, \;\; \forall z \in \overline{K}. \tag{4.9}$$

At $k = 1$, there is no MP solution, and thus $\nabla h(z_M^0)^T \beta^0 = 0$. Therefore, the first SP is a relaxation of the original problem (4.6) in which no penalty occurs.

Master Problem

The MP solution is a convex combination of all SP solutions obtained so far that is feasible with respect to the complicating constraint $h(z) \geq 0$. Recall that feasibility of the MP is an assumption already made.

For the purpose of obtaining a convex combination, it is convenient to store all SP solutions obtained at k in a matrix, Φ_S^k. In Φ_S^k, a new column is added containing the latest SP solution, z_S^k, at every iteration.

$$\Phi^k \equiv \begin{bmatrix} z_S^1 & \cdots & z_S^i & \cdots & z_S^k \end{bmatrix}. \tag{4.10}$$

To express the convex combination of SP solutions, the vector λ of length k that assigns weight to each of the SP solutions is introduced. At every iteration, the elements of λ^k sum up to 1. Accordingly, the MP solution, z_M^k is $z_M^k = \Phi_S^k \lambda^k$, and the MP feasible region is:

$$\Lambda^k = \{\lambda \in \mathbb{R}^n \mid h(\Phi_S^k \lambda^k) \geq 0, e^{kT}\lambda = 1, \lambda \geq 0\}, \tag{4.11}$$

where $e^k \in \mathbb{R}^k$ is a vector of all ones. In contrast to the set $\overline{K}$, Λ^k is dynamic, in the sense that it is enlarged at every iteration as new SP solutions are added to Φ^k. If the MP was solved for z_M^k explicitly, the mapping G from (4.6) would also apply to the MP, but because the MP is solved for λ^k, the notation of the mapping can be modified. Without modification, the kth MP would read:

$$\text{Find } \lambda^k \in \Lambda^k \text{ s.t. } G(\Phi_S^k \lambda^k)^T (\Phi_S^k \lambda - \Phi_S^k \lambda^k) \geq 0, \quad \forall \lambda \in \Lambda^k.$$

Instead, let $H(\Phi_S^k \lambda)^T = G(\Phi_S^k \lambda)^T \Phi_S^k$, so that the MP becomes:

$$\text{Find } \lambda^k \in \Lambda^k \text{ s.t. } H^k(\lambda^k)^T (\lambda - \lambda^k) \geq 0, \quad \forall \lambda \in \Lambda^k. \tag{4.12}$$

From this point, the dual variable β^k to the complicating constraint $h(z) \geq 0$ is passed to the SP together with the gradient of $h(\Phi_S^k \lambda^k)$, so that the next SP can be generated and solved.

Convergence Gap

For this method, a convergence gap, CG^k, is defined as:

$$CG^k = \left(G(z_M^{k-1}) - \nabla h(z_M^{k-1})^T \beta^{k-1}\right)^T (z_S^k - z_M^{k-1}). \tag{4.13}$$

This scalar quantity is used to determine when a solution of sufficient precision is obtained. That is when $CG^k > -\epsilon$, where ϵ is a specified scalar determining the tolerance level. The ϵ enables the algorithm to stop when a solution proposal is sufficiently close to equilibrium. It should be noted that the convergence gap not necessarily increases monotonically.

The above convergence gap is a generalization of the stopping conditions used in DWD for optimization problems. Furthermore, [13] shows that if the considered VI problem is actually an optimization problem, the two methods have the same convergence gap. For optimization problems, this scalar quantity is meaningful in the sense that it measures the gap between the primal and dual solutions. For VI problems, there is no such counterpart, but the above CG^k can be interpreted as a quantity describing the proximity to an equilibrium solution. The convergence gap and properties for which convergence can be attained, is described after the below discussion of feasibility.

Feasibility

In the above description, it is assumed that the MP is feasible, and in particular that the very first MP is feasible. It is worth noting that if the first MP solution is feasible, the same applies to the remaining iterations. This is because the first MP feasible region, Λ^1 is a subset of any later Λ^k. In the opposite case, if the first MP is infeasible, there are a few technicalities that could be introduced to make the feasibility assumption hold, see [13] and [16]. This involves the introduction of some artificial variables in the MP and artificial bounds in the SP. The artificial bounds in the SP does also provide boundedness of $\overline{K}$, so that feasibility of the SP is ensured.

Convergence

It is proved in [13] (Theorem 5) that if there is a SP solution, z_S^k, such that $CG^k = \big(G(z_M^{k-1}) - \nabla h(z_M^{k-1})^T \beta^{k-1}\big)^T (z_S^k - z_M^{k-1}) < 0$ for $z_S^k \in \overline{K}$, the solution of the next SP will expand the set Λ^{k+1}. Therefore, the algorithm should continue if $CG^k < 0$. It is also proved that if an MP solution, z_M^{k-1}, solves the SP at iteration k, z_M^{k-1} is also a solution to the initial problem, $VI(K, G)$. That is $\big(G(z_M^{k-1}) - \nabla h(z_M^{k-1})^T \beta^{k-1}\big)^T (z - z_M^{k-1}) \geq 0 \ \ \forall z \in \overline{K}$.

So far, nothing has been required for G, and it cannot be guaranteed that an MP solution that solves the next SP will ever be found. To consider convergence, [13] provides two more theorems of relevance. First, if G is strictly monotone, or if G is such that $G(z) = \binom{F(q)}{\nabla c(x)}$, and F is strictly monotone and $c(x)$ is continuous, it is proved that a solution is obtained if $CG^k \geq 0$. Next, it is proved that if $CG^k \geq 0$ not occurs within a finite number of iterations, and if G is continuous and any infinite sub sequence of $\{(z_M^k, \beta^k, z_S^{k+1})\}_{k=1}^{\infty}$ has at least one limit point, then $\lim_{k\to\infty} CG^k = 0$.

4.2.2 Benders Decomposition for VI Problems

The first studies of Benders Decomposition applied to a class of problems where both the MP and SPs are VI problems was published by [20]. Later, and with several other contributions in between, [14] introduced a similar method that requires milder assumptions, and provided a proof of convergence for realistic problems found in many applications. Because it is a good presentation of BD for general VI problems, and because it is the main source for the supplementary work in [16], a short summary is given here.

As mentioned, the BD method in [14] is derived by applying the DWD method from [13] to the Lagrangian dual version of the original problem (with complicating variables). Hence, convergence is attained according to the discussion in the previous section, and when the following assumptions hold:

- The primal VI problem has a solution.
- F is invertible.
- F is continuous.

These assumptions are necessary to make the transformation from primal to dual form without abandoning essential properties of the problem. The dual version of the involved VI problem is obtained through the KKT conditions, similarly to the steps discussed in Section 2.4.2 and shown in Section 3.4.1.

The feasible region to the primal problem considered is:

$$K_P = \{(q, x, y) \mid Aq + Bx + Dy \geq c, x \geq 0, y \geq 0\}. \tag{4.14}$$

Note that the above constraints are linear, in contrast to the definition of K in (4.6).

In (4.14) the vectors x and y are non-negative, while q is free in sign and occurs in non-constant terms of the mapping for the considered VI problem. All complicating variables are contained in y. The matrices A, B and D are of dimension suiting their respective variables in (4.14). With the mapping $(F(q)^T, d^T, l^T)$, the primal VI problem is:

$$\begin{aligned} &\text{Find } (q^*, x^*, y^*) \in K_P \text{ s.t.} \\ &F(q^*)^T(q - q^*) + d^T(x - x^*) + l^T(y - y^*) \geq 0, \quad \forall (q, x, y) \in K_P. \end{aligned} \tag{4.15}$$

In order to derive the dual of (4.15), the variable $f^* = F(q^*)$ and the inverse mapping, $F^{-1}(f^*) = q^*$, is introduced as described in Section 3.4.1. Inserting f into the KKT conditions for (4.15) gives the following dual VI problem:

$$\text{for } K_D = \{(f, v) \mid -A^T v + f^T = 0, v^T B \le d^T, v^T D \le l^T, v \ge 0\}, \tag{4.16}$$

$$\begin{aligned} &\text{find } (f^*, v^*) \in K_D \text{ s.t.} \\ &F^{-1}(f^*)^T(f - f^*) - c^T(v - v^*) \ge 0, \;\; \forall (f, v) \in K_D. \end{aligned} \tag{4.17}$$

Subproblem

According to the variables y being the primal complicating variables, the constraint $v^T D \le l^T$ is a complicating constraint in (4.17). Application of the DWD method from Section 4.2.1 gives the following SP corresponding to (4.9) in the previous section:

$$\overline{K}_D = \{(f, v) \mid v^T A - f^T = 0, v^T B \le d^T, v \ge 0\}, \tag{4.18}$$

$$\begin{aligned} &\text{find } (f_S^T, v_S^T) \in \overline{K}_D \text{ s.t.} \\ &F^{-1}(f_S^k)^T(f - f_S^k) - (c - Dy_M^{k-1})^T(v - v_S^k) \ge 0, \;\; \forall (f, v) \in \overline{K}_D, \end{aligned} \tag{4.19}$$

where the superscript k, as before, denotes the iteration counter. The matrix of SP solutions corresponding to Φ_S^k in problem (4.12) becomes:

$$\Phi^k \equiv \begin{bmatrix} f_S^1 & \cdots & f_S^i & \cdots & f_S^k \\ v_S^1 & \cdots & v_S^i & \cdots & v_S^k \end{bmatrix} \equiv \begin{bmatrix} \Phi_f^k \\ \Phi_v^k \end{bmatrix}. \tag{4.20}$$

Master Problem

Like in equation (4.11), the vector e^{kT} of all ones and length k is introduced, so that the (dynamic) feasible region to the restricted MP is:

$$\Lambda^k = \{\lambda \mid D^T \Phi_v^k \lambda \le l^T, e^{kT}\lambda = 1, \lambda \ge 0\}. \tag{4.21}$$

Furthermore, the dual master problem becomes:

$$\begin{aligned} &\text{Find } \lambda^k \in \Lambda^k \text{ s.t.} \\ &(F^{-1}(\Phi_f^k \lambda^k)^T \Phi_f^k - c^T \Phi_v^k)(\lambda - \lambda^k) \ge 0, \;\; \forall\, \lambda \in \Lambda^k. \end{aligned} \tag{4.22}$$

Now, the above dual MP (4.22) and SP (4.19) can be converted back to primal form. This is again done following Section 3.4.1. For the SP, the feasible region is defined as follows, with the complicating variable y fixed to y_M^{k-1}:

$$K_P(y_M^{k-1}) = \{(q, x) \mid Aq + Bx \ge c - Dy_M^{k-1}, x \ge 0\}, \tag{4.23}$$

and the SP is:

$$\begin{aligned} &\text{Find } (q_S^k, x_S^k) \in K_P(y_M^{k-1}) \text{ s.t.} \\ &F(q_S^k)^T(q - q_S^k) + d^T(x - x_S^k) \ge 0, \;\; \forall (q, x) \in K_P(y_M^{k-1}). \end{aligned} \tag{4.24}$$

The transformation of the MP back to primal form entails the introduction of a new variable, θ. This is the Lagrange multiplier, which together with y_M^k occurs in the KKT conditions to the dual MP. θ is associated with the constraint $e^{kT}\lambda = 1$ in the set Λ^k, while y_M^k is the vector of multipliers associated with $\lambda^T \Phi_v^{kT} D \leq l^T$. Thus, the set K_P^k is defined:

$$K_P^k = \{(q, y, \theta) \mid \Phi_f^{kT} q + \Phi_v^{kT} D y + e^k \theta \geq \Phi_v^{kT} c, y \geq 0\}, \tag{4.25}$$

and the primal MP becomes:

$$\begin{aligned} &\text{Find } (q_M^k, y_M^k, \theta^k) \in K_P^k \text{ s.t.} \\ &F^T(q_M^k)(q - q_M^k) + l^T(y - y_M^k) + (\theta - \theta^k) \geq 0, \quad \forall (q, y, \theta) \in K_P^k. \end{aligned} \tag{4.26}$$

Convergence Gap

Corresponding to the convergence gap for DWD in equation (4.13), this BD algorithm terminates when:

$$CG^k = F^{-1}(f_M^k)^T f_S^{k+1} - (c - D y_M^k)^T (v_S^{k+1} - v_M^k) > -\epsilon. \tag{4.27}$$

Again, ϵ is a scalar defining when the accuracy of the solution is sufficient.

