ABSTRACT

Title of Document: SOLVING TWO-LEVEL OPTIMIZATION PROBLEMS WITH APPLICATIONS TO ROBUST DESIGN AND ENERGY MARKETS

Sauleh Ahmad Siddiqui
Doctor of Philosophy, 2011

Directed By: Steven A. Gabriel, Associate Professor
Department of Civil and Environmental Engineering

Shapour Azarm, Professor
Department of Mechanical Engineering

This dissertation provides efficient techniques to solve two-level optimization problems. Three specific types of problems are considered. The first problem is robust optimization, which has direct applications to engineering design. Traditionally robust optimization problems have been solved using an inner-outer structure, which can be computationally expensive. This dissertation provides a method to decompose and solve this two-level structure using a modified Benders decomposition. This gradient-based technique is applicable to robust optimization problems with quasiconvex constraints and provides approximate solutions to problems with nonlinear constraints. The second types of two-level problems considered are mathematical and equilibrium programs with equilibrium constraints. Their two-level structure is simplified using Schur's decomposition and reformulation schemes for absolute value functions. The resulting formulations are applicable to game theory problems in operations research and economics. The third type of two-level problem studied is discretely-constrained mixed linear complementarity problems. These are first formulated into a two-level mathematical program with equilibrium constraints and then solved using the aforementioned technique for mathematical and equilibrium programs with equilibrium constraints. The techniques for all three problems help simplify the two-level structure into one level, which helps gain numerical and application insights. The computational effort for solving these problems is greatly reduced using the techniques in this dissertation. Finally, a host of numerical examples are presented to verify the approaches. Diverse applications to economics, operations research, and engineering design motivate the relevance of the novel methods developed in this dissertation.

SOLVING TWO-LEVEL OPTIMIZATION PROBLEMS WITH APPLICATIONS TO ROBUST DESIGN AND ENERGY MARKETS

By

Sauleh Ahmad Siddiqui

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2011

Advisory Committee:
Associate Professor Steven A. Gabriel, Co-Advisor/Chair
Professor Shapour Azarm, Co-Advisor
Associate Professor Radu V. Balan
Professor Dianne P. O'Leary
Professor Lars J. Olson, Dean's Representative

Acknowledgements

First and foremost, I would like to thank my advisers Dr. Steven A. Gabriel and Dr. Shapour Azarm for all their help and support. Their perpetual encouragement, abundant patience, and careful guidance made this work possible. My time at this university was made memorable because of them, and I am proud to look back and realize how much I have learned from them.

I would also like to thank my committee members for agreeing to advise me on this work: Dr. Radu V. Balan, for overlooking the mathematical aspects in both my preliminary oral exam and dissertation; Dr. Dianne P. O'Leary, whose survival manual for graduate study in the computer and mathematical sciences provided valuable advice; and Dr. Lars J. Olson for graciously agreeing to serve as the Dean's representative. I am grateful that they have taken time out from their busy schedules to provide insight.

I also want to acknowledge funding received from the Office of Naval Research and the Norwegian Research Council. The work presented in this dissertation was supported in part by the Office of Naval Research Contract N000140810384. The LinkS project funded by Norwegian Research Council via the Norwegian University of Science and Technology and SINTEF (UMD Award number 013768-001) also supported part of the work in this dissertation. Such support does not constitute an endorsement by the funding agency of the opinions expressed in this dissertation.

Table of Contents

List of Tables

List of Figures

Nomenclature and Abbreviations

BCM	Billion Cubic Meters
DC-MLCP	Discretely-Constrained Mixed Linear Complementary Problems
DC-Nash	Discretely-Constrained Nash Game
EPEC	Equilibrium Program with Equilibrium Constraints
f	Objective Function (Unless stated otherwise)
g	Inequality Constraint Function (Unless stated otherwise)
KKT	Karush-Kuhn-Tucker Conditions
LBF	Pound Force
LP	Linear Program
MCF	Million Cubic Feet
MIP	Mixed Integer Program
MILP	Mixed Integer Linear Program
MMBTU	One million British Thermal Units; measurement of heat energy
MPEC	Mathematical Programs with Equilibrium Constraints
NCP	Nonlinear Complementary Problem
$\Re$	The real numbers
SOS	Special Ordered Set
x	Decision Variable
WGM	World Gas Model
Z	The Integers

Chapter 1: Introduction

1.1. Motivation and Objective

Mathematical modeling of problems arising in engineering and economics often requires formulations where optimal decisions need to be made at two different levels. These levels can be distinguished by time, space, decision choices, or even sets of players. An optimal decision at each level, we assume, can be obtained using an optimization problem.

Consider some of many types of decisions made by the computer processor manufacturer Intel. First while making the processor, manufacturing errors and uncertainty can lead to their "best" design being infeasible. If not infeasible, the design might not be the best choice under uncertainty. This decision needs to be made accounting for the uncertainty or errors that can develop after manufacturing the product. Second, while deciding the price (or quantity) of the processor, Intel would have to take into account what its competitors are doing and if the government has made any regulations regarding taxation or distribution. Setting a price, thus, not only depends on Intel's own costs but the strategy of other actors at a different level than Intel. Finally, Intel needs to decide the number of processors to ship to specific locations. Even considering a simplified version of the market makes this a complex problem as network dynamics, transportation costs, and local demand all weigh into the decision. But, more importantly, the processors can only be transported in positive integer number quantities, as opposed to fractional quantities.

All the problems classified above fall under the umbrella of two-level problems. The first decision, regarding uncertainty, requires the initial proposed design of the chip to be such that the presence of uncertainty does not cause the design to be infeasible and/or suboptimal. The decision is thus made to ensure feasibility of design constraints as well as minimum variation in a design's performance under uncertainty. Such a problem will be described in this dissertation as a *Robust Optimization* problem.

The second type of problem about making a profit-maximizing decision with other players present in a non-cooperative competitive environment is known as a *Stackelberg Game* in economics and falls under the broad heading of *Mathematical Programs with Equilibrium Constraints* or *MPECs.* These problems have a wide variety of applications, and in their general form can encompass robust optimization problems as well. A special class of MPECs with certain mathematical properties will be considered in this dissertation along with their extension to *Equilibrium Programs with Equilibrium Constraints* or *EPECs.*

The third problem is about solving non-cooperative games as well, except the decision at the second level is to make sure that the choice made is integer rather than continuous. This is more of a computational issue, but nevertheless the techniques to solve such problems have important applications. These problems fall into the class of *Discretely-Constrained Mixed Linear Complementarity Problems* or (*DC-MLCPs*).

The two levels are a common feature to all these problems, and the biggest challenge to overcome this two-level structure is computational time. A nested structure causes a large increase in computational effort with an increase in variables and/or decision space (Bialas & Karwan, 1982). The focus of this dissertation is on developing decomposition based solution techniques that reduce computational effort significantly for these three types of problems. These new techniques will then be implemented on a variety of examples from engineering and energy markets.

1.2. Research Components

1.2.1. Solving Robust Optimization Problems

The goal of robust optimization problems is to find an optimal solution that is minimally sensitive to uncertain factors. Uncertain factors can include inputs to the problem such as parameters, decision variables, or both. Given any combination of possible uncertain factors, a solution is said to be robust if it is feasible and the variation in its objective function value is acceptable within a given user-specified range. Previous approaches for general nonlinear robust optimization problems under interval uncertainty involve nested optimization and are not computationally tractable. The overall objective in this dissertation is to develop an original and efficient robust optimization method that is scalable and does not contain nested optimization. The proposed method is applied to a variety of numerical and engineering examples to test its applicability. Current results show that the approach is able to numerically obtain a locally optimal robust solution to problems with quasiconvex constraints ($\leq$ type) and an approximate locally optimal robust solution to general nonlinear optimization problems. A portion of this research component has been presented in (Siddiqui et al., 2011a) and (Siddiqui et al., 2011c).

1.2.2. Solving Mathematical Programs and Equilibrium Problems with Equilibrium Constraints

This dissertation presents an original method for solving mathematical programs and equilibrium problems with equilibrium constraints (MPECs and EPECs). Schur's decomposition followed by two separate methods of approximating absolute-value functions are presented and used to solve large-scale MPECs. The advantage of this method over traditional methods for solving MPECs is that computational time is much lower, which is corroborated by numerical examples. An extension to solve EPECs is also presented, along with a small numerical example. Finally, an application of the method to an MPEC representing the United States natural gas market is given. A portion of this research component has been presented in (Siddiqui & Gabriel, 2011b) and (Gabriel et al., 2011c).

1.2.3. Solving Discretely-Constrained Mixed-Integer Linear Complementarity Problems

This research thrust presents an original modification to a recent approach for solving discretely-constrained, mixed linear complementarity problems (DC-MLCPs). Such formulations include a variety of interesting and realistic models of which discretely-constrained Nash games and network equilibrium problems are considered. A methodology is provided to solve Nash-Cournot energy production games allowing some variables to be discrete. Normally, these games can be stated as mixed complementarity problems but only permit continuous variables in order to make use of each producer's Karush-Kuhn-Tucker conditions. The proposed approach allows for more realistic modeling and a compromise between integrality and complementarity to avoid infeasible situations. A mixed-integer, linear program formulation is used to solve the DC-MLCP in which both complementarity as well as integrality are allowed to be relaxed. A portion of this research component has been presented in (Gabriel et al., 2011a) and (Gabriel et al., 2011b).

1.3. Organization of Dissertation

The remainder of this dissertation is organized as follows. Chapter 2 provides background and a thorough literature review for the three proposed research components. Chapter 3 provides the proposed solution methodology for robust optimization problems. The chapter also provides several engineering applications as well as numerical examples. The chapter is concluded by an example of an application to a carbon emissions related problem. Chapter 4 provides details on the algorithm used to solve MPECs and EPECs as well as computational issues. The chapter also provides numerical examples to corroborate these approaches, as well as an application to the North American natural gas market. Chapter 5 provides the proposed solution technique for discretely-constrained mixed linear complementary problems with examples of discretely-constrained Nash games and energy networks. Chapter 6 provides conclusions and directions for future research. Figure 1.1 displays the organization of this dissertation. Note that the dashed line shows that a technique developed in Chapter 4 will be used in Chapter 5.

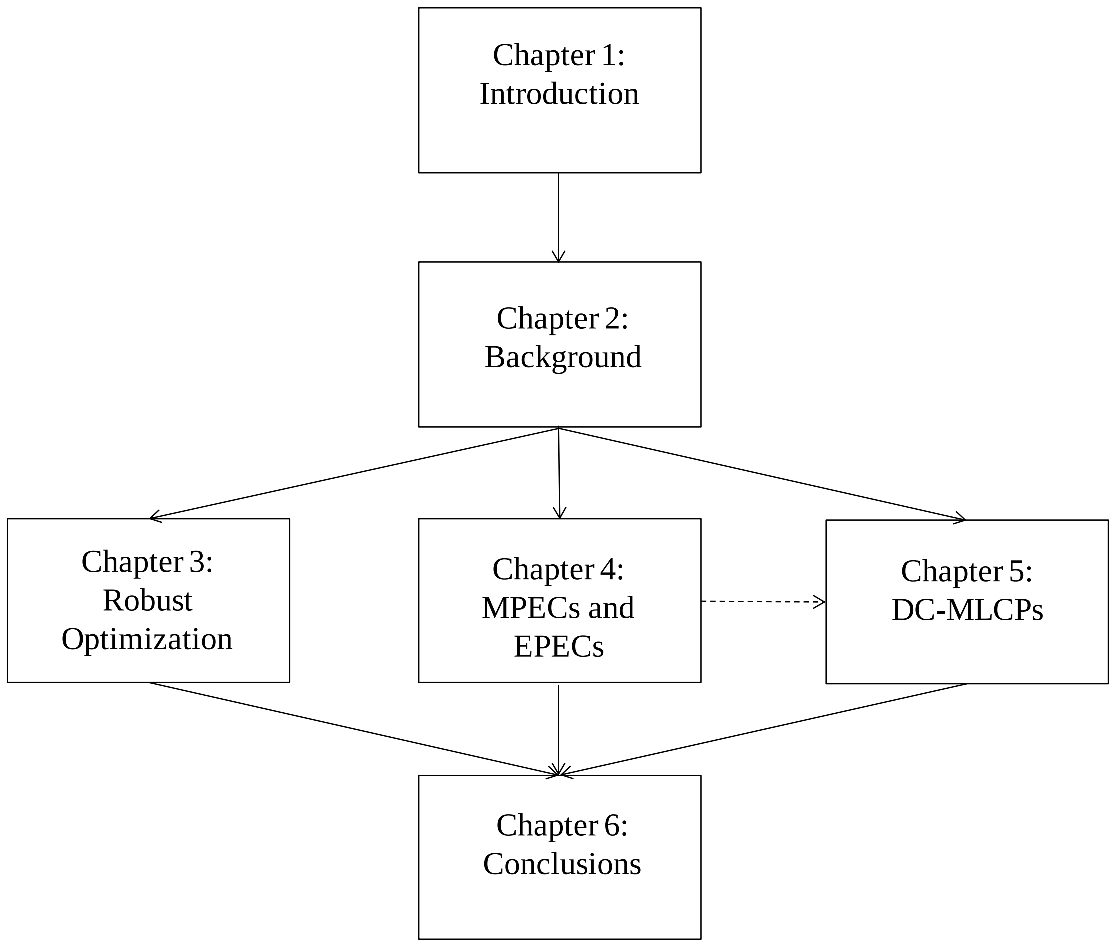

Figure 1.1: Organization of Dissertation

Chapter 2: Definitions and Literature Review

2.1. Introduction

This chapter will provide the necessary background for two-level optimization problems including definitions, terminologies, and a thorough literature review. This chapter will initially give mathematical definitions of two-level problems, and explain how robust optimization, MPECs and EPECs, and DC-MLCPs can all be cast as two-level problems.