This concludes the BD method in [14]. Details on convergence will be provided later when discussing the further extensions of BD for VI problems.

4.3 BD for Stochastic Equilibrium Problems

Based on [13] and [14], [16] complement the BD algorithm for VI the problems in [14] by introducing a few changes that tailor the method to stochastic equilibrium problems. More precisely, there are three changes of major importance. Firstly, the notation of the DWD from [13] is changed to facilitate a decomposition where the only original primal variables that appear in the MP are complicating. Such a separation allows for a better performance of the algorithm by reducing the MP as much as possible. In terms of the notation in the previous section, this means that the q_M^k in the primal MP (4.26) is evaded. The variables in q (both q_M^k and q_S^k) are instead solved in the SP, while y is the only primal variable to be solved in the MP. Details of this extension are shown in a separate subsection below.

The second change of relevance is to allow for constraints that, due to the decomposition, appear in the MP only. With the notation in [14], such a separation is not possible, because the original (primal) problem does not explicitly distinguish between different types of constraints.

Lastly, [16] formulates a distinction between equality and inequality constraints of the primal problem. The proposed BD algorithm is applicable to the problem described in Chapter 3, which is similar to the example problem given in [16].

In the following, the extensions to the DWD method facilitating the desired split of MP and SP variables and constraints assigned to the MP only are shown. Next, the BD method from [16] is described by applying it to the generalized problem (3.24). In this way, the level of complexity is reduced by leaving out a few elements that are redundant to the present application. The distinction of inequality and equality constraints in the primal problem does not involve any new theory, and is therefore only shown through the application.

4.3.1 Extensions to the DWD

Recall the initial VI problem (4.6) in the DWD from Section 4.2.1. Now, the variable $z \in \mathbb{R}^n$ is separated to $z_1 \in \mathbb{R}^{n_1}$ and $z_2 \in \mathbb{R}^{n_2}$. The vector z_1 will be a variable in the MP, while z_2 is determined in the SP. With respect to this separation, the mapping $G(z) : \mathbb{R}^n \to \mathbb{R}^n$ is $G(z) = \binom{G_1(z_1)}{G_2(z_2)}$, where $G_1(z_1) : \mathbb{R}^{n_1} \to \mathbb{R}^{n_1}$ and $G_2(z_2) : \mathbb{R}^{n_2} \to \mathbb{R}^{n_2}$. Furthermore, the (non-complicating) constraint $g(z) \geq 0$ is only dependent on z_2, while the complicating constraint $h(z) \geq 0$ depends on both z_1 and z_2. It is assumed that $h(z)$ is separable in z_1 and z_2 so that $h(z) = h_1(z_1) + h_2(z_2)$, and the dual variable to this constraint is still β.

Due to the above distinctions, the SP corresponding to (4.9) is:

$$\text{Find } z_{2S}^k \in \overline{K} \text{ s.t. } \big(G_2(z_{2S}^k) - \nabla h_2(z_{2M}^{k-1})^T \beta^{k-1}\big)^T (z_2 - z_{2S}^k) \geq 0, \quad \forall z \in \overline{K}. \tag{4.28}$$

Where $\overline{K} = \{z_2 \mid g(z_2) \geq 0\}$, and z_{2M}^{k-1} and β^{k-1} are fixed to the previous solutions from the MP at iteration $k-1$. The SP solutions are collected in a matrix $Z_{2S}^k = [z_{2S}^1, z_{2S}^k, ..., z_{2S}^k]$ and passed to the MP.

The MP variable λ gives the weights to the SP solutions in Z_{2S}^k, so that $z_{2M}^k = \Phi_{2M}^k \lambda^k$. In addition, the variable z_1 is also a variable in the MP. Hence, the feasible region to the current MP is:

$$\Lambda^k = \{z_1 \in \mathbb{R}^{n_1}, \lambda \in \mathbb{R}^k \mid h_1(z_1) + h_2\big(Z_{2S}^k \lambda\big) \geq 0, e^{kT}\lambda = 1, \lambda \geq 0\}, \tag{4.29}$$

and the MP is:

$$\begin{aligned} &\text{Find } (z_1^k, \lambda^k) \in \Lambda^k \\ &\text{s.t. } G_1(z_1^k)^T (z_1 - z_1^k) + G_2(Z_{2S}^k \lambda^k)^T Z_{2S}^k (\lambda - \lambda^k) \geq 0, \quad \forall (z_1, \lambda) \in \Lambda^k. \end{aligned} \tag{4.30}$$

4.3.2 BD Applied to the Energy Market Model

The generalized VI problem for the energy market from Section 3.4 is considered:

$$\begin{aligned}
&\text{For} \quad K = \left\{ \begin{array}{r} (q,x,y) \mid \qquad\qquad \\ \widetilde{D}y \geq \widetilde{c} \ \ (v_1) \\ \overline{A}q + \overline{D}y \geq \overline{c} \ \ (v_2) \\ \hat{A}q + \hat{B}x = 0 \ \ (v_3) \\ q \geq 0, \ x \geq 0, \ y \geq 0 \end{array} \right\}, \\
&\text{find } (q^*, x^*, y^*) \in K \text{ s.t.} \\
&F^T(q^*)(q - q^*) + d^T(x - x^*) + l^T(y - y^*) \geq 0, \ \ \forall (q,x,y) \in K.
\end{aligned} \tag{4.31}$$

Again, the dual variables are included in the notation of K_D in order to emphasize the correspondence with each of the constraints, according to the notation in equation (3.29). From earlier, it is known that BD is appropriate to problems with a complicating variable. Looking back at the derivation of the general notation for the energy market model in Section 3.4, it should be clear that complicating variables are present here. More precisely, the variables $f^I_{m,n,n',e}$ and $x^I_{m,n,e}$ stored in y, prevent the problem form being separated into a multiple of distinct problems, one for each scenario tree node. Thus, y complicates the problem.

As described in Section 4.2.2 the first step is to obtain the dual version of the considered problem. From the Sections 3.3 and 3.5, it is already known that the objective functions in the optimization problems are continuous, and invertible, and thus the same applies to $F(q)$ in (4.31). Furthermore it is shown in Section 3.5 that the original VI problem has at least one solution, and hence it can be concluded that the assumptions in [14] and Section 4.2.2 regarding the primal-dual transformation are satisfied. The dual VI problem is already found in Section 3.4.1:

$$\begin{aligned}
&\text{Find } (v_1^*, v_2^*, v_3^*, f^*) \in K_D \text{ s.t.} \\
&-\widetilde{c}^T(v_1 - v_1^*) - \overline{c}^T(v_2 - v_2^*) + F^{-1}(f^*)^T(f - f^*) \geq 0, \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad \forall (v_1, v_2, v_3, f) \in K_D,
\end{aligned} \tag{4.32}$$

with

$$K_D = \left\{ \begin{array}{l} (v_1, v_2, v_3, f) \mid \\ \widetilde{D}^T v_1 + \overline{D}^T v_2 \leq l \quad (y) \\ \overline{A}^T v_2 + \hat{A}^T v_3 \leq f \quad (q) \\ \qquad\qquad \hat{B}^T v_3 \leq d \quad (x) \\ \qquad\quad v_1 \geq 0, v_2 \geq 0 \end{array} \right\}. \tag{4.33}$$

To derive the MP and SP, the dual variables v_1 are categorized into z_1, and $z_2 = [v_2^T, v_3^T, f^T]^T$. Hence the complicating constraint, involving both z_1 and z_2 variables is $\widetilde{D}^T v_1 + \overline{D}^T v_2 \leq l$. To suit the form $h(z) \geq 0$ let $h(z) = l - \widetilde{D}^T v_1 - \overline{D}^T v_2 \geq 0$. As the complicating constraint is relaxed in the SP, but present in the MP, v_1 is solved in the MP along with the weight variables λ. In the primal problem (4.31), the constraint $\widetilde{D}y \geq \widetilde{c}$, for which v_1 is the associated multiplier vector, involves complicating variables (y) only. Hence, the presence of v_1 in the MP is natural in a BD algorithm. It is the separation of z_1 and z_2 that facilitates this adaptation.

The Dual Subproblem

When the constraint $\widetilde{D}^T v_1 + \overline{D}^T v_2 \leq l$ is relaxed, the feasible region to the SP becomes:

$$\overline{K}_D = \left\{ \begin{array}{r} (v_2, v_3, f) \mid \\ \overline{A}^T v_2 + \hat{A}^T v_3 \leq f \quad (q) \\ \hat{B}^T v_3 \leq d \quad (x) \\ v_2 \geq 0 \end{array} \right\}, \tag{4.34}$$

and the mapping in (4.32) is separated as follows:

$$G(v_1, v_2, v_3, f) = \begin{pmatrix} G_1(v_1) \\ G_2(v_2, v_3, f) \end{pmatrix} = \begin{pmatrix} -\widetilde{c} \\ -\overline{c} \\ 0 \\ F^{-1}(f) \end{pmatrix} \begin{matrix} \} G_1 \\ \\ \left.\begin{matrix} \\ \\ \\ \end{matrix}\right\} G_2. \end{matrix} \tag{4.35}$$

The dual variable to $h(v_1, v_2, v_3, f)$ is y, and the gradient of $h(v_1, v_2, v_3, f)$ is $\nabla h(v_1, v_2, v_3, f) = [-\widetilde{D}, -\overline{D}, 0, 0]$. This is used together with equation (4.35) so that the SP mapping corresponding to (4.28) becomes:

$$G_2(z_{2S}^k) - \nabla h(z_{2M}^{k-1})\beta_M^{k-1} = \begin{pmatrix} -\overline{c} \\ 0 \\ F^{-1}(f_S^k) \end{pmatrix} - \begin{pmatrix} -\overline{D}y_M^{k-1} \\ 0 \\ 0 \end{pmatrix}. \tag{4.36}$$

Thereby, the dual SP is:

$$\begin{aligned} &\text{Find } (v_{2,S}^k, v_{3,S}^k, f_S^k) \in \overline{K}_D \text{ s.t.} \\ &(-\overline{c} + \overline{D}y_M^{k-1})^T (v_2 - v_{2,S}^k) \\ &+ F^{-1}(f)^T (f - f_S^k) \geq 0, \ \ \forall (v_2, v_3, f) \in \overline{K}_D. \end{aligned} \tag{4.37}$$

The solutions to (4.37) are stored in a matrix like Z_{2S}^k, but to simplify notation, the matrix is denoted Φ^k, as in (4.20) and (4.10). The structure of Φ^k is:

$$\Phi^k = \begin{bmatrix} v_{2,S}^1 & \cdots & v_{2,S}^i & \cdots & v_{2,S}^k \\ f_S^1 & \cdots & f_S^i & \cdots & f_S^k \end{bmatrix} = \begin{bmatrix} \Phi_{v2}^k \\ \Phi_f^k \end{bmatrix}. \tag{4.38}$$

For later use, the dual variables to (4.37), i.e. q and x are stored in a similar matrix, where Φ_q^k and Φ_x^k are the rows.