While two-level problems can be shown to have a general formulation, each of the three different types considered in this dissertation need different treatment to come up with the most efficient solution. Although solving all three efficiently will involve the use of decomposition techniques, many other alternatives exist in the literature which will also be discussed. Finally, some preliminary mathematical ideas and traditional algorithms will also be introduced.

This chapter first goes through the definition and terminologies used in this dissertation. In particular, the next section defines each of the three two-level problems considered along with other definitions. A literature review is provided next followed by two preliminary topics.

2.2. Definitions and Terminologies

In general, the two-level optimization problems considered in this dissertation can be expressed as the following

$$
\begin{aligned}
&\min\ f(x^u, x^l) \\
&\text{s.t.} (x^u, x^l) \in \Omega \\
&x^l \in S(x^u)
\end{aligned}
\tag{2.1}
$$

where the continuous variables $x^u \in \Re^{n_u}$, $x^l \in \Re^{n_l}$ are, respectively, the vector of upper-level, lower-level variables, $f(x^u, x^l)$ is the upper level objective function[1], Ω is the joint feasible region between these sets of variables and $S(x^u)$ is the solution set of the lower-level problem that can be an optimization problem, a nonlinear complementarity problem (NCP) (Cottle et al., 2009), or a variational inequality problem (VI) (Faccinei & Pang, 2003). Figure 2.1 shows a diagrammatic representation of a two-level problem where the nested structure is revealed.

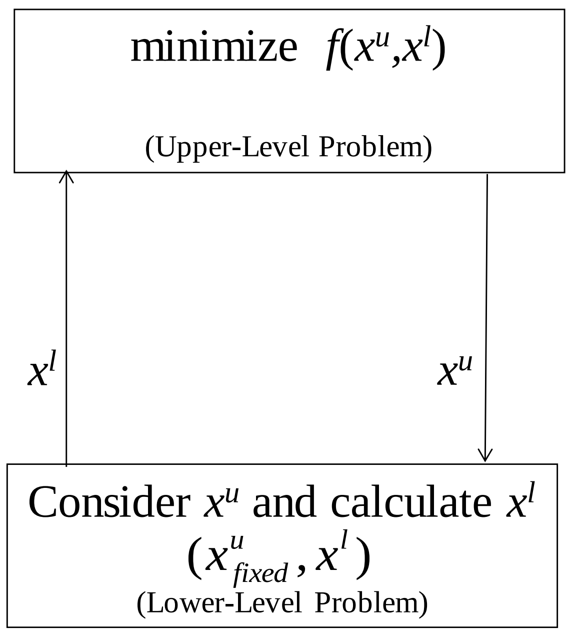

Figure 2.1: The Structure of a Two-Level Problem

[1] Note that when solving EPECs, several such two-level problems will be solved.

2.2.1. Robust Optimization

Table 2.1 describes the terminology used for robust optimization.

Table 2.1: Definition of Terms for Robust Optimization

Symbol	Interpretation
x	Vector of decision variables
f	Objective function to be minimized
$g_j(x, \hat{x})$	Constraint functions of the form "≤ 0"
Δx	Maximum deviations of uncertain variables from nominal values
$\hat{x}$	Deviations from nominal values of uncertain variables and parameters, respectively: $\hat{x} \in [-\Delta x, +\Delta x]$
Δf_0	User-specified tolerance for acceptable variation in objective function under uncertainty

The goal in robust optimization is to optimize the objective function with respect to uncertain decision variables x, satisfying all constraints and ensuring the objective function variation is kept within an acceptable range Δf_0, while accounting for uncertainty in decision variables. Specifically, this dissertation considers robust optimization problems of the form[2]:

---

[2] Note that equality constraints are considered to be formulated as two inequality constraints in formulation (2.2). Alternatively one can explore the approach for robust optimization with equality constraints (Rangavajhala et al., 2007) but that has not been explored in this dissertation.

$$
\begin{aligned}
&\min_{x} \; f(x,\hat{x}) \\
&s.t. \\
&\frac{f(x,\hat{x}) - f(x,0)}{\Delta f_0} \le 1 \\
&g_j(x,\hat{x}) \le 0 \quad j = 1,..,J \\
&x \in R^n, \hat{x} \in R^{n_u} \\
&\forall \hat{x} \in [-\Delta x, \Delta x]
\end{aligned}
\tag{2.2}
$$

where f and g are continuously differentiable in both x and $\hat{x}$. Figure 2.2 diagrammatically shows the structure of a robust optimization problem.

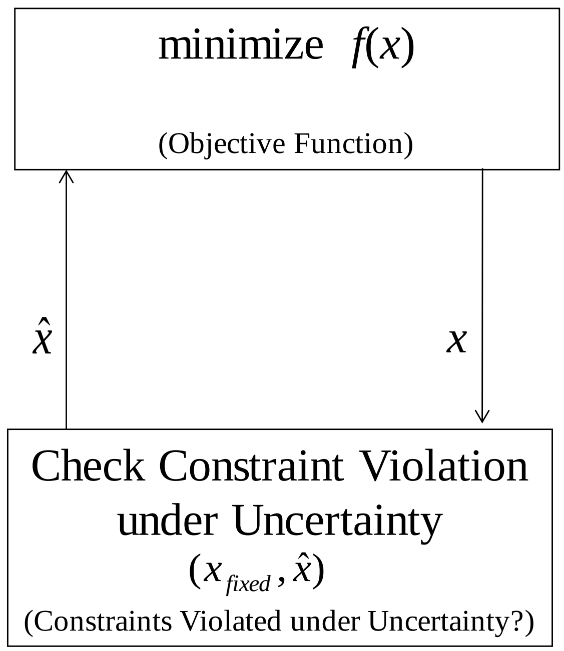

Figure 2.2: Representation of a Robust Optimization Problem

In the next few paragraphs, terms used in this dissertation are defined.

Definition 2.1: *<u>Quasiconvex Function:</u>* A function $g(x,\hat{x})$ is said to be quasiconvex in $\hat{x} \in [-\Delta x, +\Delta x]$ if for all $\hat{x} \in [-\Delta x, +\Delta x]$, $g(x,\hat{x}) \le \max\{g(x,\Delta x), g(x,-\Delta x)\}$ for all x (Bazaraa et al., 1993).

Definition 2.2: <u>*Objective robustness*:</u> For a candidate point x^c objective robustness holds if inequality

$$\left| \frac{f(x^c, \hat{x}) - f(x^c, 0)}{\Delta f_0} \right| \leq 1 \tag{2.3}$$

is satisfied for all $\hat{x} \in [-\Delta x, \Delta x]$.

Thus, this inequality ensures that the maximum objective function variation stays below a certain predetermined maximal amount Δf_0 when presented with deviations in uncertain variables and parameters.

Definition 2.3: <u>*Feasibility robustness*:</u> For a candidate solution x^c if

$$g_j(x^c, \hat{x}) \leq 0 \quad j = 1, \ldots, J \tag{2.4}$$

is satisfied for all $\hat{x} \in [-\Delta x, \Delta x]$ then feasibility robustness holds.

Note that equation (2.3) is just another constraint, so it can be easily incorporated into inequality (2.4) when stating a general formulation that only includes feasibility robustness. From this point on, inequality (2.3) will not be stated separately in any formulation but will be assumed to be incorporated in inequality (2.4). For a more detailed description on objective robustness, please refer to (Li et al., 2006).

Definition 2.4: <u>*Robust point*:</u> A robust point is both objectively and feasibly robust.

Definition 2.5: *Locally optimal robust*: For a robust optimization problem, a *locally optimal robust* solution x^*, is a robust point such that there exists a neighboring set U of robust solutions for which x^* is optimal ($f(x^*) \leq f(x), \forall x \in U$).

It is essential that the neighboring set be made up of only robust points otherwise the term is ill-defined. There is also a global counterpart as defined below.

Definition 2.6: *Globally optimal robust*: For a robust optimization problem, a *globally optimal robust* solution x^*, is a robust point such that x^* is optimal ($f(x^*) \leq f(x), \forall x$) in the feasible region.

2.2.2. Mathematical and Equilibrium Programs with Equilibrium Constraints

In general, a mathematical program with equilibrium constraints is given by [3]

$$\begin{aligned} &\min\ f(x,y) \\ &\text{s.t.} (x,y) \in \Omega \\ &y \in S(x) \end{aligned} \tag{2.5}$$

where the continuous variables $x \in \Re^{n_x}$, $y \in \Re^{n_y}$ are, respectively, the vector of upper-level, lower-level variables, $f(x,y)$ is the upper-level single-objective

---

[3] Without loss of generality, we assume that the variables x and y are nonnegative, which is incorporated in the decision space Ω.

function, Ω is the joint feasible region between these sets of variables and $S(x)$ is the solution set of the lower-level problem that can be an optimization problem, a nonlinear complementarity problem (NCP), or variational inequality problem (Luo et al., 1996).

One focus of this dissertation is when $S(x)$ is a solution to a nonlinear complementarity problem. Having a function $g: \Re^n \to \Re^n$, a nonlinear complementarity problem is to find a vector $z \in \Re^n$ such that:

$$\begin{aligned} z &\geq 0 \\ g(z) &\geq 0 \\ z^T g(z) &= 0 \end{aligned} \tag{2.6}$$

If $S(x)$ is the solution set of an NCP, (2.5) can be rewritten as

$$\begin{aligned} &\min\ f(x, y) \\ &\text{s.t.} (x, y) \in \Omega \\ &y \geq 0 \\ &g(x, y) \geq 0 \\ &y^T g(x, y) = 0 \end{aligned} \tag{2.7}$$

where $g(x, y): \Re^{n_x} \times \Re^{n_y} \to \Re^{n_y}$ is a vector-valued function.

Similarly, an EPEC is defined as a game between N players at the top level where each top-level player solves an optimization problem of the form (2.7). Hence, an EPEC with a common lower-level for each of the N upper-level players typical of Stackelberg leaders in energy production with the rest of the market represented by the lower-level problem is given by

$$
\begin{aligned}
&\min\ f_j(x,y) \qquad\qquad j=1,\ldots,N\\
&\text{s.t.}(x,y)\in\Omega\\
&y\geq 0\\
&g(x,y)\geq 0\\
&y^T g(x,y)=0
\end{aligned}
\tag{2.8}
$$

Figure 2.3 shows the diagrammatic representation[4] for an MPEC and Figure 2.4 shows the diagrammatic representation for an EPEC.

[4] Nash-Cournot in this diagram implies that an individual player solves their own optimization problem with other players' decisions being fixed.

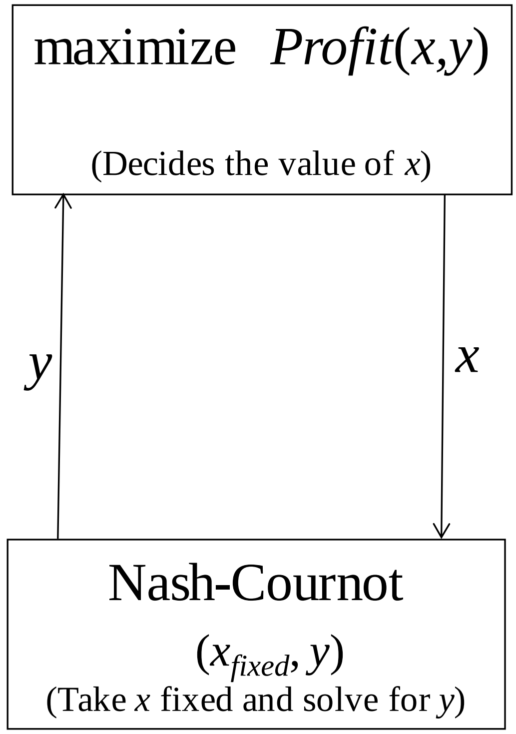

Figure 2.3: Representation of an MPEC

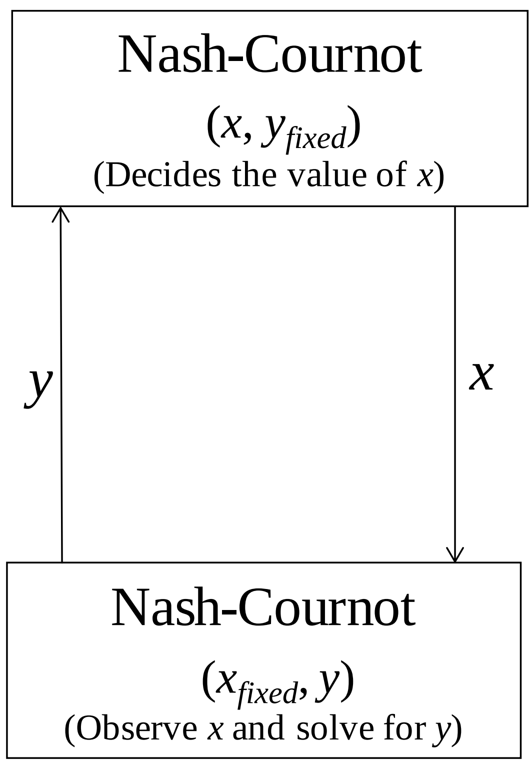

Figure 2.4: Representation of an EPEC

2.2.3 Discretely-Constrained Mixed Linear Complementarity Problems

It is not immediately obvious why the problem considered in this subsection is a two-level problem. The problem in its original form is not, but it needs to be converted into a two-level form for the particular solution technique (Gabriel et al., 2011a; Gabriel et al., 2011b) to be applicable. In general, a discretely-constrained mixed linear complementarity problem is given as follows: given the vector $q = (q_1, q_2)^T$ and matrix $A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$, find $z = (z_1, z_2)^T \in \Re^{n_1} \times \Re^{n_2}$ such that [5]

$$\begin{aligned}
&0 \le q_1 + \begin{pmatrix} A_{11} & A_{12} \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \perp z_1 \ge 0 \\
&0 = q_2 + \begin{pmatrix} A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \perp z_2, z_2 \text{ free} \\
&(z_1)_c \in \Re_+, c \in C_1, (z_1)_d \in Z_+, d \in D_1 \\
&(z_2)_c \in \Re_+, c \in C_2, (z_2)_d \in Z_+, d \in D_2
\end{aligned} \tag{2.9}$$

The indices for z_i, $i = 1, 2$ are partitioned into continuous-valued (denoted by the set C_i) and discrete-valued variables (denoted by the set D_i), i.e., $z_i = \left((z_i)_{C_i}^T, (z_i)_{D_i}^T \right)^T, i = 1,2$ with the continuous variables shown first without loss of generality. From here on, unless otherwise indicated, the discrete sets, $D_1 = \{0,1,..., N\}$ and $D_2 = \{-N_1,...,-1,0,1,..., N_2\}$ will be assumed with N, N_1, N_2, nonnegative integers.