The Dual Master Problem

The set Λ^k according to (4.11) which is the feasible region to the not yet described MP:

$$\Lambda^k = \{(v_1, \lambda) \mid l - \widetilde{D}^T v_1 - \overline{D}^T \Phi^k_{v2}\lambda \geq 0, e^{kT}\lambda = 1, v_1 \geq 0, \lambda \geq 0\}. \tag{4.39}$$

Let θ be the Lagrange multiplier to the constraint $e^{kT}\lambda = 1$. At this point, the original G_2 mapping is kept in its original form for consistency with G_1 in the MP. Hence, by giving $(\Phi^k\lambda)$ as input to the mapping G_2 from (4.35), the MP is:

$$\begin{aligned}
&\text{Find } (v_1^k, \lambda^k) \in \Lambda^k \\
&\text{s.t. } -\widetilde{c}^T(v_1 - v_1^k) - \overline{c}^T(\Phi^k_{v2}\lambda - \Phi^k_{v2}\lambda^k) \\
&+ F^{-1}(\Phi^k_f\lambda^k)^T(\Phi^k_f\lambda - \Phi^k_f\lambda^k) \geq 0, \;\; \forall (v_1, \lambda) \in \Lambda^k
\end{aligned} \tag{4.40}$$

The Primal Subproblem

The primal SP is found by following the steps in Section 3.4.1:

$$\begin{aligned}
&\text{Find } (q_S^k, x_S^k) \in K_P(y_M^{k-1}) \\
&\text{s.t. } F(q_S^k)^T(q - q_S^k) + d^T(x - x_S^k) \geq 0, \;\; \forall (q, x) \in K_P(y_M^{k-1}),
\end{aligned} \tag{4.41}$$

with

$$K_P(y_M^{k-1}) = \{(q, x) \mid \overline{A}q \geq \overline{c} - \overline{D}y_M^{k-1}, \; \hat{A}q + \hat{B}x = 0, \; q \geq 0, x \geq 0\}.$$

MCP formulation of SP and MP

The KKT conditions to the primal SP (4.41) are:

$$0 \leq q \perp F(q) - \overline{A}^T v_2 - \hat{A}^T v_3 \geq 0 \tag{4.42a}$$

$$0 \leq x \perp d - \hat{B}^T v_3 \geq 0 \tag{4.42b}$$

$$0 \leq v_2 \perp \overline{A}q + \overline{D}y_M^{k-1} - \overline{c} \geq 0 \tag{4.42c}$$

$$v_3 \; free, \quad \hat{A}q + \hat{B}x = 0 \tag{4.42d}$$

It turns out, that both the primal SP and its KKT conditions have several similarities with the original problem (4.31). Comparison with (3.23) shows that the only difference in (4.42) is that the variable y is fixed to its MP solution y_M^{k-1}. This is not surprising, but a valuable fact in terms of implementing the algorithm.

The transformation of the MP back to its primal form does not reveal the same similarities. Nor is the primal version presented in [16], because the primal MP would invoke the variable q back into the MP, breaking down the advantage of having a MP with complicating variables only. Hence this section is confined to deriving the KKT conditions of the dual MP:

$$0 \leq y_M^k \perp l - \widetilde{D}v_1 - \overline{D}\Phi_{v2}^k\lambda^k \geq 0 \tag{4.43a}$$
$$0 \leq \lambda^k \perp -\Phi_{v2}^{kT}\overline{c} + \Phi_f^{kT}F^{-1}(\Phi_f^k\lambda^k) + \Phi_{v2}^{kT}\overline{D}y_M^k + \theta \geq 0 \tag{4.43b}$$
$$0 \leq v_1 \perp -\widetilde{c} + \widetilde{D}y_M^k \geq 0 \tag{4.43c}$$
$$\theta \; free, \quad -e^{kT}\lambda^k + 1 = 0 \tag{4.43d}$$

Convergence Gap

The last step to complete the method description is to find the convergence criterion. This is defined for DWD in (4.13) and gives:

$$CG^k = \big(F(q^k) - F(\Phi_q^k\lambda^k)\big)^T\Phi_q^k\lambda^k - \big(v_2 - \Phi_{v2}^k\lambda^k\big)^T(\overline{c} - \overline{D}y_M^k). \tag{4.44}$$

The criteria to guarantee convergence in this BD method are the same as in [13] and Section 4.2.1, but the additional requirements concerning the primal-dual transformation in [14] and Section 4.2.2 must also be included.

All the convergence results given in [16] are in compliance with the preceding papers, but one of the proofs is different, due to the separation in z_1 and z_2 and in G_1 and G_2. It turns out that it is sufficient to require either G_2 strictly monotone, or $F(q)$ in $G_2 = \binom{F(q)}{\nabla c(x)}$ strictly monotone and $c(x)$ a convex function to guarantee that a solution is found when $CG^k \geq 0$. Furthermore, to provide the limit $\lim_{k\to\infty} CG^k = 0$, the requirement that any infinite subsequence of $\{(z_M^k, \beta^k, z_S^{k+1})\}_{k=1}^{\infty}$ has at least one limit point remains unchanged, but it is sufficient to require G_2 continuous.

A discussion of how the convergence criteria applies to the present application is given later, after the discussion of a few summarizing features.

Stepwise Description of Algorithm

To summarize and provide overview, the algorithmic steps are listed below.

Step 0 - Initialization

Set $\epsilon > 0$ and let $k = 0$, $\Phi^0 = \mathbf{0}$ and $\nabla h(z_{2M}^0)\beta^0 = 0$

Step 1

Solve the SP (4.42) at $k + 1$.
If: $k = 0$ and SP infeasible; Stop

Else: Update the matrix Φ^{k+1} in (4.38) by adding the latest SP solution as a new column to Φ^k.
If: $k = 0$; go to step 3

Step 2
Stop if $CG^k > -\epsilon$.

Step 3
Set $k = k + 1$.
Solve MP (4.43) at k.
Go to step 1.

The steps above are also applicable to the DWD from Section 4.2.1 and BD from Section 4.2.2.

Relationship to BD for Linear Problems

To see that the DWD method described in Section 4.2.1 follows the information flow chart in Figure 4.5 is quite easy. An interesting question at this point is whether the same link applies to the BD methods. According to the presentation above, the MP provides the SP with y_M, the complicating variable. This is in accordance with Figure 4.2 and the description in Section 4.1.2. Furthermore, the SP information passed to the MP is $v_{2,S}$, $v_{3,S}$ and f. These are the dual variables to the constraints not handled in the MP, and in this fashion, one could say that the information flow in the above algorithm is similar to Figure 4.2, but not exactly the same.

In [10], it was shown for a problem similar to the present application that deriving the KKT conditions to the problems in a BD algorithm for optimization problems results in the same procedure as in [16], except for one point; the equation (4.43b), differs from the alternative in [10]. Of course, nothing else can be expected as the original MCP is not eligible to be represented as a single optimization problem. It is however a neat way to derive some parts of the algorithm equations for implementation in GAMS, as it evades some of the rather complicating concepts described above, and it is more intuitively connected to the application formulations.

Convergence for the Energy Market Model

To summarize, there are two requirements that must be met in order to ensure convergence of the BD method. Firstly, to guarantee that a solution is found when $CG^k \geq 0$, the mapping G_2 must either be strictly convex, or $G_2 = \binom{F(q)}{\nabla c(x)}$, with $F(q)$ strictly monotone and $c(x)$ a convex function. The second requirement arises if $CG^k < 0$ for all k. In this case, if any infinite subsequence of $\{(z_M^k, \beta, z_S^{k+1})\}_{k=1}^{\infty}$ has at leas one limit point, and if G_2 is continuous, then

$$lim_{k\to\infty} CG^k = 0.$$

First, $G_2 = \binom{F(q)}{\nabla c(x)}$ is considered. As in equation (4.35), the mapping G_2 is:

$$G_2(v_2, v_3, f) = \begin{pmatrix} -\overline{c} \\ 0 \\ F^{-1}(f) \end{pmatrix} = G_2(v_2, f). \tag{4.45}$$

One can see that G_2 is separable in v_2 and f, and that $-\overline{c}$ is the gradient of the function $c(v_2) = -\overline{c}^T v_2$ which is convex. To verify that $F^{-1}(f)$ is strictly monotone, its inverse, which is the more familiar function $F(q)$, will be studied. In Section 3.5, it is found that $F(q)$ is an affine mapping of the form $F(q) = Cq + b$, and the matrix C is symmetric positive definite. Due to these facts, the inverse of F is also an affine function: $F^{-1}(f) = C^{-1}f - C^{-1}b$, where the matrix C^{-1} is symmetric positive definite. Hence, by the result in equation (2.21), $F^{-1}(f)$ is a strictly monotone function, and the first convergence criterion is satisfied.

Next, for the sequence $\{(z_M^k, \beta^k, z_S^{k+1})\}_{k=1}^{\infty}$, generated through the iterations, it is sufficient to find a limit point for every triple $(z_M^k, \beta^k, z_S^{k+1})$ of this sequence. This verifies that the method will converge, as G is clearly a continuous function (see. equation (4.35)). A limit point of this sequence would exist if the feasible region to the original dual problem, K_D, and the SP, $\overline{K}_D$, both were bounded. Boundedness of $\overline{K}_D$ implies that K_D is also bounded as $\overline{K}_D$ is a relaxation of K_D. Recall that boundedness of $\overline{K}$, which corresponds to $\overline{K}_D$ in this application, is one of the convergence criteria in [13] given in Section (4.2.1).

To investigate the SP, reconsider equation 4.34:

$$\overline{K}_D = \left\{ \begin{array}{r} (v_2, v_3, f) \mid \\ \overline{A}^T v_2 + \hat{A}^T v_3 \leq f \quad (q) \\ \hat{B}^T v_3 \leq d \quad (x) \\ v_2 \geq 0 \end{array} \right\}.$$

The set $\overline{K}_D$ provides the feasible region to the VI problem (4.37):

$$\begin{aligned} &\text{Find } (v_{2,S}^k, v_{3,S}^k, f_S^k) \in \overline{K}_D \text{ s.t.} \\ &(-\overline{c} + \overline{D} y_M^{k-1})^T (v_2 - v_{2,S}^k) \\ &+ F^{-1}(f)^T (f - f_S^k) \geq 0, \quad \forall (v_2, v_3, f) \in \overline{K}_D. \end{aligned} \tag{4.46}$$

According to Section 2.4, the problem in (4.46) can be interpreted as the following minimization problem, when f^* is a known solution to the variable f in (4.46).

$$
\begin{aligned}
\min_{v_2,v_3,f} \quad & (-\overline{c} + \overline{D}y_M^{k-1})^T v_2 + F^{-1}(f^*)^T f \\
\text{st.} \quad & \overline{A}^T v_2 + \hat{A}^T v_3 \leq f \quad (q) \\
& \hat{B}^T v_3 \leq d \quad (x) \\
& v_2 \geq 0
\end{aligned} \tag{4.47}
$$

By investigating the equations in $\overline{K}_D$ or the problem in (4.47), one can see that the constraints are not sufficient to provide a bounded set. For instance, the variable v_2 is the only one with a lower bound.

Therefore, it cannot be guaranteed that CG^k in the BD algorithm approach 0 for the considered energy market model. As a natural consequence, it would be desirable to add some limitation in number of iterations to the algorithm description on page 56. More interestingly, the artificial variables to the MP and artificial bounds for the SP suggested in [13] provide any infinite subsequence of $\{(z_M^k, \beta^k, z_S^{k+1})\}_{k=1}^{\infty}$ with a limit point. This also ensures boundedness of $\overline{K}$, which is a requirement to ensure feasibility of the dual SP. In this fashion it can be assured that $\lim_{k\to\infty} CG^k = 0$, when the suggested additions also are implemented.

Chapter 5

Numerical Experiments with Decomposition

In this chapter, the performance of the previously described algorithm is addressed through the results of numerical experiments. Not surprisingly, the performance depends on the characteristics of the solved problems, and in this regard, the test examples are tailored to present result affected by of variations in these characteristics.

In the first section below, details of the implementation is described, together with some workarounds for an arising issue with the stopping condition. Next, two examples are given, one of a small model where the number of scenarios is increased, and one that is more realistic in its application. By the first example, running times and computational technicalities are addressed, while the second example emphasizes the connection between model features and algorithm performance. Results of the model variables for the examples are not included in this section, as the algorithm performance is the most important here, and because the representation of such solutions requires an extensive amount of details. The chapter ends with a discussion of the results obtained for the two examples.

5.1 Implementation and Computational Technicalities

As before, the PATH solver in GAMS was used to solve the partial MCP problems in the algorithm. The computer used in all tests has the following specifications; Intel Processor of 1.80 GHz and 4.00 GB RAM.

The algorithm is designed to obtain an SP that is separable with respect to scenarios, a structure that is well suited for parallel processing. GAMS allows for grid computing, but when tested for BD in [9], this did not provide any speed-up due to time consuming file- I/O. Hence, this was not done for the current application. In stead, the individual SPs are solved sequentially. For practical purposes, the iteration counter was in this implementation initialized with the value 1 rather than 0 as described in the previous chapter. As for the previous test experiment, the implemented GAMS codes are available at GitHub [25], together with a selection of input datasets.

Initial testing revealed that the algorithm did not always stop before a stopping criterion concerning maximum number of iterations was met. Still, the algorithm seemed to find a solution that was in compliance with the solution obtained without decomposition. Other tests stopped within a few iterations, before the solution was obtained. This indicates that the implementation of the previously defined CG^k, with problem specific details given in Appendix B, equation (B.17), was insufficient to use as stopping metric. More specifically, the convergence gap appeared to have varying values, both negative and positive, before the solution was found, explaining why the algorithm stopped too early. If it did not stop early, it was discovered that the value of CG^k at the later iterations did approach zero. However, at some point before the precision requirement, initially $\epsilon = 0.0001$, was met, the decline in CG^k stopped, and for the remaining of iterations the value of CG^k simply stayed small and constant.

The rapid changes in sign of CG^k might be due to some implementation error, or because direct application of the gap function found in [16] among others is insufficient for the present problem test. To provide a more functional stopping condition, the absolute value of CG^k was introduced in stead, and it was required that this value should be smaller than 0.001, a threshold value ten times larger than the previously defined ϵ. Further testing for a problem with 2380 variables revealed that this stopping condition made the iterations stop at a solution that deviated from the 'direct solution' with no more than the threshold. This deviation represents an error of 0.0056% of the largest value in the MP decision variables and 0.029% of the smallest value of MP decision variables. Such deviations should be acceptable for problem with this level of complexity. However, when not imposing the stopping condition and letting the algorithm proceed for several more iterations, no better value of CG^k was discovered. Hence it can be concluded that the algorithm is not capable of providing solutions of unlimited accuracy. This is reasonable as both the convergence gap and the Benders Cuts added to every MP contains several elements that could cause numerical instabilities, due to the many subtractions and multiplications with small numbers and numbers varying in size.

The use of absolute value of CG^k in the stopping criterion could still be functional if $\lim_{k\to\infty} CG^k = 0$. According to the discussion in Section 4.3.2, this property can be guaranteed if some additions are made to the implementation. No such additions were made to the present implementation as no further issues concerning convergence were observed. Instead, an alternative stopping condition inspired by [9], who also describes issues with the previously defined CG^k, was implemented. The additional condition allows the iterations to stop when the largest absolute difference in each of the expansion variables from one iteration to the next was sufficiently small. This means

$$\left.\begin{array}{l} |f^{Ik}_{m,n,n',e} - f^{I(k-1)}_{m,n,n',e}| < \epsilon_2 \\ |x^{Ik}_{m,n,e} - x^{I(k-1)}_{m,n,e}| < \epsilon_2 \end{array}\right\} \; \forall m \in M, n, n' \in N, e \in E. \tag{5.1}$$

The value of ϵ_2 was set to be ten times smaller than ϵ used for the modified original criterion. It can be argued that if the MP solution will not change much from one step to another, nor will the SP, and in this situation, no further changes will take place. This justifies the use of the conditions in (5.1). Alternatively, one could have used the variations in the variable λ^k to determine whether the variations in the MP solutions form one iteration to the next are sufficiently small.

The introduced stopping condition is not problem specific. It should work for the complicating variables solved in the MP of any BD algorithm for VI problems. Furthermore, this is probably a practical condition as the number of complicating variables preferably is small relative to the total problem size. Hence this condition is also relatively easy to compute in comparison to CG^k.

It was found that no changes within the displayed output of four digits precision occurred after the conditions in (5.1) were satisfied. In other words, the new stopping condition worked well. Furthermore, it was discovered that the two convergence criteria were satisfied in the same iteration in the wast majority of test examples. In the opposite case, it was sufficient to make at most two more iteration before both conditions were satisfied.

Based on the above reasoning and observations, it can be assumed that the additional criterion and the absolute value of CG^k can be used as convergence metrics. In fact, the latter of the two conditions provides an alternative that is easier to compute and possibly not subject to the same occurrence of numerical instabilities. Hence this alternative may contribute to a reduction in running times when used alone as stopping criterion. However, a short test for the 2380 variable test experiment showed that the times it take to compute the original and the new metric constitutes 0.016% and 0.0058% of the total running time measured in wall time. This shows that it is more than twice as time consuming to compute the original CG^k, but none of the computations are of significance for the total running time. Therefore, no further investigations were made concerning time consumption in the computation of convergence metrics, and both the conditions were used throughout the tests preformed.

Figure 5.1 below shows the development of the two metrics throughout the iterations for the 2380-variable test example. For this particular case the new convergence criterion was met after 12 iterations, and the old was met after 14. The figure also illustrates how neither of the metrics are monotonically decreasing. According to the code on page 56, no convergence check is carried out in the first iteration, and therefore no such metric is plotted at iteration 1.

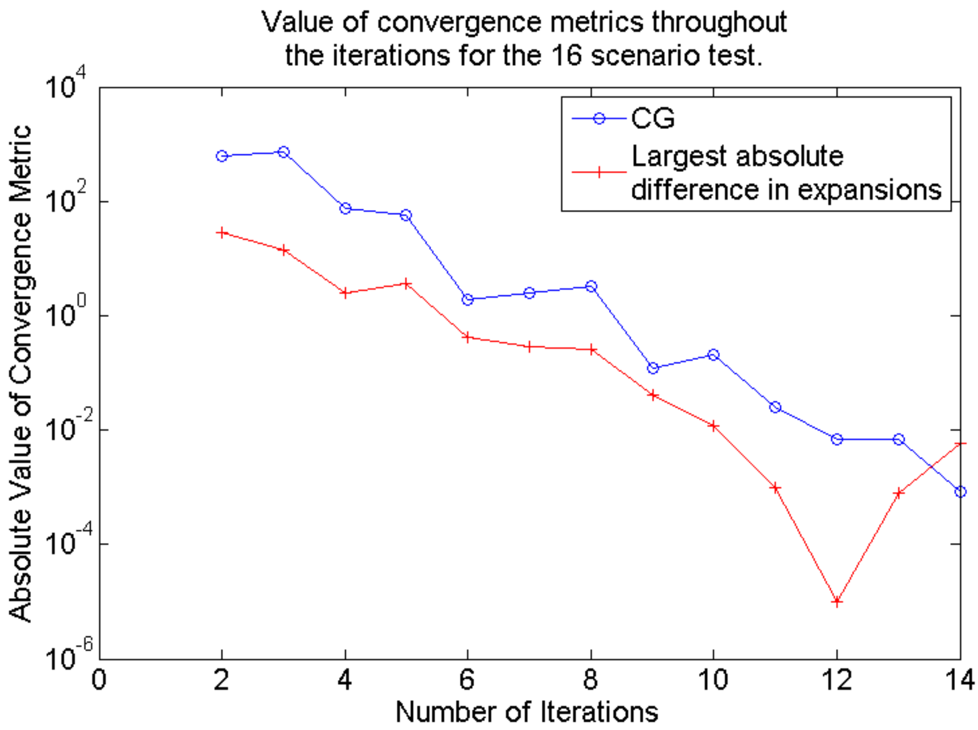


Figure 5.1: Value of convergence metrics throughout the iterations of a test example.

5.2 Test Examples and Results

5.2.1 Two-stage Problem with Increasing Number of Scenarios

The scenario-wise separation in the SP is the feature that the algorithm is expected to gain the most speed form. Hence, it seems reasonable that a test example with many independent scenarios is among the types of problems with the best possibility to reduce running times relative to a non-decomposed solution alternative. More precisely, with a large number of independent scenarios, the MP does not need to suggest many changes before a global equilibrium is obtained, because there are not that many variables to suggest adjustments for.

To investigate these assumptions, a test case similar to the one in Section 3.7.1 was considered. First a problem with two time stages and two scenarios was constructed. As before, the demand was the only uncertain factor in stage 2, but it was now varying between a 50% reduction and a 5 times increase in the values of $int_{m,n,d,e}$ relative to stage 1. The probabilities of each scenario were also changed to be 0.5 for each case. From this basic example, new problems were generated by doubling the number of scenarios, and letting the probability of each scenario follow a normal distribution with expected value 2.75. In this way, the number of data-points describing the uncertain elements were increased. The set-up is illustrated in Figure 5.2, and inspired by a similar test found in [10], which allows for comparison of the results.