---

[5] Here the superscript T denotes the transpose function. The symbol $\perp$ denotes complementary which means that the product of the two terms must be zero.

Finding a solution to this DC-MLCP can be thought of as a two-level problem, even though (2.9) formulates it in one level. The upper level minimizes deviations from an integer solution and complementary, i.e., ensures that as close as possible to an integer solution is obtained while satisfying complementary conditions with a minimum deviation as well, while the lower level solves a complementary problem assuming some deviation from integers has been fixed at the upper level. Figure 2.5 shows the diagrammatic representation of a discretely-constrained mixed linear complementary problem, while the following formulation describes the two-level formulation. Note that the first two inequality constraints and the first equality constraint (the first three constraints) form a complementary problem. Hence, the two-level structure[6] is apparent in the following formulation. Chapter 5 will describe in detail how this two-level formulation is obtained.

---

[6] Compare (2.10) to (2.1). The upper-level variables are ε and σ, i.e., $x^u \equiv \begin{pmatrix} \varepsilon \\ \sigma \end{pmatrix}$ and the first three lines in (2.10) define the upper-level problem. The lower level variables z, have $x^l \equiv \begin{pmatrix} z_1 \\ z_2 \end{pmatrix}$ and are part of the solution set of the discretely-constrained complementary problem given by the last four lines of (2.10).

$$
\begin{aligned}
&\min\left(1^T \varepsilon + 1^T \sigma\right) \\
&s.t. \\
&(\varepsilon)_d = \begin{pmatrix} \varepsilon_1 \\ \varepsilon_2 \end{pmatrix}_d \in \Re, d \in D_1 \cup D_2 \\
&(\sigma)_c \in \Re_+, c \in C_1 \\
&\left(q_1 + \begin{pmatrix} A_{11} & A_{12} \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} - \sigma\right) \perp (z_1 - \sigma) = 0 \\
&0 = q_2 + \begin{pmatrix} A_{21} & A_{22} \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \perp z_2, z_2 \text{ free} \\
&(z_1)_c \in \Re_+, c \in C_1, (z_1)_d - (\varepsilon_1)_d \in Z_+, d \in D_1 \\
&(z_2)_c \in \Re_+, c \in C_2, (z_2)_d - (\varepsilon_2)_d \in Z_+, d \in D_2
\end{aligned}
\tag{2.10}
$$

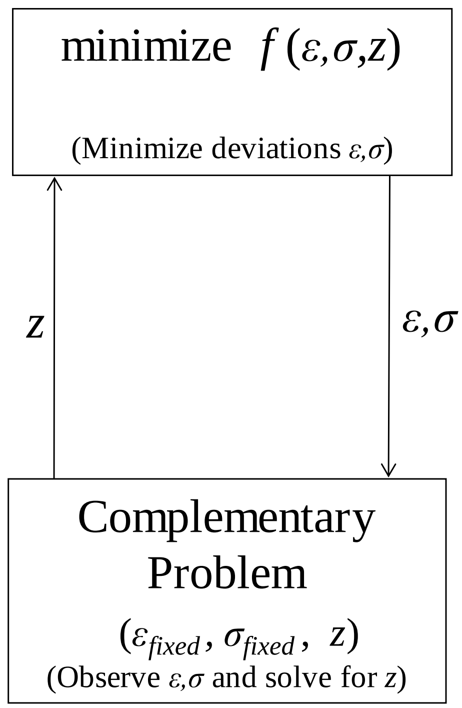

Figure 2.5: Representation of a Discretely-Constrained Mixed Linear Complementary Problem

2.3. Overview of Previous Work

2.3.1 Robust Optimization

This dissertation's approach for solving robust optimization problems (hereafter referred to as the *modified Benders method*), which will be described in Chapter 3, will now be compared to previous methods. A comprehensive review of the literature was conducted and the main distinctions between the proposed modified Benders method and previous works are presented as follows.

The robust optimization problems in the proposed modified Benders method also involve nonlinear (for example, Welded Beam and Heat Exchanger, which both involved nonconvex constraint functions) constraint functions[7]. This is more general than only considering linear constraint functions in the problem as reported in the literature (e.g., Balling et al., 1986; Ben-Tal & Nemirovski, 2002; Bertsimas & Sim, 2006; Soyster, 1973) or quadratic (e.g., Li et al., 2011) as well as other versions involving convex programs (e.g., Ganzerli & Pantelides, 1999) or linearization to solve the problem (e.g., Balling et al., 1986). The modified Benders method is able to obtain exact locally optimal robust solutions to problems with quasiconvex constraints as well as non-convex quadratic programs, which no one method in the reported literature is able to achieve. Other approaches also consider distributions for uncertainty (e.g., Lee et al., 2009; Lagaros & Papadrakakis, 2007) while the approach of this dissertation looks at a worst-case analysis for interval uncertainty[8] without any explicit probability distribution or a nested optimization structure. Moreover, the modified Benders method is able to handle large uncertainties which earlier methods

[7] In some cases, although not considered in this dissertation, a slightly stricter condition with convexity in the lower-level of the Benders decomposition method is needed. However, we did not encounter this in any of our test problems. A workaround to this problem is available in (Gabriel et al., 2010). This involves sampling the domain of the objective function of the lower-level optimization problem to determine the convex portions of this function. This numerical approximation scheme can be applied to the modified Benders method to determine convexity of the objective function of the lower-level optimization problem.

[8] Note that this dissertation considers robust optimization problems with interval uncertainty, while there is a substantive amount of literature considering other types of uncertainty. Refer to (Bertsimas & Sim, 2006; (Ben-Tal et al., 2009).

(e.g., Balling et al., 1986; Soyster, 1973; Ganzerli & Pantelides, 1999) were not able to tackle.

The proposed approach preserves the computational tractability, theoretically and practically, of the deterministic (i.e., nominal) problems. By contrast, under interval uncertainty, the computational effort for previous methods (e.g., Gunawan & Azarm, 2004; Li et al., 2006) to obtain robust solutions is much higher than their deterministic counterparts. However, results from a variety of numerical experiments show that the computational effort of solving the robust optimization problems is not much greater than that of their deterministic counterparts for the modified Benders method. Moreover, the modified Benders method is scalable, in that by numerical tests, the number of function calls per iteration increases at most linearly (numerical result) with an increase in the number of variables, uncertainty variables, and constraints.

Since this dissertation's approach is based on gradient-based methods, a globally optimal robust solution can never be guaranteed for the complete class of continuous, non-convex problems. However, this dissertation uses the idea of a locally optimal robust solution, and shows that this approach can obtain a locally optimal robust solution for nonlinear robust optimization problems.

In addition to the uncertainty in the data of the problems (i.e., the parameters), interval uncertainty is considered in the decision variables corresponding to manufacturing tolerances, implementation errors, etc. where optimized values cannot be achieved exactly, which is very common in practical engineering applications. For the current robust optimization formulations in the literature (e.g., Ben-Tal &

Nemirovski, 2008; Lu et al., 2010; Qiu & Wang, 2010; Zhu & Ting, 2001), considering uncertainty in the decision variables may considerably change those formulations or increase the complexity of the problem. The approach in this dissertation, however, keeps the same formulation and obtains locally optimal robust solutions to these problems with not much greater computational effort than the deterministic problem.

There has been an abundance of literature modifying Benders decomposition method (Benders, 1962) to solve various types of optimization problems. However, to my knowledge, there have not been any modifications to Benders method that solve nonlinear robust optimization problems with interval uncertainty although Benders-based robust optimization problems have been considered in other contexts. For example, (Velarde & Laguna, 2004) provided a Benders-based heuristic to solve the international source allocation problem. In this problem, a subset of international suppliers needs to be selected to meet local demand. The uncertainty is in the demand function parameters and exchange rates. However, their approach did not consider uncertainty in variables. For their approach to work, they needed to include control variables, which change depending on the uncertainty scenario to provide an easier route to solution. The approach in this dissertation does not require the introduction of such variables. Also, their methodology can't be extended to general nonlinear robust optimization problems. Saito and Murota (Saito & Murota, 2007) described a method to apply Benders decomposition to solve linear, mixed-integer, robust optimization problems with ellipsoidal uncertainty. However, this approach only works for linear problems. Finally, Montemanni (Montemanni, 2006) applied a Benders algorithm to a specific robust spanning tree problem, while Ng et al. (Ng et al., 2010) applied it to a specific semiconductor allocation problem that had uncertainty. Again, both approaches are not applicable to continuous, nonlinear robust optimization problems with interval uncertainty and have not modified Benders decomposition in the way this dissertation does.

There are related topics to robust optimization such as anti-optimization (e.g., (Qiu & Wang, 2010) and reliability-based design optimization (e.g., Zou & Mahadevan, 2006) that run into the same problems as described above of not being computationally tractable or only working for a certain simple type of problems. For example, Youn and Xi (Youn & Xi, 2009) modified a double loop problem (like robust optimization) into a single loop so that it becomes computationally easier. This work involves using an eigenvector dimension reduction method, and probability distributions, which may not be applicable to general nonlinear robust optimization problems. Also, neither of these papers has techniques that include interval uncertainty in parameters, in decision variables, along with being computationally efficient. The modified Benders method of this dissertation is not only directly relevant to robust optimization, but it handles the specific two-level structure of robust optimization in a less computationally intensive way.

2.3.3 Mathematical and Equilibrium Programs with Equilibrium Constraints

Finding optimal points for mathematical programs with equilibrium constraints (MPECs) involves solving a two-level optimization where the lower level is an equilibrium problem. In particular, having a complementarity problem (Cottle et al.,

2009) as the lower level implies that the complementarity constraint is a non-convex bilinear multiplicative term.

Many techniques exist to solve MPECs (Luo et al., 1996) but a popular way such MPECs have been solved is by using a disjunctive-constraints technique (Fortuny-Amat & McCarl, 1981). However, the two biggest drawbacks of disjunctive constraints are that the method is computationally expensive for large models (Luo et al., 1996) and that selecting a particular constant in the method is often troublesome (Gabriel & Leuthold, 2010). The solution can be extremely sensitive to the selection of this constant, and be far from the true answer if not selected correctly. Other methods (Steffensen & Ulbrich, 2010) and (Uderzo, 2010) also exist but have not been shown to work for large-scale models.

This dissertation presents a new method for solving MPECs, based on handling the bilinear, non-convex term using Schur's Decomposition and Special Ordered Sets of Type 1 (SOS Type 1) variables (Gabriel et al., 2006), along with a reformulation technique for absolute value terms. This method is applied to solve a small Stackelberg game with the number of players allowed to vary and an MPEC for the U.S. natural gas market to validate the proposed approach. A proposed extension along with a simple example to solve equilibrium programs with equilibrium constraints (EPECs) is also provided.

2.3.4. Discretely-Constrained Mixed Linear Complementarity Problems

As discussed before, complementary problems have had several applications in the literature, including solving Nash-Cournot games and network problems. Both Nash-Cournot games and network problems can be converted to mixed complementary problems by taking the Karush-Kuhn-Tucker (KKT) (Bazaraa et al., 1993) conditions to each player's optimization problem and combining them (Cottle et al., 2009).

A lot of applications of both these problems relate to energy markets. For example, Bard (1983,1988) developed algorithms for linear and convex two-level programming problems with applications to energy. Continuing, Bard and Moore (Bard & Moore, 1990) introduced a branch and bound algorithm for two-level problems resulting from complementary problems. Karlof and Wang (Karlof & Wang, 1996) applied a two-level approach to solving a flow shop scheduling problem, while (Labbéb et al., 1998) applied it to a model of taxation and highway pricing. Moore and Bard (Moore & Bard, 1990) and Wen and Huang (Wen & Huang, 1996) provided methods to solve mixed-integer two-level problems, but these methods are not applicable for solving DC-MLCPs.

More recently (Bard et al., 2000), (Fuller, 2010), (Fuller, 2008), (Gabriel & Leuthold, 2010), (Gabriel et al., 2010), (Hu et al., 2009), (Marcotte et al., 2001), (O'Neill et al., 2005), and (Scaparra & Church, 2008) have had applications of game theory problems to energy but none has considered complementary problems which are discretely constrained.

In some cases, solutions to discretely-constrained complementary problems do not exist. This is because satisfying integrality and complementary conditions together can prove to be more difficult than satisfying integrality and complementary conditions individually. However, solutions to the original optimization problem whose KKT conditions were used to formulate the complementary problem might still exist.