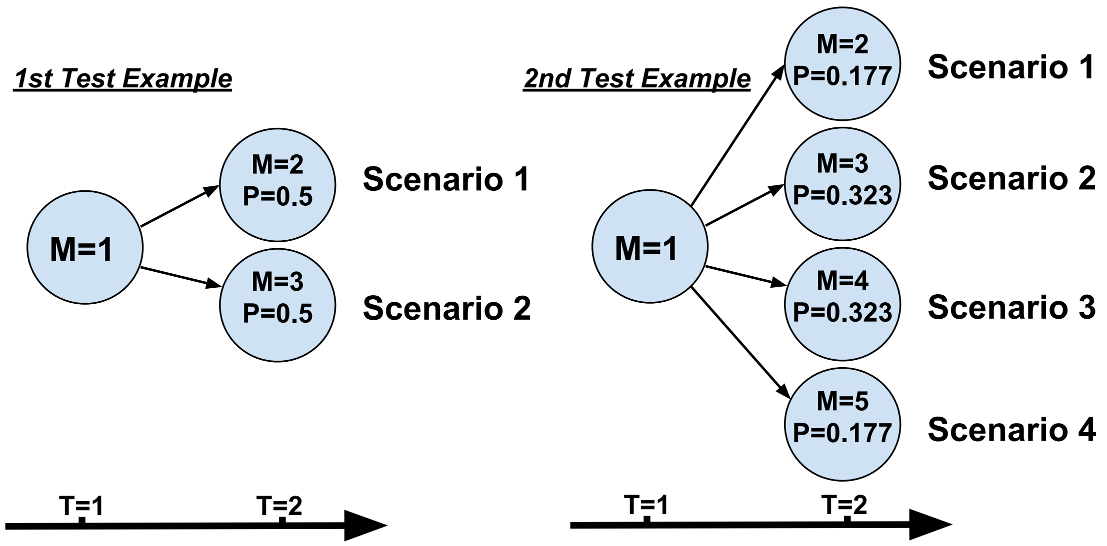


Figure 5.2: Illustration of how larger test examples were generated.

The first round of testing revealed that the running times are highly dependent on the number of binding constraints at the solution that involves complicating variables only. It was observed that the first iterations of the algorithm suggest solutions that are either bounded by these constraints, or the non-negativity constraints. In this regard it is reasonable to believe that the algorithm can finish in a fewer number of iterations when the upper bounds on transportation and transformation capacities are active, or when the solution not suggests any expansions at all.

To obtain a test problem that is slightly more interesting when discussing algorithm performance, some other input parameters (initial flow and transformation capacities and costs) were also changed from the Section 3.7.1 example, until the none of the investments were bounded by constraints in any of the solutions. At this point, the meaning of the model and its results were not stressed. Nevertheless, the correctness of the solutions were verified with the solutions obtained without decomposition for the smallest problems (number of scenarios in the range 2 to 128). The largest problem instances (256 and 512 scenarios) were not solvable without use of the decomposition algorithm within a reasonable amount of time.

The table below (Table 5.1) shows details of the input such as number of scenarios and total number of variables in the original problem formulation. The performance results listed in the same table include wall time measured for the total execution of the entire algorithm, all MPs and all SPs. Execution time is defined as the time it takes to generate the problem, solve it, write output to screen and generate an report file, while solution time is simply the time it takes the solver to find a solution only. For comparison with the results shown in [9], the number of iterations for each test and total CPU times (obtained using the .resUsd function in GAMS) spent to solve the MPs and SPs are given as well. In Figure 5.3 the running times are also plotted for increasing number of scenarios for all tests. Figure 5.4 is the last presented plot of the 2-stage problem results. This plot shows the increase in total execution time (wall time), and total time (both wall time and CPU time) spent solving the SPs for each test of the decomposition algorithm.

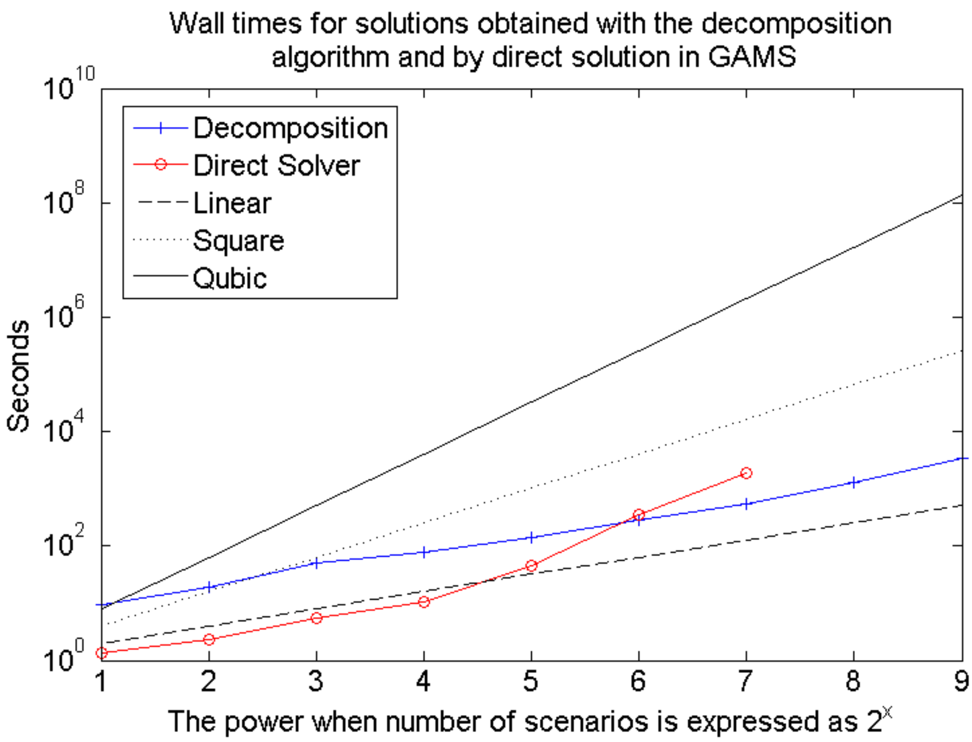


Figure 5.3: Wall times for all tests.

Table 5.1: Running time results for the 2-stage problem with increasing number of scenarios.

	All times in seconds.								
Number of scenarios	2	4	8	16	32	64	128	256	512
Number of variables	420	700	1260	2380	4620	9100	18060	35980	71820
Number of expansion variables	4	4	4	4	4	4 4	4	4	
Total number of SP variables	348	580	1044	1972	3828	7540	14964	29812	59508
Number of iterations	5	7	12	12	11	11	11	13	14
Total running time (wall time)	9.275	19.134	51.221	76.229	137.527	277.475	535.202	1321.184	3421.152
SP running time (wall time)	7.840	16.985	46.664	71.608	132.587	271.210	526.171	1302.801	3380.021
SP solving time (wall time)	6.840	15.705	42.573	64.516	114.423	239.897	453.962	1074.599	2271.738
SP solving time (CPU time)	3.421	7.017	16.589	16.262	34.584	61.341	95.989	218.140	546.786
MP running time (wall time)	1.423	2.125	4.503	4.504	4.723	5.819	8.075	15.838	34.412
MP solving time (CPU time)	0.358	0.502	0.877	0.657	0.766	0.671	0.751	0.844	1.265
Running time without decomposition (wall time)	1.412	3.975	5.625	10.991	43.365	386.315	1889.877	-	-
Running time without decomposition (CPU time)	1.109	2.203	5.313	10.640	42.969	385.781	1888.641	-	-

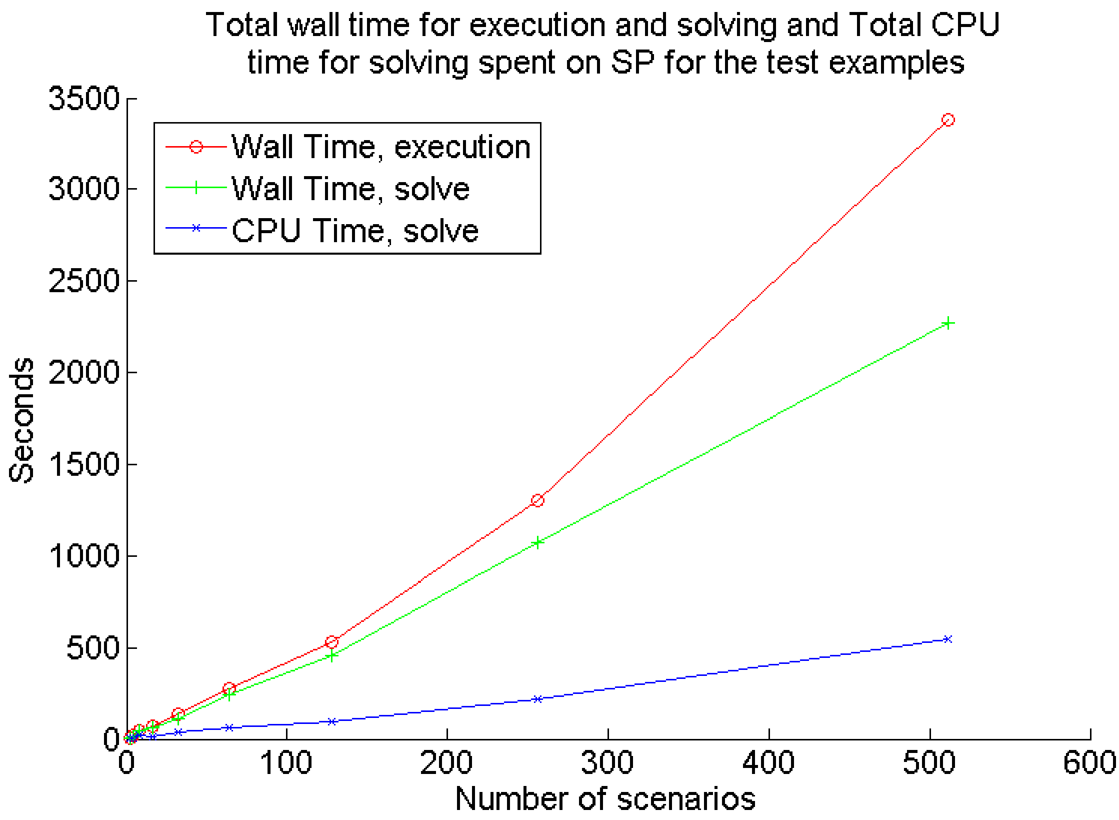


Figure 5.4: Total time spent solving SPs in the decomposition algorithm, measured in both CPU time and wall time, and total execution time spent on SPs measured in wall time.

5.2.2 Three-stage Case for North-western Europe

To investigate the performance for a more realistic case, an example of trade with natural gas and electricity in Norway, France, the UK, Belgium, Netherlands, and Germany was investigated over a three stage period, including the years 2004, 2008 and 2012. Each of the countries was modelled as one node in each, and each country was considered as a single produced/consumer. Indicative input data such as production, transformation and consumption of natural gas and electricity and heat were inspired by statistics from the International Energy Agency [5]. For transportation networks, data was found at ENTSOE (European Network of Transmission System Operators for Electricity) [4] and ENTSOG (European Network of Transmission System Operators for Gas) [2]. As the aim of this example is to show performance, the details in the input data were slightly simplified for easy handling.

According to statistics found in a survey provided by BP [1], the production of natural gas in the UK has declined with more than 50% from 2004 to 2012, due to depletion in reservoirs. Based on this fact, a scenario tree was designed to include uncertainties related to the production in the UK and the Netherlands (NL) in 2008. More specifically, three cases for 2008 were featured;

(a) Decline in UK production, NL production constant.