(Gabriel et al., 2011a), (Gabriel et al., 2011b) provide ways around this, and provide a relaxation technique that is able to solve DC-MLCPs. The distinguishing features of the proposed technique in both these papers with respect to other procedures reported in the technical literature (e.g., (Galiana et al., 2003), (Hogan & Ring, 2003), (Bjørndal & Jörnsten, 2008) are two-fold. First, the initial Nash-Cournot game or network problem is not manipulated to achieve prices that support market outcomes. Instead, optimality conditions of the original problem, with integrality and complementary conditions relaxed, are formulated and incorporated into a relaxation problem that allows realizing the tradeoff of integrality vs. complementarity. Second, instead of using a two-step procedure as in the literature, the technique in (Gabriel et al., 2011a) is single-step formulation of a two-level problem, and does not require altering the original problem by fixing integer variables to their optimal values to formulate a continuous problem. Hence this dissertation will concentrate on the work from the two papers (Gabriel et al., 2011a), (Gabriel et al., 2011b).

In both papers, (Gabriel et al., 2011a), (Gabriel et al., 2011b), however, disjunctive constraints (Fortuny-Amat & McCarl, 1981) are used to solve the resulting two-level problem. In this dissertation, a method developed in Chapter 4 will be used to solve these problems instead of disjunctive constraints. This is the main contribution in this dissertation, in that the method of Chapter 4 solves the DC-MLPCs much quicker than the disjunctive constraints method, and does not require the selection of a specific constant for disjunctive constraints.

2.4. Preliminaries

2.4.1. Benders Decomposition

This section describes standard Benders decomposition as a modified version will be used in Chapter 3. Table 2.2 describes the terminology used for explaining Benders decomposition.

Table 2.2: Definition of Terms for Benders Decomposition

Symbol	Interpretation
v_c	Complicating vector of variables to explain standard Benders decomposition
v_u	Uncomplicating vector of variables to explain standard Benders decomposition
$c(v_c)$	Constraints of complicating variable
$d(v_c,v_u)$	Constraints of uncomplicating and complicating Variables

Benders decomposition is used to efficiently solve linear and nonlinear programs (Conejo et al., 2006) and decomposes the original set of variables into both complicating and uncomplicating ones. Normally, integer variables are defined as *complicating* variables as fixing their values allows the problem to have a structure that provides an easy solution (assuming the rest of the problem is relatively easy to solve). However, in general, the complicating variables need not be integer and can be real-valued which, if fixed, render a simple or decomposable problem. Using notation provided above, Benders decomposition seeks to solve an optimization problem of the following form:

$$
\begin{aligned}
&\min_{v_c, v_u} f(v_c, v_u) \\
&s.t. \\
&c(v_c) \le 0 \\
&d(v_c, v_u) \le 0 \\
&where \\
&v_c \in U \subseteq \Re^n \\
&v_u \in \Re^m \\
&f(v_c, v_u) : \Re^{n \times m} \to \Re \\
&c(v_c) : \Re^n \to \Re^p \\
&d(v_c, v_u) : \Re^{n \times m} \to \Re^q
\end{aligned}
\tag{2.10}
$$

To solve this problem, the Benders decomposition technique fixes values of v_c (which are part of a set U that can be integers or other subsets of $\Re^n$) and solves the problem after first decomposing into a master problem and sub-problem (Conejo et al., 2006). To explain these notions of master and sub-problem, first define an auxiliary function $\alpha(v_c)$ as follows which expresses the objective function of the original problem as a function solely of the complicating variables.

$$
\begin{aligned}
&\alpha(v_c) = \min_{v_u} f(v_c, v_u) \\
&s.t. \\
&d(v_c, v_u) \le 0
\end{aligned}
\tag{2.11}
$$

Using the definition of $\alpha(v_c)$, the original problem (2.10) can be expressed as follows.

$$
\begin{aligned}
&\min_{v_c} \alpha(v_c) \\
&s.t. \\
&c(v_c) \le 0
\end{aligned}
\tag{2.12}
$$

Iteratively, a subproblem (2.13) is solved to approximate $\alpha(v_c)$ from above by fixing values of the complicating variables $(v_c = v_c^{fixed})$ and obtaining the dual variables λ to these constraints as shown in (2.13).

$$
\begin{aligned}
&\min_{v_u} f(v_c, v_u) \\
&s.t. \\
&v_c = v_c^{fixed} \quad (dual : \lambda) \\
&d(v_c, v_u) \leq 0
\end{aligned}
\tag{2.13}
$$

Then, the solution to the above problem, $\left(v_u^{sol}\right)$ and the dual variables $\left(\lambda^{sol}\right)$ are used to construct "Benders cuts" in the master problem to approximate the function[9] $\alpha(v_c)$ from below. Note that these cuts are iteratively added at each step until convergence. For simplicity, only one cut is shown here.

$$
\begin{aligned}
&\min_{\alpha, v_c} \alpha \\
&s.t. \\
&c(v_c) \leq 0 \\
&\alpha \geq f(v_c^{fixed}, v_u^{sol}) + \left(\lambda^{sol}\right)^T (v_c - v_c^{fixed})
\end{aligned}
\tag{2.14}
$$

Since the master problem (2.14) has a larger feasible region, it provides a lower bound (z_{lo}) while the more restricted subproblem (2.13) provides an upper bound (z_{up}) for the solution objective function value. These problems are solved iteratively until $(z_{up} - z_{lo})/(z_{lo})$ is less than some tolerance. As long as the function $\alpha(v_c)$ is convex, Benders decomposition converges to an optimal solution (Conejo et al., 2006).

---

[9] A sufficient condition for convergence is that the objective function in formulation (2.11) needs to be convex. However, the modification this paper presents, from the experimental results, does not need this convexity as the Benders cuts are modified. For more information, please refer to (Conejo et al., 2006) and (Gabriel et al., 2010).

2.4.2. Disjunctive Constraints

From (2.7), the set of solutions to

$$y^T g(x, y) = 0 \tag{2.15}$$

is nonconvex and can be computationally challenging to find even if $g(x, y)$ is linear.

One way is to use disjunctive constraints (Fortuny-Amat & McCarl, 1981). A large constant K is introduced, which can be difficult to select and cause computational issues (Gabriel & Leuthold, 2010), as well as a vector of binary variables r. Then, (2.8) is rewritten as

$$\begin{aligned}
&\min\ f(x, y) \\
&\text{s.t.}(x, y) \in \Omega \\
&0 \le y \le K(1 - r) \\
&0 \le g(x, y) \le Kr \\
&\textit{where} \\
&r \in \{0,1\}^{n_y} \text{ is a vector of binary values} \\
&K \in \Re_{++} \text{ is a large constant}
\end{aligned} \tag{2.16}$$

For large enough K, the solution set to (2.16) is equivalent to that of (2.7). The binary vectors and large K force componentwise, at least one of y or g to be 0. However, choosing K too small can cause errors in problem formulation (Gabriel & Leuthold, 2010) while choosing K too large can cause the condition number of the optimization problem (Renegar, 1995), (Renegar, 1994) to be high and result in numerical errors. One of the main aims of this section is to get around this problem by using decomposition and approximation techniques.

2.4.3. Approximating Nonlinear Functions using SOS Type 1 and Type 2 Variables

Note that disjunctive constraints are used to state disjunctive (either/or) logic statements. Hence, Chapter 4 will provide a different way to state such logic statements mathematically. For this, a reformulation of the absolute value function will be required, for which special ordered sets will be used.

Definition 2.7: A *Special Ordered Set of Type One* (SOS1 or SOS Type 1) is defined to be a set of non-negative variables for which at most one member from the set may be non-zero in a feasible solution. There are no other restrictions on the elements of the set, and they can be ordered in any way.

Among the uses for SOS1 variables, one popular one is to approximate functions. For example, consider a nonlinear function *g*(*x*) over a closed interval $x \in [x_{low}, x_{high}]$ on the positive real line. Given a partition of *n* points of the interval $x_i \in \{x_{low} = x_1, x_2, x_3, ..., x_n = x_{high}\}$, a new SOS Type 1 set of *n* variables $\{v_i\}_{i=1}^n$ can be introduced to approximate this nonlinear function. Then, *g*(*x*) can be expressed as

$$
\begin{aligned}
& g(x) = \sum_{i=1}^{n} v_i g(x_i) \\
& \textit{where} \\
& x = \sum_{i=1}^{n} v_i x_i \qquad\qquad (2.17) \\
& \sum_{i=1}^{n} v_i = 1 \\
& \{v_i\}_{i=1}^{n} \text{ are SOS 1 Variables}
\end{aligned}
$$

Figure 2.6 shows this nonlinear function being approximated by SOS Type 1 variables.

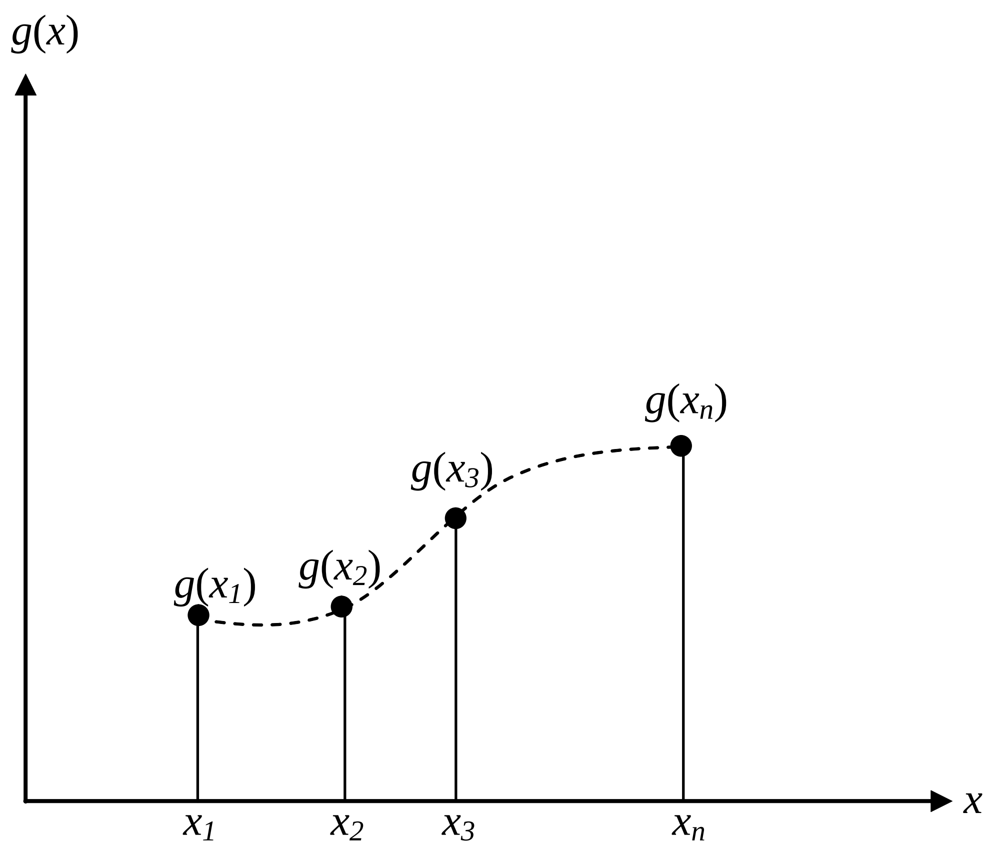

Figure 2.6: Approximating a Nonlinear Function Using SOS Type 1 Variable

Similarly, a piecewise approximation can also be developed using SOS Type 2 variables.

Definition 2.8: A *Special Ordered Set of type Two* (SOS 2 or SOS Type 2) is a set of nonnegative consecutive variables in which not more than two adjacent members may be non-zero in a feasible solution. No other restrictions are placed on the set.

Again, consider a nonlinear function *g*(*x*) over a closed interval $x \in [x_{low}, x_{high}]$ on the real line. Given a partition of *n* points of the interval $x_i \in \{x_{low} = x_1, x_2, x_3, ..., x_n = x_{high}\}$, a new SOS Type 2 set of *n* variables $\{u_i\}_{i=1}^{n}$ can be introduced to approximate this nonlinear function. Then, *g*(*x*) can be expressed as

$$
\begin{aligned}
& g(x) = \sum_{i=1}^{n} u_i g(x_i) \\
& where \\
& x = \sum_{i=1}^{n} u_i x_i \\
& \sum_{i=1}^{n} u_i = 1 \\
& \{u_i\}_{i=1}^{n} \text{ are SOS 2 Variables}
\end{aligned}
\tag{2.18}
$$

Figure 2.7 shows this nonlinear function being approximated by SOS Type 2 variables. The red line shows the piecewise linear approximation of the function.

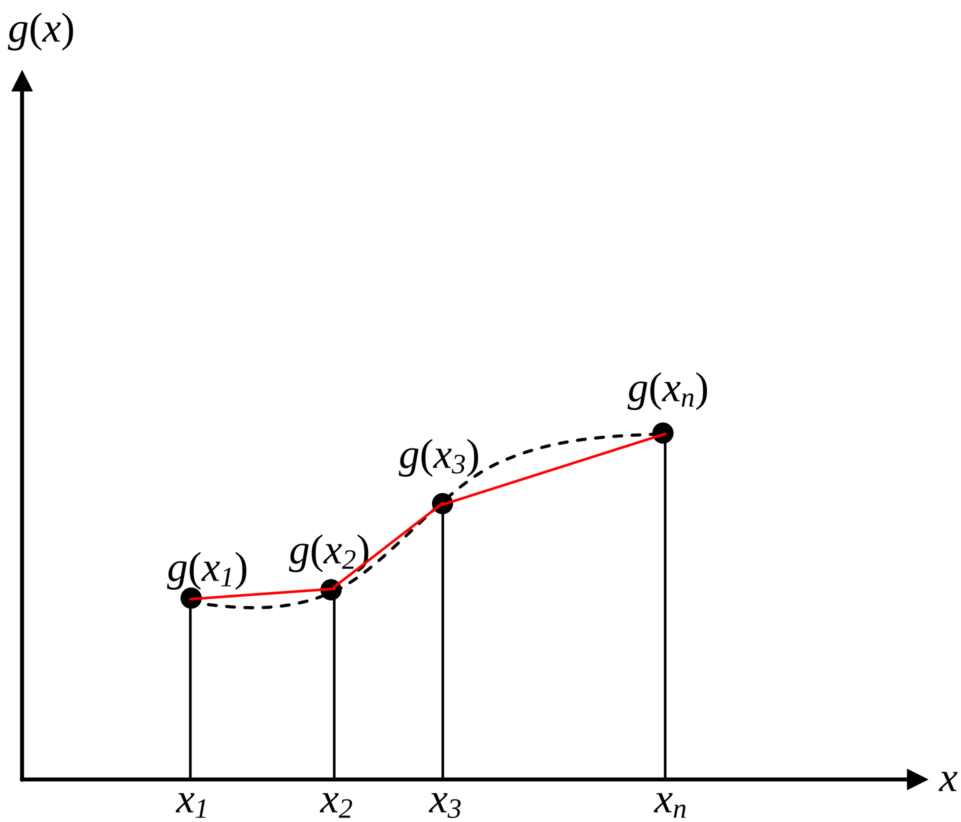

Figure 2.7: Approximation of Nonlinear Functions using SOS Type 2 Variables

The downside of SOS Type 2 variables is, of course, that it requires much more computational power than if SOS Type 1 variables were used. In Chapter 4, however, an absolute value function will be used. Hence, setting $g(x) = |x|$. For this purpose, only a set of two SOS Type 1 variables is required to reformulate this over the entire range. This is described by the following formulation.

$$\begin{aligned} & g(x) = v^{+} - v^{-} \\ & where \\ & x = v^{+} + v^{-} \\ & v^{+}, v^{-} \text{ are SOS Type1 Variables} \end{aligned} \qquad (2.19)$$

Since these variables encompass the whole range, no restriction on the sum being 1 is required. Moreover, this formulation will be used in Chapter 4 to solve MPECs.

Chapter 3: Solving Robust Optimization Problems Using a Modified Benders Method

3.1. Introduction

Engineering optimization problems often involve uncontrollable variations or uncertainties in factors like decision variables and/or parameters. Optimal solutions that might be deterministically feasible often end up being infeasible for a given realization of uncertain factors. Additionally, even small levels of variations can cause large degradations in the objective function value. Manufacturing errors, measurement problems, and uncertainty in environmental conditions are examples of sources for these variations.

Uncertainty can be handled with or without a probability distribution. Optimization problems that involve probability distributions are referred to as stochastic optimization problems. These are more suited for situations where accounting for worst-case uncertainty might result in foregoing performance. Optimization problems in this chapter are more suited for situations where any violation of constraints under uncertainty could result in the solution being unsuitable. Hence, a worst-case analysis needs to be appropriate for problems considered in this chapter.

In this section, an approach for robust optimization, e.g., (Ben-Tal et al., 2009), for linear, quadratic, convex, and non-convex programs is developed by applying a worst-case analysis using a decomposition method. No probability distribution is presumed[10], but only intervals with a nominal point (user- or problem-defined) are used to represent the uncertainty in decision variables and/or parameters. The problem structure in this dissertation reflects a real-world design situation, e.g., when information about uncertain factors during the early stages of a design process is often limited.

The two-level structure is apparent in robust optimization problems. The upper-level of a robust optimization problem is a decision based on a fixed level of uncertainty. The lower-level checks the feasibility of an optimal solution obtained from the upper-level. This chapter provides a way to decompose this two-level structure using Benders decomposition to solve the robust optimization problem. A portion of the material in this chapter has been presented previously, see (Siddiqui et al., 2011a) and (Siddiqui et al., 2011c).

3.2. Interval Uncertainty

A simple example will be presented first to motivate this method. Consider the optimization problem

---

[10] The statement that no probability distribution being presumed is to ensure the fact that probability does not come into play in any part of the discussed formulation. For instance, a uniform distribution over the whole interval of uncertainty can be assumed. However, the solution technique for robust optimization would involve solving the problem while ensuring there is a zero probability of constraint violation, thus taking probability out of the question. Therefore, it is informative to presume no probability distribution.

$$\begin{aligned}
&\min\ f(x) = -x_1 - 2x_2 \\
&s.t. \\
&\quad g_1 \equiv x_1 + x_2 \le 8 \\
&\quad g_2 \equiv -2x_1 + 1x_2 \le 5 \\
&\quad g_3 \equiv -x_1 - 3x_2 \le -10
\end{aligned} \tag{3.1}$$

A version of this problem with uncertainty looks like

$$\begin{aligned}
&\min\ f(x) = -x_1 - 2x_2 \\
&s.t. \\
&\quad g_1 \equiv (1+\hat{x}_1)x_1 + (1+\hat{x}_2)x_2 \le 8 \\
&\quad g_2 \equiv (-2+\hat{x}_3)x_1 + (1+\hat{x}_4)x_2 \le 5 \\
&\quad g_3 \equiv (-1+\hat{x}_5)x_1 + (-3+\hat{x}_6)x_2 \le -10 \\
&where \\
&\quad \forall \hat{x}_i \in [-0.1, +0.1] \qquad i = 1,\ldots,6
\end{aligned} \tag{3.2}$$

Note that parameter uncertainty has been introduced in the constraints of the problem. Realize also that, for example, if $\hat{x}_1 = \hat{x}_2 = 0.1$ in the first constraint of (3.2), then if x_1 and x_2 satisfy the following inequality

$$(1+0.1)x_1 + (1+0.1)x_2 \le 8 \tag{3.3}$$

then x_1 and x_2 also satisfy

$$\begin{aligned}
&(1+\hat{x}_1)x_1 + (1+\hat{x}_2)x_2 \le 8 \\
&\forall \hat{x}_i \in [-0.1, +0.1] \qquad i = 1,2
\end{aligned} \tag{3.4}$$

Hence, this “trick” can be applied to all parameters and we can get an optimization problem which will give us a robust solution. This approach will also define a *robust feasible region* which is the subset of the feasible region that only contains points feasible under worst case uncertainty as shown above. Figure 3.1 shows a comparison of the original feasible region (3.1) and the resulting robust feasible region (3.5), and the constraint functions of following equation (3.5) define the robust feasible region.

$$
\begin{aligned}
&\min \ f(x) = -x_1 - 2x_2 \\
&s.t. \\
&\quad g_1 \equiv (1+0.1)x_1 + (1+0.1)x_2 \le 8 \\
&\quad g_2 \equiv (-2+0.1)x_1 + (1+0.1)x_2 \le 5 \\
&\quad g_3 \equiv (-1+0.1)x_1 + (-3+0.1)x_2 \le -10
\end{aligned} \tag{3.5}
$$

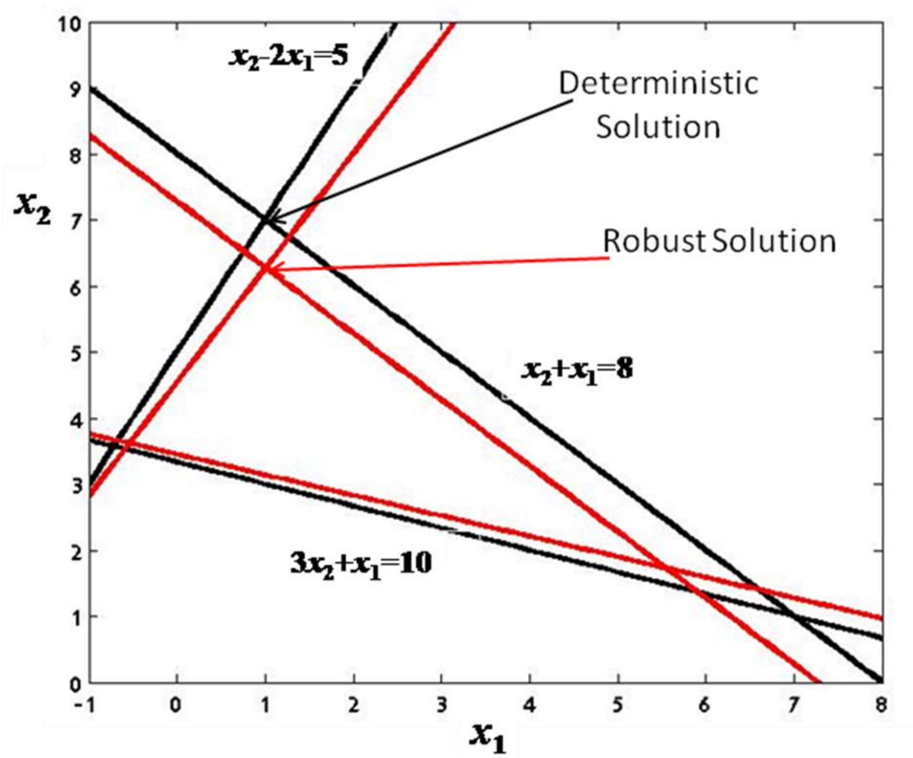

Figure 3.1: Comparison of the Feasible Region (Black) with the Robust Feasible Region (Red)[11]

Since this is a linear program, a solution will be one of the corner points of the feasible region. The solution to the deterministic problem (3.1) is $x_1 = 1$, $x_2 = 7$, $f(x) = -15$. The solution to the robust problem (3.2) can be found by looking at the corner points of the robust feasible region which gives $x_1 = 1$, $x_2 = 69/11$ (approximately 6.27), $f(x) = -149/11$ (approximately -13.54). Clearly, finding this robust feasible region greatly simplifies the robust optimization problem. The motivation behind the modified Benders method was to find this feasible region and then solve the easier optimization problem (3.5).

---

[11] Both the feasible region and the robust feasible region are the regions enclosed by the respective black and red lines.

Recall that the formulation from Chapter 2 for robust optimization problems (with objective robustness included in the constraints) is given by

$$\begin{aligned}
&\min_x \ f(x) \\
&s.t. \\
&g_j(x,\hat{x}) \le 0 \quad j = 1,\dots,J \\
&x \in R^n, \hat{x} \in R^{n_u} \\
&\forall \hat{x} \in [-\Delta x, \Delta x]
\end{aligned} \tag{3.6}$$

The end goal of this chapter is to solve problem (3.6). The method used is a modification of Benders decomposition to be described later. As described in Section 2.4.1, Benders decomposition decomposes an optimization problem into a master problem and a subproblem, with the variables being divided into complicating and uncomplicating ones. In (3.6), x are the uncomplicating variables and $\hat{x}$ are the complicating ones. As in Benders decomposition, an auxiliary function of $\hat{x}$ will be defined. A set of theoretical results will then be proven about the complicating variables $\hat{x}$ and the auxiliary function. The first set of theoretical results will show that an application of standard Benders decomposition to (3.6) when the objective and constraint functions are linear will yield a globally optimal robust solution (Algorithm 3.1). Then, an assumption on the quasiconvexity of the constraint functions $g_j(x,\hat{x})$ will be made to simplify the application of a modified Benders decomposition. This modified Benders decomposition with modified Benders cuts will then be applied to solve (3.6) when the constraint functions $g_j(x,\hat{x})$ are quasiconvex (heuristic Algorithm 3.2). Finally, a third heuristic is also presented which can be used to solve (3.6) when the constraint functions $g_j(x,\hat{x})$ are nonlinear (not necessarily quasiconvex).

To prove the theoretical results in this chapter, certain assumptions have to be made about optimization problem (3.6). In Section 2.2.1, assumptions on the functions f and g_j being continuous were stated. While the new Assumptions 3.1 and 3.2 may be relaxed for numerical application of the algorithms presented later, the theoretical results depend on them. The following are these assumptions and they have to do with the existence of solutions.

Assumption 3.1: The constraints g_j in (3.6), for any fixed value of uncertainty $\hat{x}$, form a convex, compact, nonempty feasible region over x.

Assumption 3.2: A globally optimal robust solution to (3.6) always exists.

Assumption 3.1 ensures that continuous functions g_j, $j = 1,\ldots, J$ are over a nonempty compact set so they obtain their maximum within this set via the Weierstrass Theorem (Royden, 1988). The convexity of the feasible region is included to ensure that a gradient-based algorithm can be applied successfully. Assumption 3.2 is stronger, and assumes that a solution exists to the robust optimization problem, while Assumption 3.1 does not take into account the $\forall \hat{x} \in [-\Delta x, \Delta x]$ clause in (3.6). Existence of solutions to robust optimization problems are difficult to prove. The presence of uncertainty means that with large enough values of $|\Delta x|$, there may not be even a feasible solution to the robust optimization problem, let alone a globally optimal robust solution. However, for example, problem (3.2) has an optimal robust solution. The algorithms in this chapter can be used with solvers which could detect if an optimization problem did not have a feasible solution.

The goal is to obtain values of x such that the formulation (3.6) gives an optimal solution to x regardless of the values of $\hat{x}$. Since this chapter only considers the worst-case analysis, the method aims to get the "worst" values of $\hat{x}$ for this problem (3.6). These are called "interval-optimal" values, as defined next.

Definition 3.1: *Interval-optimal*: An *interval-optimal* value for a particular candidate solution x^c and a constraint function g_j (for one j = 1,…, J) is defined as a point $\hat{x}^c \in [-\Delta x, +\Delta x]$ such that $g_j(x^c, \hat{x}) \leq g_j(x^c, \hat{x}^c)$ for all realizations of $\hat{x} \in [-\Delta x, +\Delta x]$. The point $\hat{x}^c \in [-\Delta x, +\Delta x]$ is a particular value of the $\hat{x}$ such that the constraint attains its maximum value at that particular value of uncertainty.

An interval-optimal point can be thought of as the value of uncertainty $\hat{x}$ that maximizes the value of g_j over all other realizations of uncertainty for a fixed value of x. The next definition takes this further.