(b) UK production constant, small decline in NL production.

(c) Extensive decline in UK production, small increase in NL production.

Furthermore, the fact that there in 2004 were no pipelines for gas transportation from Norway (NOR) to the UK and from NL to the UK, was included in the network, by allowing for expansions on the arcs connecting these countries, but with no initial transportation capacity in 2004.

In the third stage, in 2012, the scenario tree covers uncertainties in German (GER) demands for electricity. Two cases were considered:

(i) A 20% reduction in the parameter *int*.

(ii) A 20% increase in the parameter *int*.

In total the considered uncertainties over three time stages gives a scenario tree with 10 nodes. An illustration is given in Figure 5.5.

With six countries included, the network structure describing the market has six nodes. Norway and the UK are the only producers of both natural gas and electricity. France (FRA), Belgium (BEL) and Germany produces electricity only, while the Netherlands has production of natural gas only. The network is shown in Figure 5.6 where the modelled arcs are indicated. The last feature included is the facility to produce electrical power with natural gas as input fuel and electricity as output. These facilities are located in the UK and Germany, as these are the countries where the largest volumes of such transformations take place according to [5].

Adding all the features explained and showed in the Figures 5.5 and 5.6, the MCP problem deriving from the model consists of 13 080 variables, out of which 24 are complicating. Even at this size (measured in number of variables), the model lacks from the somewhat simplified input data and the exclusion of other elements of significance such as energy from coal and oil, as well as more trading companies rather than single counties. Nevertheless, the results were reasonable in comparison to the statistics. For example, the building of pipelines from Norway and the Netherlands both ending in the UK, was among the results.

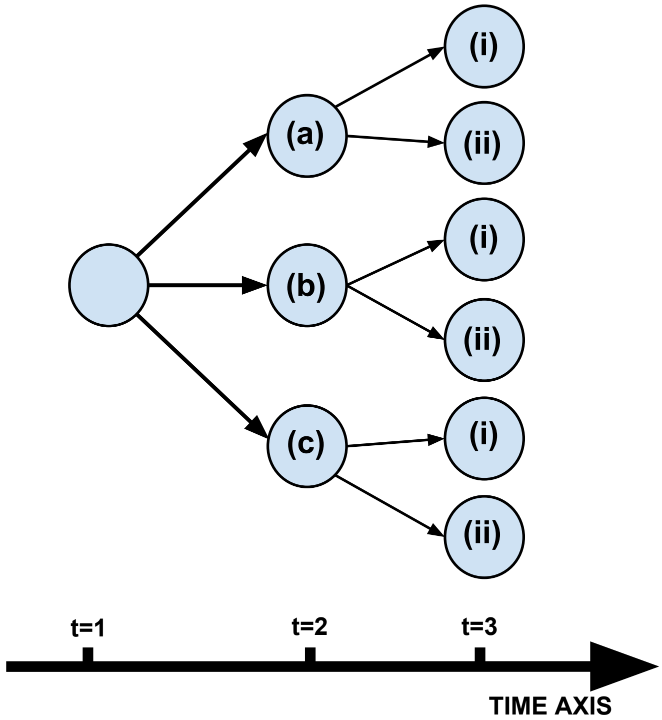


Figure 5.5: Illustration of the scenario tree.

The solution was obtained both by applying the MCP solver in GAMS directly, and by use of the implemented decomposition algorithm. In this way it could be verified that the two approaches gave the same results for the variables. The performance results are shown in the table below. In addition it should be noted that the solution did not indicate any other expansions than the ones for natural gas from Norway and the Netherlands to the UK. This simplifies the solution procedure for both the decomposition algorithm and the direct application of the PATH solver, in contrast to the previous 2-stage experiment, where such an situation was evaded on purpose.

5.3 Discussion of Results

5.3.1 The 2-stage Problem

The results in Table 5.1 and the plot in Figure 5.3 show that the implemented algorithm in most cases is faster, in terms of both wall time and CPU time, than the alternative with direct application of the PATH solver for MCPs in GAMS. For the largest tested problems, the decomposition algorithm was also the only solver

Table 5.2: Running time results for the 3-stage problem .

	All times are in seconds.
Total number of variables	13 080
Total number of SP variables	11 400
Number of iterations	7
Total running time (wall time)	126.984
SP running time (wall time)	123.814
SP running time (CPU time)	102.097
MP running time (wall time)	3.002
MP running time (CPU time)	0.518
Running time without decomposition (wall time)	268.640
Running time without decomposition (CPU time)	268.172

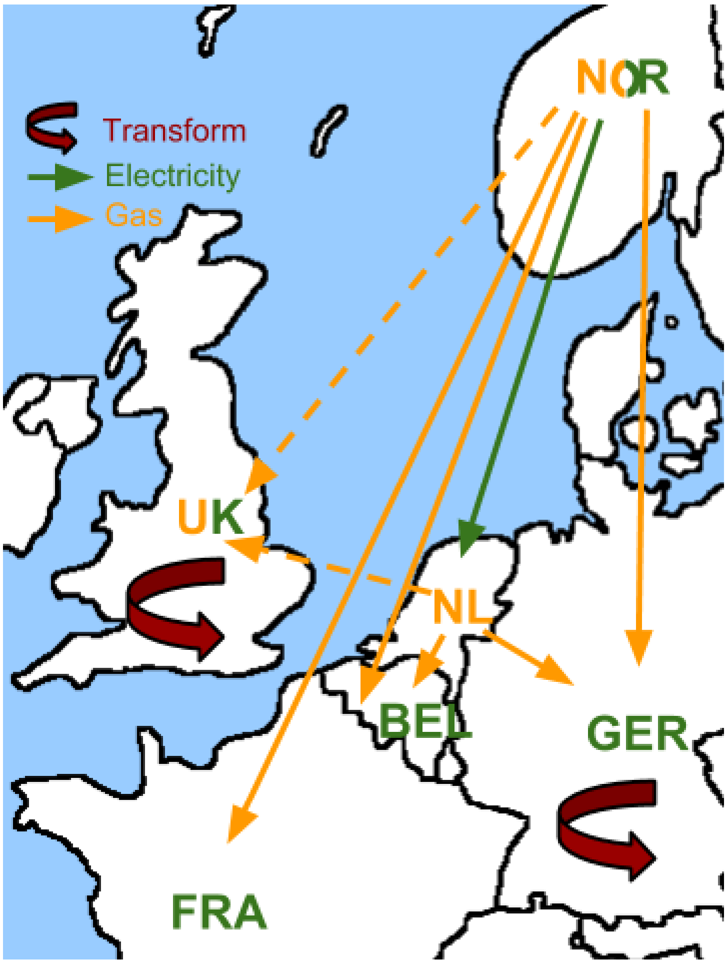


Figure 5.6: Illustration of the network of trade with natural gas and electricity in North-Western Europe. The colors on the country names (abbreviated) indicate the production of electricity, natural gas or both.

option that actually could provide a solution within a reasonable amount of time. However, and not unexpectedly, the smallest test problems where the number of scenarios are in the range $2 - 32$, solved the fastest without utilization of the implemented algorithm. All problems have the same ratio of number of variables in the full size problem relative to the total number of variables in the SP, and thus, the observation cannot be explained by this ratio. It must be the PATH solver that is sufficiently effective for the smallest problems, making it disadvantageous to run multiple iterations with the decomposition algorithm.

When the problem size increases, the PATH execution time grows, as one can see from the red lines in Figure 5.3. For smaller problems (2-16 scenarios) the growth seems to be linear, but for the larger tests that were finished (16-128 scenarios) the growth in running times shifts to a cubic tendency. This fact is what makes the decomposition algorithm beneficial for large problems. As the number of scenarios gets larger, the SP element corresponding to a single scenario tree node is still constant in size. Hence, the SP solution time should grow according to a linear pattern relatively to the number of scenarios in the problem. This pattern, or rather patterns, are shown in Figure 5.4, verifying that the growth when measured both in wall time and CPU time is at least close to linear for the SPs. The blue line in Figure 5.3 shows that the total running times, in which SPs is the most demanding task, also is close to linear. Irregularities in this blue line can be explained by the variations in growth of number of iterations carried out for each test case.

Yet, the plot in Figure 5.4 does not show perfect linearity in total execution time of each SP. A reason for this may be that the time spent reading/writing to file and generating the problem grows more rapidly. The increasing gap between the execution time and solution time (both in wall time) verifies this. Inspired by this observation, it was discovered that the largest output file produced by GAMS for the 512-scenario problem contained 230 MB of data, which is large enough to affect running time. It should be noted that there are provided options in GAMS for reducing the size and amounts of contents in an output file. As mentioned, the time it takes to generate every SP may also have a growth rate more rapid than linear, but without more options to measure this, nothing can be concluded.

The difference in total execution time and solution time for the SPs is, as shown in Figure 5.4 not that big, despite the rapid growth in non-solution related tasks. In the most extreme case, with 512 scenarios, the solution time measured in wall time constitutes 66% of the total SP execution time, and this is clearly more than for any other test. Still, if the algorithm was solved in parallel, the solution time for the SP would decrease substantially, letting the other aforementioned factors represent a much larger share of the execution times. Hence, the possibilities found in GAMS to reduce the volume of outputted files would have been of interest to investigate more. The problem generation at every iteration cannot be evaded when using GAMS, so to optimize this part of the code, one would have to use another programming language that facilitates the handling of each SP as a function. Based on the experiments with a parallel GAMS implementation in [9], the use of another language may seem to be a necessity to obtain efficient grid computing in any way.

The experiment set-up for the 2-stage problem was inspired by a similar presentation of results for a global natural gas market model found in [10], to allow for comparison. In comparison to the present experiment, the problems in [10] are significantly larger, with 7313 variables for 2 scenarios, and 620 215 for the largest test problem consisting of 256 scenarios. Among the results in [10], 'Net calculation time' is measured as the sum of total MP and SP CPU times, and these are comparable with the sum of the corresponding MP and SP CPU times found in Table 5.1. When comparing the two experiments, the difference in running times is largest for the problems with the most scenarios. This seems reasonable when considering the previous findings of a slight tendency of a growth rate more rapid than linear for larger problems. Hence it can be concluded that the present test experiments have solutions that are identifiable with what is found in [10]. The relatively large differences in number of variables for each instance of the tests should, however, be taken into account, precluding any further analysis.

5.3.2 The North-western European Problem

The results in Table 5.2 shows that the decomposition algorithm was capable of detecting a solution in less than 50% of the wall time spent by the direct option. This shows that the decomposition algorithm is a good alternative for models and input data like the tested example, and possibly also for even more complex models and for a higher level of details in the input dataset. As the problem in total consisted of 13 080 variables, it is interesting to compare the results with the results in the 2-stage cases with 64 and 128 scenarios, consisting of 9 100 and 18 060 variables respectively (see Table 5.1). It appears that the running times for the three stage example are significantly less than the same results for the 64-scenario, 2-stage example. This is for both the decomposition algorithm (126.984 vs. 277.475 seconds) and the full MCP option (268.640 vs. 386.315 seconds). These differences can be explained by the fact that only two of the complicating variables were non-zero in the solution, and that the number of scenarios is much smaller for the three stage example (10 vs. 64). The differences in running times are not surprising, but rather illustrative in terms of showing the level of variations that may occur in many similar energy market models.