Definition 3.2: *Globally Interval-optimal*: A *globally interval-optimal* value for a particular candidate solution x^c and set of constraint functions g_j; j = 1,…, J; is defined as a point $\hat{x}^c \in [-\Delta x, +\Delta x]$ such that $g_j(x^c, \hat{x}) \leq \max_j g_j(x^c, \hat{x}^c)$ for all realizations of $\hat{x} \in [-\Delta x, +\Delta x]$. The point $\hat{x}^c \in [-\Delta x, +\Delta x]$ is a particular value of the $\hat{x}$ such that the constraints attain their global maximum value at that particular value of uncertainty.

Note that any globally interval-optimal point is interval-optimal for at least one of the constraint functions g_j, j = 1,…, J. From example (3.2), for the constraint g_1 and the candidate solution $x^c = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ x_1 = 1, x_2 = 69/11 the associated interval-optimal value of $\hat{x}$ is $\hat{x}_i = 0.1$, i = 1,…, 6. This also happens to be the associated globally interval-optimal value of $\hat{x}$ for the solution x_1 = 1, x_2 = 69/11 and the set of constraints g_1, g_2, g_3.

The following lemma makes the connection between a robust point and its globally interval-optimal point. In equation (3.5), the globally interval-optimal values of the uncertainty elements helped determine the robust solution. Lemma 3.1 further strengthens this connection between a robust point and a globally interval-optimal point.

Lemma 3.1: A candidate solution (x^c) for problem (3.6) is a robust point if and only if its globally interval-optimal point

$$\hat{x}^c \in [-\Delta x, +\Delta x]$$

is such that

$$\max_j g_j(x^c, \hat{x}^c) \le 0 .$$

Proof: If (x^c) is a robust point (Definition 2.4), then it must be true that $\max_j g_j(x^c, \hat{x}) \le 0$ for all realizations of $\hat{x} \in \left[-\Delta x, +\Delta x\right]$. Hence, this implies that for the associated globally interval-optimal point $(\hat{x}^c)$, $\max_j g_j(x^c, \hat{x}^c) \le 0$ as

$\hat{x}^c \in [-\Delta x, +\Delta x]$. For the other side of the if and only if argument, suppose the associated globally interval-optimal point has $\max_j g_j(x^c, \hat{x}^c) \leq 0$. Then by the definition of globally interval-optimal, $\max_j g_j(x^c, \hat{x}) \leq 0$ for all realizations of $\hat{x} \in [-\Delta x, +\Delta x]$ which implies that (x^c) is a robust point.■

The next step is to relate Definition 3.2 to Benders decomposition as explained in Section 2.4.1. The robust optimization problem (3.6) will be solved using a modification of Benders decomposition. In this modification, the uncertainty variables $\hat{x}$ will be the complicating variables. Since there is a need for an auxiliary function as in equation (2.11), define the following function[12]

$$\begin{aligned}
&\alpha_u(\hat{x}) = \min_x \; f(x) \\
&s.t. \\
&g_j(x, \hat{x}) \leq 0 \quad j = 1, \ldots, J \\
&x \in R^n \\
&where\ \hat{x} \in R^{n_u}
\end{aligned} \tag{3.7}$$

Before proceeding, it is important to define one more term and make an assumption. The next definition is a slightly different one than Definitions 3.1 and 3.2, but is related to our motivation for finding interval-optimal points using a modified Benders decomposition.

---

[12] By Assumption 3.2, a solution to (3.7) always exists. This is because the optimization problem (3.7) is a relaxed version of the optimization problem (3.6).

Definition 3.3: <u>*Worst-Case Uncertainty*:</u> A *worst-case uncertainty* value $\hat{x}^{wc} \in [-\Delta x, +\Delta x]$ for optimization problem (3.6) is such that when $\hat{x} = \hat{x}^{wc}$ is fixed in (3.6), the solution of (3.8) below yields a globally optimal-robust solution.

$$
\begin{aligned}
&\min_{x} \; f(x, \hat{x}) \\
&s.t. \\
&g_j(x, \hat{x}) \le 0 \quad j = 1, \ldots, J \\
&x \in R^n, \hat{x} \in R^{n_u} \\
&\hat{x} = \hat{x}^{wc}
\end{aligned}
\tag{3.8}
$$

Note that a worst-case uncertainty value differs from an interval-optimal or globally interval-optimal value of uncertainty in that a worst-case uncertainty value does not have an associated predetermined variable *x* (but it is associated with a globally optimal robust solution *after* solving (3.8)) and is for the entire optimization problem. But it is trivial to note that a worst-case uncertainty value of $\hat{x}$ is a globally interval-optimal point of uncertainty for some globally optimal robust solution. The next assumption is required for the theoretical background of the modified Benders decomposition presented in this chapter.

Assumption 3.3: A worst-case uncertainty value exists for robust optimization problem (3.6) and is a globally interval-optimal point for a globally optimal robust solution x^*.

Note that Assumption 3.3 ensures that finding worst-case uncertainty values enables us to find a globally optimal robust solution. Problem (3.2) was an example of a robust optimization problem where $\hat{x}_i = 0.1$, i = 1,…, 6 was figured out to be the worst-case uncertainty value.

As an example, linear programs satisfy Assumption 3.3. In general, a linear constraint function g_j can be written as $g_j(x,\hat{x}) = \sum_{i=1}^{n_x} c_i x_i + \sum_{i=1}^{n_u} d_i \hat{x}_i + C$ where C is a real number. Here, as in example (3.2), if there is a variable x such that $g_j(x) = \sum_{i=1}^{n_x} c_i x_i + \sum_{i=1}^{n_u} |d_i| \Delta x_i + C \le 0$, then it is also true that $g_j(x,\hat{x}) = \sum_{i=1}^{n_x} c_i x_i + \sum_{i=1}^{n_u} d_i \hat{x}_i + C \le 0$ for all $\hat{x} \in [-\Delta x, +\Delta x]$. Therefore, the interval-optimal value of $\hat{x}_i$ can be calculated to be $+\Delta x_i$ if d_i is positive and $-\Delta x_i$ if d_i is negative. From this argument, for all the g_j constraints a globally interval-optimal value can be calculated and so can a worst-case uncertainty value. For more information on problems that satisfy Assumption 3.3, please refer to (Ben-Tal et al., 2009).

The following lemma shows a property of the new auxiliary function (3.7) which connects a globally optimal robust point to its globally interval-optimal point. This will later be used in modifying Benders decomposition to obtain solutions to robust optimization problems.

Recall that for problem (3.2), the globally optimal robust point was x_1 = 1, x_2 = 69/11 and its associated globally interval-optimal point (and the worst-case uncertainty value) was $\hat{x}_i = 0.1$, i = 1,…, 6. Note that for example (3.2), $\alpha_u(0.1) = -149/11$, which happens to be the function value of the globally optimal robust solution to example (3.2). This is no coincidence as shown by the following lemma.

Lemma 3.2: Under Assumptions 3.1-3.3, let *x be a globally optimal robust solution (Definition 2.6) to (3.6) and $\hat{x}*$ an associated globally interval-optimal point. If (i)** $\alpha_u(\hat{x}*) \geq \alpha_u(\hat{x})$ **for all realizations of** $\hat{x} \in [-\Delta x, +\Delta x]$ **, then (ii)** $\alpha_u(\hat{x}*) = f(x*)$.

Proof: Since *x** is a globally optimal robust point, it is automatically a robust point (Definition 2.6) so by Lemma 3.1, $\max_j g_j(x*, \hat{x}) \leq 0$ for all realizations of $\hat{x} \in [-\Delta x, +\Delta x]$. Note that the value $\alpha_u(\hat{x}*)$ was calculated by minimizing *f*(*x*) while fixing $\hat{x} = \hat{x}*$ in (3.7). Since *x** is in the feasible region for (3.7) and by Assumption 3.2, a solution always exists to (3.7), $\alpha_u(\hat{x}*) \leq f(x*)$. The next step will show with the help of a contradiction argument that $\alpha_u(\hat{x}*) \geq f(x*)$.

Suppose that $\alpha_u(\hat{x}*) < f(x*)$. By the statement of this lemma, $\alpha_u(\hat{x}*) \geq \alpha_u(\hat{x})$ for all $\hat{x} \in [-\Delta x, +\Delta x]$. Using (3.7) by fixing $\hat{x} = \hat{x}*$, let x' (dependent on $\hat{x}*$) be a solution to the minimization problem in (3.7) such that $\alpha_u(\hat{x}*) = f(x')$. Then (i) implies $f(x') \geq \alpha_u(\hat{x})$ for all $\hat{x} \in [-\Delta x, +\Delta x]$. By our contradictory assumption, this also implies $f(x*) > \alpha_u(\hat{x}*) = f(x') \geq \alpha_u(\hat{x})$ which simplifies to $f(x*) > \alpha_u(\hat{x})$ for all $\hat{x} \in [-\Delta x, +\Delta x]$. Note that the condition $f(x*) > \alpha_u(\hat{x})$ for all $\hat{x} \in [-\Delta x, +\Delta x]$ violates Assumption 3.3. By Assumption 3.3, there exists a worst-case uncertainty value, i.e., there exists a $\hat{x}^{wc} \in [-\Delta x, +\Delta x]$ such that $f(x^*) = \alpha_u(\hat{x}^{wc})$. But this would imply $\alpha_u(\hat{x}^{wc}) > \alpha_u(\hat{x})$ for all $\hat{x} \in [-\Delta x, +\Delta x]$ which is a contradiction. Hence, this contradiction shows that $\alpha_u(\hat{x}^*) \geq f(x^*)$. Combining the two inequalities $\alpha_u(\hat{x}^*) \geq f(x^*)$ and $\alpha_u(\hat{x}^*) \leq f(x^*)$ gives $\alpha_u(\hat{x}^*) = f(x^*)$.■

The next two theorems form the basis of the modified Benders method to be introduced later in this chapter. The first shows a particular characteristic of a worst-case value of uncertainty. The second shows that a particular characteristic of an uncertainty variable value can be used to find a globally optimal robust solution. The modified Benders method of this chapter will aim to find this value.

Theorem 3.1: Under Assumptions 3.1-3.3, let the worst-case value of uncertainty for (3.6) be $\hat{x}^{wc}$. Then, $\alpha_u(\hat{x}^{wc}) \geq \alpha_u(\hat{x})$ for all realizations of $\hat{x} \in [-\Delta x, +\Delta x]$.

Proof: Let x^* be a globally optimal robust solution to (3.6). Then, by Definition 3.3, $\alpha_u(\hat{x}^{wc}) = f(x^*)$. Problem (3.6) has the same objective function as (3.7) but the feasible region of (3.6) is a subset of the feasible region of (3.7). Therefore, for any fixed $\hat{x}$ $f(x^*) \geq \alpha_u(\hat{x})$. Therefore, $\alpha_u(\hat{x}^{wc}) \geq \alpha_u(\hat{x})$ for all realizations of $\hat{x} \in [-\Delta x, +\Delta x]$.■

Theorem 3.2: Under Assumptions 3.1-3.3, suppose there exists a unique uncertainty value $\hat{x}^c$ for which $\alpha_u(\hat{x}^c) > \alpha_u(\hat{x})$ for all realizations of

$\hat{x} \in [-\Delta x, +\Delta x]$. Then, a solution to optimization problem (3.9) will be a globally optimal robust solution to problem (3.6).

$$\begin{aligned}
&\min_{x} \ f(x) \\
&s.t. \\
&g_j(x, \hat{x}) \le 0 \quad j = 1,\dots,J \\
&\hat{x} = \hat{x}^c \\
&x \in R^n, \hat{x} \in R^{n_u}
\end{aligned} \tag{3.9}$$

Proof: Let x^c be a solution to (3.9). Note that $\alpha_u(\hat{x}^c) = f(x^c)$ by (3.7). By Assumption 3.3, there exists a worst-case uncertainty value $\hat{x}^{wc}$ such that $\alpha_u(\hat{x}^{wc}) = f(x^*)$, where x^* is a globally optimal robust solution. Since x^* is a solution to (3.6), it is also feasible to (3.9) as the feasible region for (3.6) is a subset of the feasible region for (3.9). Hence, $f(x^c) \le f(x^*)$, which implies $\alpha_u(\hat{x}^c) \le \alpha(\hat{x}^{wc})$. But according to the statement of this theorem, $\alpha_u(\hat{x}^c) > \alpha_u(\hat{x})$ for all realizations of $\hat{x} \in [-\Delta x, +\Delta x]$. Hence, $\hat{x}^c = \hat{x}^{wc}$ because this theorem also states that this value of $\hat{x}^c$ is unique. Therefore, $\hat{x}^c = \hat{x}^{wc}$. By Definition 3.3, (3.9) gives a globally optimal robust solution. ■

The purpose of Theorem 3.2 is that if the following optimization problem [13] (3.10) has a unique solution, that solution can be used to find the solution to (3.6).

---

[13] Note that the function $\alpha_u(\hat{x})$ is not known in closed form but will be later be approximated using a variation on Benders cuts.

$$\begin{aligned} &\max_{\hat{x}} \alpha_u(\hat{x}) \\ &s.t. \\ &\hat{x} \in [-\Delta x, \Delta x] \\ &\hat{x} \in R^{n_u} \end{aligned} \tag{3.10}$$

Theorem 3.2 also shows that finding globally interval-optimal points can help us obtain a globally optimal robust solution. For quasiconvex constraint functions, we know that the globally interval-optimal values will lie on one of the endpoints of the vector interval[14] $[-\Delta x, \Delta x]$, which follows directly from the definition of quasiconvexity provided in Chapter 2 and is taken advantage of in the following Corollary 3.1. Note that there are 2^{n_u} such endpoints, where n_u is the dimension of the vector Δx and also the dimension of the endpoints. For purposes of notation, let the endpoints of the vector interval be denoted by $(V)_1$, $(V)_2$, $(V)_3$, …, $(V)_{2^{n_u}}$. Each endpoint vector $(V)_k$, $k = 1,\dots, 2^{n_u}$ is defined such that each of its elements $(V_i)_k$ is either Δx_i or $-\Delta x_i$, i.e., $(V_i)_k \in \{-\Delta x_i, \Delta x_i\}$ for $i = 1,\dots, n_u$. The idea that the maximum of the constraint functions lies on one of the endpoints of the vector interval can be used to ascertain that any globally interval-optimal point (and thus, worst-case uncertainty value) will also lie on one of the endpoints of the vector interval.