Chapter 6

Conclusions

In this thesis, a stochastic energy market equilibrium model has been developed. With the aim of obtaining a fast solution procedure to this model, a Benders Decomposition algorithm was designed and implemented in GAMS. Testing showed that the implemented algorithm, after some adjustments, provided solutions in compliance with the solutions obtained by direct application of the MCP solver in GAMS.

Due to issues with the intentional implemented convergence criterion found in the literature, an alternative, and easier convergence criterion was proposed and successfully tested for the present application. This convergence metric was inspired by a similar approach in [9] and is a substantial simplification of the convergence metric found in [13], [14] and [16]. Furthermore, this alternative stopping condition should be applicable to any Benders Decomposition algorithm for MCPs or VI problems.

For the tested examples, the developed decomposition algorithm provided a good and efficient solution method in comparison to the alternative of using the PATH solver in GAMS directly. It was also found that the results were in compliance with an experiment found in [10]. The algorithm proved to be the most efficient relative to the non-decomposition option for the test cases that were the largest in terms of input data and number of variables. Also for the most realistic test experiment, the developed algorithm was the fastest to obtain a solution. This is probably the most promising of the results obtained though testing. As the example represents a minimum of details in input data to be in accordance with reality, it can be concluded that the decomposition algorithm is likely to be ideal for problems of a larger scale as well.

Among other important findings in this thesis, the results verify that the algorithm has a good potential to gain speed-up when customized for parallel computing.

Future Work

A natural continuation of the work presented in this thesis would be to proceed in studying possibilities to reduce running time, either for the suggested model or a similar problem. In this regard, the following list contains good starting points for further research.

- Study alternative formulations of the model equations that can reduce the number of variables. For instance, one could let transportation arcs be defined as a distinct entity in stead of defining all arcs by their start and end node, as done in [10].

- Make use of the grid computing facilities in GAMS to study how this works for the current application. The experience of no speed-up for smaller problems in [9] does not provide sufficient information to conclude that this will not prove to be an efficient alternative for the present application.

- Explore the possibilities of reducing the outputted report files after each execution of the PATH solver. This may reduce both running time and demand for memory.

- In general, and especially in light of the fact that the algorithm seemed to have limited capabilities of providing solutions of a high level of accuracy, it would be of interest to carry out a thorough stability analysis. No such thing is mentioned in any of the main sources to this work: [13], [14] and [16].

- Of course, when seeking fast and efficient solution strategies, the use of a fast and efficient programming language such as C, is an option that is likely to provide a successful result. However, this shift would entail a much harder implementation of the application, and it would require that some other suitable solver was found and utilized for the MCPs or VI problems arising in the studied Benders Decomposition algorithm.

Bibliography

[1] Bp statisitical review of world energy june 2013. http://www.bp.com/content/dam/bp/pdf/statistical-review/statistical_review_of_world_energy_2013.pdf.

[2] Entsog capacity map dataset in excel format. http://www.entsog.eu/maps/transmission-capacity-map.

[3] The path solver. http://pages.cs.wisc.edu/ ferris/path.html.

[4] Production data. https://www.entsoe.eu/data/data-portal/production/.

[5] Statistics. http://www.iea.org/statistics/statisticssearch/.

[6] J.F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, pages 238–252, 1962.

[7] Castillo E. Minguez R. Conejo, A.J. and R. Garcia-Bertrand. *Decomposition Techniques in Mathematical Programming*. Springer, 2006.

[8] George B. Dantzig and Phillip Wolfe. Decomposition principle for linear programs. *The Annals of Mathematics*, (1):101–111, 1960.

[9] R.G. Egging. *Multi-Period Natural Gas Market Modelling-Applications, Stochastic Extensions and Solution Approaches*. PhD thesis, 2010.

[10] R.G. Egging. Benders decomposition for multi-stage stochastic mixed complementarity problems - applied to a global natural gas market model. *European Journal of Operational Research*, 226(2):341–353, 2013.

[11] R.G. Egging and D. Huppmann. Investigating a co2 tax and a nuclear phase out with a multi-fuel market equilibrium model. In *9th International Conference on the European Energy Market (EEM)*. IEEE, 2013.

[12] F. Facchinei and J. Pang. *Finite-Dimensional Variational Inequalities and Complementarity Problems, vol. 1*. Springer, 2003.

[13] J.D. Fuller and W. Chung. Dantzig-wolfe decomposition of variational inequalities. *Comutational Economics*, 25(4):303–326, 2005.

[14] J.D. Fuller and W. Chung. Benders decomposition for a class of variational inequalities. *European Journal of Operational Research*, 185(1):76–91, 2007.

[15] Conejo A.J. Fuller J.D. Hobbs B.F. Ruiz C. Gabriel, S.A. *Complementarity Modelling in Energy Markets.* Springer, New York, 2013.

[16] S. A Gabriel and J.D. Fuller. A benders decomposition method for solving stochastic complementarity problems with an application in energy. *Comutational Economics*, 35(4):301–329, 2010.

[17] A.M. Geoffrion. Generalized benders decomposition. *Journal of Optimization, Theory and Applications*, pages 237–260, 1972.

[18] C. J Goh and Xiaoqi Yang. *Duality in optimization and variational inequalities.* Taylor and Francis, 2002.

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

[20] Siriphong Lawphongpanich and D. W. Hearn. Benders decomposition for variational inequalities. *Mathematical Programming*, (1-3):231–247, 1990.

[21] Rönnquist M. Värbrand P. Lundgren, J. *Optimization.* Studentlitteratur, Malmö, 2010.

[22] John Nash. Non-cooperative games. *The Annals of Mathematics*, (2):286–295, 1951.

[23] J. Nocedal and S. Wright. *Numerical Optimization, Second edition.* Springer, 2006.

[24] R. E. Rosenthal. Gams - a user's guide. http://www.gams.com/dd/docs/bigdocs/GAMSUsersGuide.pdf, 2014.

[25] Emily Siggerud. Gams codes developed as part of thesis. https://github.com/emilysig/master/.

[26] James Stewart. *Calculus: Applications and Technology.* Thomson Books, 2005.

Appendix A

KKT Conditions

In this appendix, the KKT conditions for the energy market model described in Chapter 3 are presented.

A.1 KKT Conditions for the Producers

$$\begin{aligned}
0 \leq q^P_{m,n,p,e} \quad &\perp \quad prob_m \cdot disc_m \cdot (2 \cdot k_1 \cdot q^P_{m,n,p,e} + k_2) \\
&\quad + \epsilon_{m,n,p,e} + \zeta_{m,n,p,e} \geq 0 && \text{(A.1)} \\
0 \leq q^S_{m,n,p,d,e} \quad &\perp \quad -\, prob_m \cdot disc_m \cdot \Big(int_{m,n,d,e} - slp_{m,n,d,e} \cdot q^S_{m,n,p,d,e} \\
&\quad - slp_{m,n,d,e} \cdot \sum_{p' \in P(n)} q^S_{m,n,p',d,e} \Big) - \zeta_{m,n,p,e} \geq 0 && \text{(A.2)} \\
0 \leq q^T_{m,n,n',p,e} \quad &\perp \quad prob_m \cdot disc_m \cdot \upsilon_{m,n,n',e} - \zeta_{m,n,p,e} + \zeta_{m,n',p,e} \geq 0 && \text{(A.3)} \\
0 \leq q^C_{m,n,p,e,e'} \quad &\perp \quad prob_m \cdot disc_m \cdot \phi_{m,n,e,e'} \\
&\quad - \zeta_{m,n,p,e} + l_{m,n,e',e} \cdot \zeta_{m,n,p,e'} \geq 0 && \text{(A.4)} \\
0 \leq \epsilon_{m,n,p,e} \quad &\perp \quad \overline{q^P}_{m,n,p,e} - q^P_{m,n,p,e} \geq 0 && \text{(A.5)} \\
\zeta_{m,n,p,e} \;\; free, \quad &\quad\;\; q^P_{m,n,p,e} + \sum_{n' \in N(p)} q^T_{m,n',n,p,e} + \sum_{e' \in E} l_{m,n,e',e} \cdot q^C_{m,n,p,e',e} \\
&\quad - \sum_{d \in D} q^S_{m,n,p,d,e} - \sum_{n' \in N(p)} q^T_{m,n,n',p,e} - \sum_{e' \in E} q^C_{m,n,p,e,e'} = 0 && \text{(A.6)}
\end{aligned}$$

A.2 KKT Conditions for the Transporter

This section contains the KKT conditions to the optimization problems described in Chapter 3.

$$0 \leq f_{m,n,n',e} \quad \perp \quad prob_m \cdot disc_m \cdot (2 \cdot k_3 \cdot f_{m,n,n',e} - \upsilon_{m,n,n',e}) + \iota_{m,n,n',e} \geq 0 \tag{A.7}$$

$$0 \leq f^I_{m,n,n',e} \quad \perp \quad prob_m \cdot disc_m \cdot k_4 + \kappa_{m,n,n',e} - \sum_{m' \in S^O(m)} \iota_{m',n,n',e} \geq 0 \tag{A.8}$$

$$0 \leq \iota_{m,n,n',e} \quad \perp \quad \overline{f}_{m,n,n',e} + \sum_{m' \in S^A(m)} f^I_{m',n,n',e} - f_{m,n,n',e} \geq 0 \tag{A.9}$$

$$0 \leq \kappa_{m,n,n',e} \quad \perp \quad \overline{f^I}_{m,n,n',e} - \quad f^I_{m,n,n',e} \geq 0 \tag{A.10}$$

A.3 KKT Conditions for the Transformer

$$0 \leq x_{m,n,e',e} \quad \perp \quad prob_m \cdot disc_m \cdot (2 \cdot k_5 \cdot x_{m,n,e',e} - \phi_{m,n,e',e}) + l_{m,n,e',e} \cdot \nu_{m,n,e} \geq 0 \tag{A.11}$$

$$0 \leq x^I_{m,n,e} \quad \perp \quad prob_m \cdot disc_m \cdot k_6 + \xi_{m,n,e} - \sum_{m' \in S^O(m)} \nu_{m',n,e} \geq 0 \tag{A.12}$$

$$0 \leq \nu_{m,n,e} \quad \perp \quad \overline{x}_{m,n,e} + \sum_{m' \in S^A(m)} x^I_{m',n,e} - \sum_{e \in E} l_{m,n,e',e} \cdot x_{m,n,e',e} \geq 0 \tag{A.13}$$

$$0 \leq \xi_{m,n,e} \quad \perp \quad \overline{x^I}_{m,n,e} - x^I_{m,n,e} \geq 0 \tag{A.14}$$

A.4 KKT Conditions for the Market Clearing Equations

$$\upsilon_{m,n,n',e} \ \ free, \quad f_{m,n,n',e} - \sum_{p \in P(n)} q^T_{m,n,n',p,e} = 0 \tag{A.15}$$

$$\phi_{m,n,e,e'} \ \ free, \quad x_{m,n,e,e'} - \sum_{p \in P(n)} q^C_{m,n,p,e,e'} = 0 \tag{A.16}$$

Appendix B

Application of BD in Full Notation

This appendix shows the subproblem, master problem and convergence gap in full notation for the energy market model. For readability, the iteration counter is not included in the notation.