Corollary 3.1: If the constraint functions g_j , j = 1,…, J are quasiconvex in (3.6), then solving (3.11) is equivalent to solving (3.6) and solving (3.12) is equivalent to

[14] This vector interval $[-\Delta x, \Delta x]$ is of n_u dimensions. So the endpoints of this vector interval are actually all the corner points of an n_u-dimensional rectangle.

solving (3.7). Additionally, the maximum value of $\alpha_u(\hat{x})$ is achieved at one of the endpoints $\{(V)_k\}_{k=1}^{2^{n_u}}$.

$$\begin{aligned}
&\min_x \ f(x) \\
&s.t. \\
&g_j(x,\hat{x}) \le 0 \quad j = 1,\dots,J \\
&x \in R^n, \hat{x} \in R^{n_u} \\
&\forall \hat{x} \in \{(V)_k\}_{k=1}^{2^{n_u}}
\end{aligned} \tag{3.11}$$

$$\begin{aligned}
&\alpha_u(\hat{x}) = \min_x \ f(x) \\
&s.t. \\
&g_j(x,\hat{x}) \le 0 \quad j = 1,\dots,J \\
&x \in R^n \\
&where \ \hat{x} \in \{(V)_k\}_{k=1}^{2^{n_u}}
\end{aligned} \tag{3.12}$$

Proof: The aim of this theorem is to show that if g_j, for all $j = 1,\dots, J$ is quasiconvex in $\hat{x}$, then $\forall \hat{x} \in \{(V)_k\}_{k=1}^{2^{n_u}}$ can replace the condition $\forall \hat{x} \in [-\Delta x, \Delta x]$ in (3.6). Note that $g_j(x,\hat{x}_i) = g_j(x, \lambda(-\Delta x_i) + (1-\lambda(\Delta x_i))) \le \max\{g_j(x,-\Delta x_i), g_j(x,\Delta x_i)\}$ for all $\lambda \in [0,1]$ and a real-valued $\hat{x}_i \in [-\Delta x_i, \Delta x_i]$. For any x, if $g_j(x,\hat{x}) \le 0, \forall \hat{x} \in [-\Delta x, \Delta x]$, then $g_j(x,\hat{x}) \le 0, \forall \hat{x} \in \{(V)_k\}_{k=1}^{2^{n_u}}$ because $\{(V)_k\}_{k=1}^{2^{n_u}} \subseteq [-\Delta x, \Delta x]$. Moreover, for any x if $g_j(x,\hat{x}) \le 0, \forall \hat{x} \in \{(V)_k\}_{k=1}^{2^{n_u}}$, then $g_j(x,\hat{x}) \le 0, \forall \hat{x} \in [-\Delta x, \Delta x]$ because g_j is quasiconvex in $\hat{x}$. Hence, $g_j(x,\hat{x}) \le 0, \forall \hat{x} \in \{(V)_k\}_{k=1}^{2^{n_u}}$, if and only if $g_j(x,\hat{x}) \le 0, \forall \hat{x} \in [-\Delta x, \Delta x]$. Therefore, $\forall \hat{x} \in \{(V)_k\}_{k=1}^{2^{n_u}}$ can replace the condition $\forall \hat{x} \in [-\Delta x, \Delta x]$ in (3.6) whenever g_j, for all j

= 1,…, J is quasiconvex in $\hat{x}$. Therefore, solving (3.11) is the same as solving (3.6) and solving (3.7) is the same as solving (3.12).

Note that a value of $\hat{x}$ that maximizes the constraint g_j for a fixed value of x in the expression $\max_{\hat{x}\in\{-\Delta x,\Delta x\}^{n_u}} \{g_j(x,\hat{x})\}$ is an interval-optimal value. Moreover, a value of $\hat{x}$ that maximizes the expression $\max_j \left\{ \max_{\hat{x}\in\{-\Delta x,\Delta x\}^{n_u}} \{g_j(x,\hat{x})\} \right\}$ is a globally interval-optimal value of $\hat{x}$. Hence, the globally interval-optimal values of $\hat{x}$ lie on the endpoints $\{(V)_k\}_{k=1}^{2^{n_u}}$. In particular, by Assumption 3.3, a worst-case value of uncertainty $\hat{x}^{wc}$ for problem (3.11) exists and is a globally interval-optimal value. Hence, $\hat{x}^{wc}$ also lies on one of the endpoints. By Theorem 3.1, $\alpha_u(\hat{x}^{wc}) \geq \alpha_u(\hat{x})$ for all realizations of $\hat{x} \in [-\Delta x, +\Delta x]$, which is the maximum value of $\alpha_u(\hat{x})$. Hence, the maximum value of $\alpha_u(\hat{x})$ is achieved at one of the endpoints.■

The next section provides a modified Benders decomposition method to solve robust linear programs. Note that so far we have proven facts about the auxiliary function $\alpha_u(\hat{x})$. Standard Benders decomposition approximates this function using standard Benders cuts. With the support of theoretical results, the first algorithm in the next chapter will apply standard Benders decomposition to obtain a solution to (3.6). Two further heuristic algorithms are provided to solve (3.6) when the constraint functions are quasiconvex and nonlinear, respectively.

3.3 Modified Benders Decomposition

3.3.1. Formulation of Approach: Solving Robust Linear Programs

The strategy will be to find optimal values for variables x and interval-optimal values for $\hat{x}$. One can think of this as attempting to check the robustness of a candidate solution by partitioning the uncertainty interval and checking feasibility at each point. Clearly, if all constraints are feasible when fixed with the interval-optimal values for uncertainty elements, then the candidate solution is robust.

Standard Benders decomposition is advantageous when the problem structure dictates that fixing certain variables will lead to a simpler problem to solve. The idea behind Benders decomposition is to fix a set of complicating variables and solve a resulting simpler subproblem while iterating between it and solving a master problem that computes values for the complicating variables. The robust optimization problem (3.6) also has a simple structure when certain variables are fixed. Fixing $\hat{x}$ results in an optimization problem much simpler to solve than a robust optimization problem.

The Benders cuts added in (2.14) serve to approximate the function α described in (2.11). Since the objective function in (3.7) is being maximized by equation (3.10), for the successful application of Benders decomposition, the function $\alpha_u(\hat{x})$ needs to be concave.

Theorem 3.3: For linear objective and constraint functions in (3.6), the function $\alpha_u(\hat{x})$ is concave.

Proof: This is a well-known result from linear programming theory (Murty, 1983). The function $\alpha_u(\hat{x})$ is piecewise-linear, continuous, and concave. ■

Because Theorem 3.2 requires a unique maximum value of $\hat{x}$ to guarantee a solution, the following lemma gives conditions under which this is possible for linear robust optimization problems. Before proceeding, the definition of a slope between two points is needed.

Definition 3.4: *Slope:* The slope of the function $\alpha_u(y): R^{n_u} \to R$ between two vectors $y^1 \in R^{n_u}, y^2 \in R^{n_u}$ is defined as the vector $\lambda \in R^{n_u}$ where $\lambda_i = \dfrac{\alpha(y^2) - \alpha(y^1)}{y_i^2 - y_i^1}$ for each element $i = 1,\ldots, n_u$.

Lemma 3.3: Let $\alpha_u(\hat{x})$ be a piecewise linear, continuous, concave function over $[-\Delta x, \Delta x]$. Then $\alpha_u(\hat{x})$ achieves its maximum at a point $\hat{x}^{wc} \in [-\Delta x, \Delta x]$. Now suppose this maximum is achieved at one of the endpoint vectors $\{(V)_k\}_{k=1}^{2^{n_u}}$ and suppose the slope between $\hat{x}^{wc}$ and any other vector $\hat{x} \in [-\Delta x, \Delta x]$ is nonzero (every element of the slope between the two points is nonzero). Then $\alpha_u(\hat{x})$ has a unique maximum.

Proof: Since $\alpha_u(\hat{x})$ is a continuous function over a nonempty compact set, the vector interval $[-\Delta x, \Delta x]$, by the Weierstrass theorem (Royden, 1988) $\alpha_u(\hat{x})$ achieves its maximum at a vector in $[-\Delta x, \Delta x]$.

Let the maximum of $\alpha_u(\hat{x})$ be achieved at $\hat{x}^{wc} \in \{(V)_k\}_{k=1}^{2^{n_u}}$, which is one of the endpoints . We want to show that this point is unique. This proof will follow a contradiction argument. Suppose there exists another distinct point $\hat{x}' \neq \hat{x}^{wc}$ such that $\alpha_u(\hat{x}^{wc}) = \alpha_u(\hat{x}')$ and $\alpha_u(\hat{x}') \geq \alpha_u(\hat{x})$ for all $\hat{x} \in [-\Delta x, \Delta x]$. Since $\alpha_u(\hat{x})$ has nonzero slope between $\hat{x}^{wc}$ and any other point in $[-\Delta x, \Delta x]$, there exists a point $\hat{x}^m$ such that $\hat{x}^m \neq \hat{x}', \hat{x}^m \neq \hat{x}^{wc}$ which is a strict convex combination of $\hat{x}^{wc}$ and $\hat{x}'$ such that $\alpha_u(\hat{x}^m) \neq \alpha_u(\hat{x}^{wc})$. Hence, there exists a $\lambda \in (0,1)$ such that $\hat{x}^m = \lambda \hat{x}^{wc} + (1-\lambda)\hat{x}'$. Since $\alpha_u(\hat{x})$ is concave, $\alpha(\hat{x}^m) \geq \lambda\alpha(\hat{x}^{wc}) + (1-\lambda)\alpha(\hat{x}')$ which implies $\alpha(\hat{x}^m) \geq \alpha(\hat{x}^{wc})$ because $\alpha_u(\hat{x}^{wc}) = \alpha_u(\hat{x}')$. But we had assumed that $\alpha_u(\hat{x}^m) \neq \alpha_u(\hat{x}^{wc})$ so it must be that $\alpha(\hat{x}^m) > \alpha(\hat{x}^{wc})$. This is a contradiction as it violates the statement in the theorem that a maximum is achieved at one of the endpoints. Therefore, the assumption $\alpha_u(\hat{x}^{wc}) = \alpha_u(\hat{x}')$ ends up concluding a contradictory result and $\alpha_u(\hat{x})$ has a unique maximum. ■

The use of $\hat{x}^{wc}$ as the notation for the point where $\alpha_u(\hat{x})$ achieves its maximum was not coincidental. It is used to relate the result of Lemma 3.3 to Theorem 3.1 and Corollary 3.1. Note that there exist other functions than linear that are both concave and achieve their maximum at an endpoint of their interval domain. Examples are log(*y*) over [1,100], $-y^2$ over [0,1], etc. The following is Algorithm 3.1 for solving robust optimization problems with linear objective and constraint functions.

Algorithm 3.1 (Standard Benders Method):

Step 0: Set iteration counter (*it*) to 0. Pick a small positive constant for tolerance (*tol*).

Step 1: Set iteration counter (*it*) to *it* = *it* + 1. The original master problem will be:

$$
\begin{aligned}
&\max_{\alpha_u,\hat{x}} \alpha_u \\
&s.t. \\
&-\Delta x \le \hat{x} \le \Delta x \\
&\alpha_u \le \alpha_u^{\max}
\end{aligned}
\tag{3.13}
$$

The bounds on α_u are user-defined[15] depending on the problem. Solving the above problem gives $\alpha_u = \alpha_u^{it}$ and $\hat{x} = \hat{x}_{fixed}^{it}$.

Step 2: Fix the values of the complicating variables $\hat{x}$, and then solve the following subproblem as in the standard Benders decomposition method.

$$
\begin{aligned}
&w = \min_{x} \ f(x) \\
&s.t. \\
&g_j(x,\hat{x}) \le 0 \quad\quad j = 1,\ldots,J \\
&where \\
&\hat{x} = \hat{x}_{fixed}^{it} \quad\quad (\lambda^{it} \ dual)
\end{aligned}
\tag{3.14}
$$

---

[15] Further suggestions are available to achieve this upper bound in (Conejo et al., 2006). For Algorithm 3.1, a good value of the upper bound can be achieved by maximizing (as opposed to minimizing) the objective function in (3.6) over the entire space of decision and uncertainty variables. This is a definite upper bound to α_u as α_u is an auxiliary function to (3.6), given Assumption (3.2) holds.

Step 3: Check for convergence. Set $z_{sub} = w$ and $z_{mas} = \alpha_u^{it}$. If the difference $(|z_{sub} - z_{mas}|)/ z_{sub} \le tol$ then stop.

Step 4: Add a Benders cut to the master problem (3.13).

Step 1 (returned): Solve the following master problem after adding the Benders cut

$$\begin{aligned}
&\max_{\alpha_u, \hat{x}} \alpha_u \\
&s.t. \\
&-\Delta x \le \hat{x} \le \Delta x \\
&\alpha_u \le f(x_{sol}^{it}, \hat{x}_{sol}^{it}) + \lambda_{sol}^{it}((\hat{x})^T - (\hat{x}_{sol}^{it})^T) \\
&\alpha_u \le \alpha_u^{\max}
\end{aligned} \tag{3.15}$$

Return to Step 2 and proceed in this manner until convergence is met.

Theorem 3.4: If at the final iteration $\lambda_{sol}^{it} \ne 0$ for every element of λ_{sol}^{it}, Algorithm 3.1 converges to a globally optimal robust solution x^* of (3.6) and worst-case uncertainty value $\hat{x}^*$ in a finite number of steps.