B.1 Subproblem

The SP is presented here according to equations (4.42), with $f^{Ik}_{m',n,n',e}$ and $x^{Ik}_{m',n,e}$ denoting the fixed values suggested by the previous MP.

$$
\begin{aligned}
0 \leq q^P_{m,n,p,e} \quad &\perp \quad prob_m \cdot disc_m \cdot (2 \cdot k_1 \cdot q^P_{m,n,p,e} + k_2) \\
&\qquad + \epsilon_{m,n,p,e} + \zeta_{m,n,p,e} \geq 0 \qquad \text{(B.1)}\\
0 \leq q^S_{m,n,p,d,e} \quad &\perp \quad - prob_m \cdot disc_m \cdot \Big(int_{m,n,d,e} - slp_{m,n,d,e} \cdot q^S_{m,n,p,d,e} \\
&\qquad - slp_{m,n,d,e} \cdot \sum_{p' \in P(n)} q^S_{m,n,p',d,e} \Big) - \zeta_{m,n,p,e} \geq 0 \qquad \text{(B.2)}\\
0 \leq q^T_{m,n,n',p,e} \quad &\perp \quad prob_m \cdot disc_m \cdot \upsilon_{m,n,n',e} - \zeta_{m,n,p,e} + \zeta_{m,n',p,e} \geq 0 \qquad \text{(B.3)}\\
0 \leq q^C_{m,n,p,e,e'} \quad &\perp \quad prob_m \cdot disc_m \cdot \phi_{m,n,e,e'} \\
&\qquad - \zeta_{m,n,p,e} + l_{m,n,e',e} \cdot \zeta_{m,n,p,e'} \geq 0 \qquad \text{(B.4)}\\
0 \leq f_{m,n,n',e} \quad &\perp \quad prob_m \cdot disc_m \cdot (2 \cdot k_3 \cdot f_{m,n,n',e} - \upsilon_{m,n,n',e}) + \\
&\qquad \iota_{m,n,n',e} \geq 0 \qquad \text{(B.5)}\\
0 \leq x_{m,n,e',e} \quad &\perp \quad prob_m \cdot disc_m \cdot (2 \cdot k_5 \cdot x_{m,n,e',e} - \phi_{m,n,e',e}) + \\
&\qquad l_{m,n,e',e} \cdot \nu_{m,n,e} \geq 0 \qquad \text{(B.6)}\\
0 \leq \epsilon_{m,n,p,e} \quad &\perp \quad \overline{q^P}_{m,n,p,e} - q^P_{m,n,p,e} \geq 0 \qquad \text{(B.7)}\\
\zeta_{m,n,p,e} \;\; free, \quad &\quad\;\; q^P_{m,n,p,e} + \sum_{n' \in N(p)} q^T_{m,n',n,p,e} + \sum_{e' \in E} l_{m,n,e',e} \cdot q^C_{m,n,p,e',e}
\end{aligned}
$$

$$-\sum_{d\in D} q^S_{m,n,p,d,e} - \sum_{n'\in N(p)} q^T_{m,n,n',p,e} - \sum_{e'\in E} q^C_{m,n,p,e,e'} = 0 \quad \text{(B.8)}$$

$$0 \le \iota_{m,n,n',e} \quad \perp \quad \overline{f}_{m,n,n',e} + \sum_{m'\in S^A(m)} f^{Ik}_{m',n,n',e} - f_{m,n,n',e} \ge 0 \quad \text{(B.9)}$$

$$0 \le \nu_{m,n,e} \quad \perp \quad \overline{x}_{m,n,e} + \sum_{m'\in S^A(m)} x^{Ik}_{m',n,e} - \sum_{e\in E} l_{m,n,e',e} \cdot x_{m,n,e',e} \ge 0 \quad \text{(B.10)}$$

$$\upsilon_{m,n,n',e} \;\; free, \qquad f_{m,n,n',e} - \sum_{p\in P(n)} q^T_{m,n,n',p,e} = 0 \quad \text{(B.11)}$$

$$\phi_{m,n,e,e'} \;\; free, \qquad x_{m,n,e,e'} - \sum_{p\in P(n)} q^C_{m,n,p,e,e'} = 0 \quad \text{(B.12)}$$

B.2 Master Problem

The master problem according to (4.43), is given below in full notation. From the previous SP, the following parametrized variables are extracted: $q^{Sk}_{m,n,p,d,e}$, $q^{Pk}_{m,n,p,e}$, $f^k_{m,n,n',e}$, $x^k_{m,n,e,e'}$, $\epsilon^k_{m,n,p,e}$, $\iota^k_{m,n,n',e}$ and $\nu^k_{m,n,e}$. To simplify the notation, the summation over several sets is abbreviated. That is $\sum_{n,p,d,e,k'}$ means $\sum_{n\in N}\sum_{p\in P(n)}\sum_{d\in D}\sum_{e\in E}\sum_{k'\in\{1,\ldots,k\}}$, and $\sum_{n',e'}$ means $\sum_{n'\in N}\sum_{e'\in E}$ etc.

$$0 \le f^I_{m,n,n',e} \quad \perp \quad prob_m \cdot disc_m \cdot k_4 + \kappa_{m,n,n',e} - \sum_{m'\in S^O(m)} \iota_{m',n,n',e} \ge 0$$

$$0 \le x^I_{m,n,e} \quad \perp \quad prob_m \cdot disc_m \cdot k_6 + \xi_{m,n,e} - \sum_{m'\in S^O(m)} \nu_{m',n,e} \ge 0$$

$$0 \leq \lambda^k \quad \perp \quad \theta + \sum_{m \in M} \left[\begin{array}{ll} & \sum_{n,p,e} \epsilon^k_{m,n,p,e} \big(\overline{q^P}_{m,n,p,e} \big) \\ + & \sum_{n,n',e} \iota^k_{m,n,n',e} \big(\overline{f}_{m,n,n',e} + \sum_{m' \in S^A(m)} f^I_{m',n,n',e} \big) \\ + & \sum_{n,e} \nu^k_{m,n,e} \big(\overline{x}_{m,n,e} + \sum_{m' \in S^A(m)} x^I_{m',n,e} \big) \\ - & prob_m \cdot disc_m \sum_{n,p,d,e,k'} \Big(int_{m,n,d,e} - slp_{m,n,d,e} \cdot \\ & \big(q^{Sk}_{m,n,p,d,e} + \sum_{p' \in P(n)} q^{Sk}_{m,n,p',d,e} \big) \Big) \lambda^{k'} q^{Sk'}_{m,n,p,d,e} \\ + & prob_m \cdot disc_m \sum_{n,p,e,k'} (2 \cdot k_1 \cdot q^{Pk}_{m,n,p,e} + k_2) \lambda^{k'} q^{Pk'}_{m,n,p,e} \\ + & prob_m \cdot disc_m \sum_{n,n',e,k'} (2 \cdot k_3 \cdot f^k_{m,n,n',e} - \upsilon^k_{m,n,n',e}) \lambda^{k'} f^{k'}_{m,n,n',e} \\ + & prob_m \cdot disc_m \sum_{n,e,e',k'} (2 \cdot k_5 \cdot x^k_{m,n,e,e'} - \phi^k_{m,n,e,e'}) \lambda^{k'} x^{k'}_{m,n,e,e'} \end{array} \right] \geq 0 \tag{B.13}$$

$$0 \leq \kappa_{m,n,n',e} \quad \perp \quad \overline{f^I}_{m,n,n',e} - \quad f^I_{m,n,n',e} \geq 0 \tag{B.14}$$

$$0 \leq \xi_{m,n,e} \quad \perp \quad \overline{x^I}_{m,n,e} - x^I_{m,n,e} \geq 0 \tag{B.15}$$

$$\theta \; free, \qquad -e^{kT} \lambda^k + 1 = 0 \tag{B.16}$$

B.3 Convergence Gap

$$CG^k = \sum_{m \in M} prob_m \cdot disc_m \cdot \Bigg[\sum_{n,p,d,e} \Bigg(\Bigg(\Big(int_{m,n,d,e} - slp_{m,n,d,e} \cdot (q^{Sk}_{m,n,p,d,e} + \sum_{p' \in P} q^{Sk}_{m,n,p',d,e})\Big) - \Big(int_{m,n,d,e} - slp_{m,n,d,e} \cdot \Big(\sum_{k' \in K} q^{Sk'}_{m,n,p,d,e} + \sum_{p' \in P} q^{Sk'}_{m,n,p',d,e} \cdot \lambda^{k'} \Big)\Big)\Bigg) \sum_{k' \in K} q^{Sk'}_{m,n,p',d,e} \cdot \lambda^{k'} \Bigg)$$
$$+ \sum_{n,p,e} \Bigg(\Big((2 \cdot k_1 \cdot q^{Pk}_{m,n,p,e} + k_2) - \Big(2 \cdot k_1 \cdot (\sum_{k' in K} q^{Pk'}_{m,n,p,e} \cdot \lambda^{k'}) + k_2 \Big)\Big) \cdot \sum_{k' in K} (q^{Pk'}_{m,n,p,e} + k_2) \cdot \lambda^{k'} \Bigg)$$
$$+ \sum_{n,n',e} \Bigg(\Big((2 \cdot k_3 \cdot f^{k}_{m,n,n',e} - \upsilon^{k}_{m,n,n',e}) - \Big(2 \cdot k_3 \cdot \Big(\sum_{k' \in K} f^{k'}_{m,n,n',e} \cdot \lambda^{k'} \Big) - \upsilon^{k}_{m,n,n',e} \Big)\Big) \cdot \sum_{k' \in K} f^{k'}_{m,n,n',e} \cdot \lambda^{k'} \Bigg)$$
$$+ \sum_{n,e,e'} \Bigg(\Big((2 \cdot k_5 x^{k}_{m,n,e,e'} - \phi^{k}_{m,n,e,e'}) - \Big(2 \cdot k_5 \cdot \Big(\sum_{k' \in K} x^{k'}_{m,n,n',e} \cdot \lambda^{k'} \Big) - \phi^{k}_{m,n,e,e'} \Big)\Big) \cdot \sum_{k' \in K} x^{k'}_{m,n,e,e'} \cdot \lambda^{k'} \Bigg)$$
$$+ \sum_{n,n',e} \Bigg(\Big(\iota^{k}_{m,n,n',e} - \sum_{k' \in K} \iota^{k'}_{m,n,n',e} \cdot \lambda^{k'}\Big) \cdot \Big(\overline{f}_{m,n,n',e} + \sum_{m' \in S^A(m)} f^{I\ k-1}_{m,n,n',e}\Big) \Bigg)$$
$$+ \sum_{n,e,e'} \Bigg(\Big(\nu^{k}_{m,n,e,e'} - \sum_{k' \in K} \nu^{k'}_{m,n,e,e'} \cdot \lambda^{k'}\Big) \cdot \Big(\overline{x}_{m,n,e,e'} + \sum_{m' \in S^A(m)} x^{I\ k-1}_{m,n,e,e'}\Big) \Bigg)$$
$$+ \sum_{n,p,e} \Bigg(\Big(\epsilon^{k}_{m,n,p,e} - \sum_{k' \in K} \epsilon^{k'}_{m,n,p,e} \cdot \lambda^{k'}\Big) \cdot \overline{x}_{m,n,p,e} \Bigg) \Bigg] \qquad (B.17)$$