Proof: By the theory of Benders decomposition for linear programs (Benders, 1962), Algorithm 3.1 converges to a maximum value for α_u in a finite number of steps. The algorithm also provides x^* and $\hat{x}^*$ such that $\alpha_u(\hat{x}^*) = f(x^*)$. If at the final iteration $\lambda_{sol}^{it} \ne 0$ for each element, then the function α_u approximated by the Benders cuts does not have zero slope between $\hat{x}^*$ and any other point in $[-\Delta x, \Delta x]$. Moreover, by

Lemma 3.3, α_u has a unique maximum point at $\hat{x}^*$. By Theorem 3.2, $\hat{x}^*$ is a worst-case uncertainty value, and fixing that value in (3.9) gives a globally optimal robust solution to (3.6).■

Note that if at the final iteration a cut is added where $\lambda_{it}^{sol} = 0$, then Algorithm 3.1 need not necessarily converge to a globally optimal robust solution because Theorem 3.2 requires this maximum point to be unique. If $\lambda_{it}^{sol} = 0$ at the final iteration, then there can be several points that maximize α_u, not all necessarily a worst-case uncertainty value. Algorithm 3.1 could then converge to a point that was not a robust point. In that case, the use of heuristic Algorithm 3.3 is needed.

3.3.2. Formulation of Approach: Solving Robust Optimization Problems with Quasiconvex Constraints

For convergence of the Benders decomposition algorithm, the function α_u needs to be concave. There is a larger class of functions than simply linear programs for which these conditions are valid. Indeed, for many engineering applications as well as numerical examples, local concavity of α_u can be sufficient (Conejo et al., 2006).

However, due to the worst-case analysis performed, α_u is quasiconvex as given by Corollary 3.1. Unfortunately standard Benders cuts cannot be used to approximate quasiconvex functions (Conejo et al., 2006). The reason is that optimal solutions may be omitted when cuts are added. Our advantage in a robust optimization setting with quasiconvex constraints as in problem (3.11) is that we only need good approximations to the functions at the endpoints. Approximations of the function α_u are not needed in between the endpoints, as the function attains its maximum at the endpoints. Figure 3.2 shows the idea behind these new cuts. The very top horizontal cut (labeled as "Cut 0") is an upper bound set for α_u as would normally occur in Benders decomposition. The numbers next to the cuts show the order of the cuts made in the iterative process. At iteration *it* a new modified Benders cut added to the master problem looks like the following.

$$\alpha_u \le f(x_{it}^{sol}, \hat{x}_{it}^{sol}) + \frac{f(x_{it}^{sol}, \hat{x}_{it}^{sol}) - f(x_{it-1}^{sol}, \hat{x}_{it-1}^{sol})}{(\hat{x}_{it}^{sol})^T - (\hat{x}_{it-1}^{sol})^T}((\hat{x})^T - (\hat{x}_{it}^{sol})^T) \tag{3.16}$$

This cut is one way to approximate the function around the endpoints and see which value of α_u at the endpoints is larger as shown in Figure 3.2.

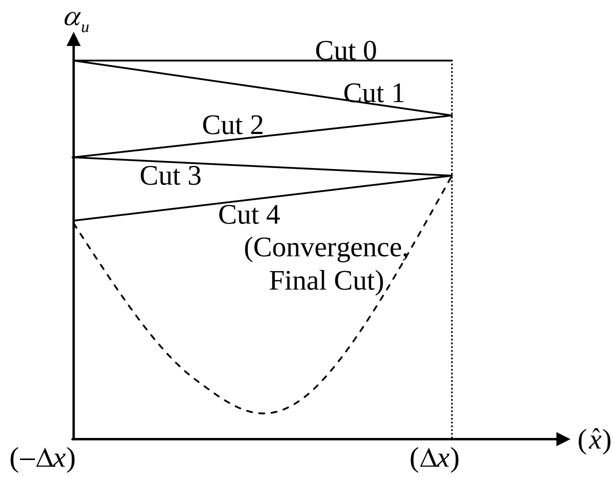

Figure 3.2: The Robust Benders Cuts[16] to Estimate the Maximum Endpoint of α_u

---

[16] The dashed line denotes the quasiconvex function α_u that is supposed to be approximated.

The following Algorithm 3.2 describes the method for quasiconvex constraints. Algorithm 3.2 differs from Algorithm 3.1 in that the modified Benders cuts described above are used.

Algorithm 3.2 (Heuristic Algorithm for Robust Optimization Problems with Quasiconvex Constraint Functions):

Step 0: Set iteration counter (*it*) to 0. Pick a small positive constant for tolerance (*tol*).

Step 1: Set iteration counter (*it*) to *it* = *it* + 1. The original master problem will be:

$$\begin{aligned}
&\max_{\alpha_u, \hat{x}} \alpha_u \\
&s.t. \\
&-\Delta x \le \hat{x} \le \Delta x \\
&\alpha_u \le \alpha_u^{\max} \\
&\text{Starting point } \alpha_u = \alpha_u^{\max}, \hat{x} = -\Delta x
\end{aligned} \tag{3.17}$$

The bounds on α_u are user-defined depending on the problem. A good value is an optimal objective function value for the non-robust nominal problem. Solving the above problem gives $\alpha_u = \alpha_u^{it}$ and $\hat{x} = \hat{x}_{fixed}^{it}$.

Step 2: Fix the values of the complicating variables $\hat{x}$, and then solve the following subproblem as in the standard Benders decomposition method.

$$
\begin{aligned}
& w = \min_{x} \ f_x(x) \\
& s.t. \\
& g_j(x, \hat{x}) \le 0 \quad\ j = 1, \ldots, J \\
& where \\
& \hat{x} = \hat{x}^{it}_{fixed}
\end{aligned}
\tag{3.18}
$$

<u>Step 3</u>: Check for convergence. Set $z_{sub} = w$ and $z_{mas} = \alpha_u^{it}$. If the difference $\left(\left|z_{sub} - z_{mas}\right|\right) / z_{sub} \le tol$ then stop.

<u>Step 4</u>: Add a modified Benders cut to the master problem. If this is any iteration greater than one, do not add an additional cut but just update the previous cut. To problem (3.17), add the modified Benders cut (3.16).

<u>Step 1 (returned)</u>: Solve the following master problem after adding the modified Benders cuts[17]

$$
\begin{aligned}
& \max_{\alpha_u, \hat{x}} \alpha_u \\
& s.t. \\
& -\Delta x \le \hat{x} \le \Delta x \\
& \alpha_u \le f(x_{it}^{sol}, \hat{x}_{it}^{sol}) + \frac{f(x_{it}^{sol}, \hat{x}_{it}^{sol}) - f(x_{it-1}^{sol}, \hat{x}_{it-1}^{sol})}{(\hat{x}_{it}^{sol})^T - (\hat{x}_{it-1}^{sol})^T} ((\hat{x})^T - (\hat{x}_{it}^{sol})^T) \\
& \alpha_u \le \alpha_u^{\max}
\end{aligned}
\tag{3.19}
$$

Return to Step 2 and proceed in this manner until convergence is met.

[17] Note, for the first iteration take $\hat{x}_{it-1}^{sol}$ equal to the value of Δx.

3.3.3. Formulation of Approach: Solving Robust Optimization Problems with Nonlinear Constraints

To extend the modified Benders decomposition method to general nonlinear constraints, the strategy will be to partition uncertainty intervals for uncertainty variables and attempt to find the interval-optimal points. Essentially, one can think of this as attempting to check the robustness of a candidate solution by partitioning the uncertainty interval by points and checking feasibility at each point. Clearly, if all constraints are feasible when fixed with the interval-optimal values for uncertainty elements, then the candidate solution is robust. Partitions can be selected depending on which type of constraint functions have uncertainty. In particular, quasiconvex constraint functions are simplest to consider.

Hence, under uncertainty in x, the maximum value of the function g lies on one of the endpoints of uncertainty when g is quasiconvex. So finding the interval-optimal point for quasiconvex constraint functions under uncertainty entails checking one of the two endpoints of uncertainty.

However, for nonlinear, not necessarily quasiconvex constraint functions, the interval-optimal values need not lie on the endpoints. For this, the constraint functions in the uncertainty interval range need to be checked at intermediary points to find the interval-optimal points. Figure 3.3 shows how checking further points helps. For quasiconvex constraint functions, Figure 3.3(a), only the endpoints need to be checked. For general nonlinear constraints, however, problems might be encountered if enough points within the uncertainty interval are not checked. This is shown in

Figure 3.3(b) where the method fails if enough points are not selected. Since quadratic constraints are symmetric, for concave quadratic constraints in particular (that are not quasiconvex) checking three points (endpoints plus central point of uncertainty interval, where $\hat{x} = 0$) is enough to guarantee a robust solution (in general true for all symmetric concave constraints), Figure 3.3(c). Selecting more points, Figure 3.3(d), solves this problem. Points maybe selected according to the accuracy desired for a locally optimal robust solution.

Checking additional points[18] entails adding additional constraints that have different uncertainty variables $\hat{x}_k \in [-\Delta x_k, \Delta x_k]$, k = 1,…,K, with uncertainty ranges that are subsets of the original uncertainty ranges, i.e. $[-\Delta x_k, \Delta x_k] \subset [-\Delta x, \Delta x]$ for all k = 1,…,K. In particular, the modified Benders method considers a uniform distribution of these points to be checked with $\Delta x_k = \Delta x / k$, k = 1,…,K. To check for center points, the constraints with $g_j(x,0) \le 0,\ j = 1,..,J$ need to be added. Note that this constraint is just the constraint g_j without any consideration for uncertainty. Since 0 is directly in the middle of the uncertainty interval, the constraint $g_j(x,0) \le 0,\ j = 1,..,J$ is simply the constraint with no uncertainty.

[18] Usually the number of additional points to be checked depends on the type of constraint functions as well as the accuracy desired. A good baseline for nonlinear constraint functions is to uniformly partition the uncertainty interval into enough additional points so that the distance between any two points is less than or equal to a preset tolerance for the problem. More points signify more accuracy.

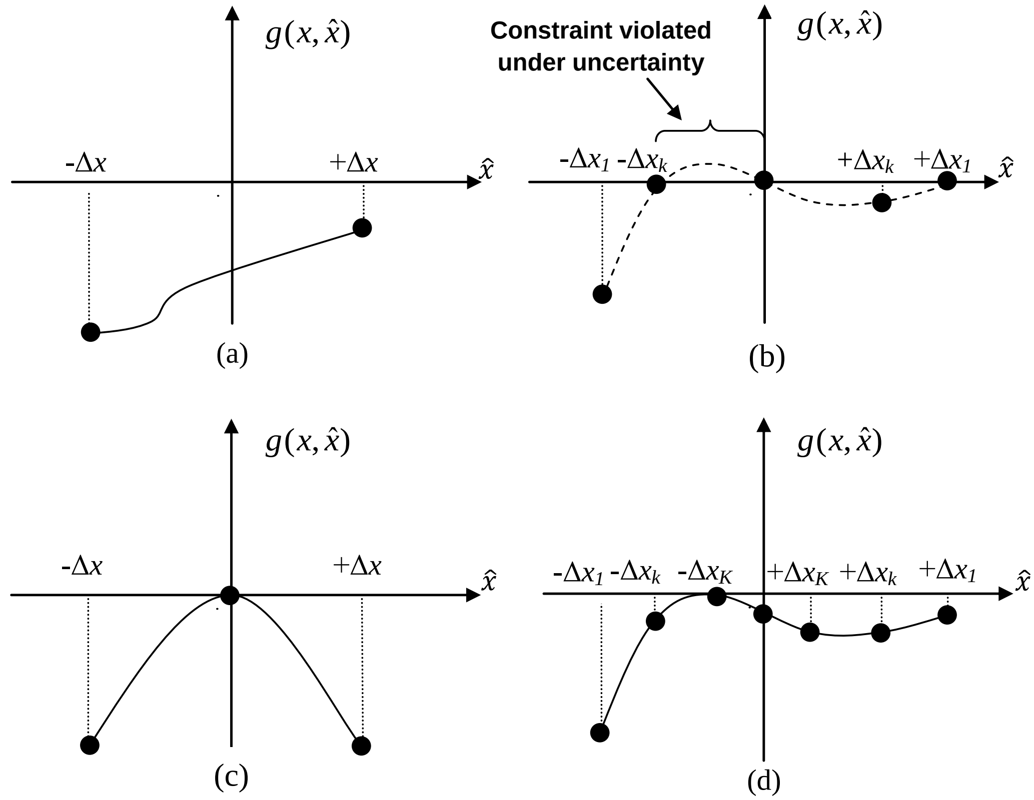

Figure 3.3: Checking Feasibility by Interval-Optimal Points for Constraints: (a) Quasiconvex: Successful Check by Endpoints; (b) Non-convex: Failed Check due to Insufficient Number of Points; (c) Symmetric Concave: Successful Check by Middle Point and Endpoints; (d) Non-convex: Successful Check by Sufficient Number of Intermediary Points

So, the formulation changes from (3.11) by adding further sample points (2*K* sample points).

$$
\begin{aligned}
&\min_x \; f(x) \\
&s.t. \\
&g_j(x,\hat{x}) \le 0 \quad j = 1,\dots,J \\
&g_j(x,0) \le 0 \quad j = 1,\dots,J \\
&g_j(x,\hat{x}_k) \le 0 \quad k = 1,\dots,K,\; j = 1,\dots,J \\
&x \in R^n, \hat{x} \in R \\
&\forall \hat{x} \in \{-\Delta x, \Delta x\}^{n_u} \\
&\forall \hat{x}_k \in \{-\Delta x_k, \Delta x_k\}^{n_u} \quad k = 1,\dots,K
\end{aligned}
\tag{3.20}
$$

Here, Δx_k, $k = 1,\ldots, K$ are the different sample points for the uncertainty interval as described by the black dots in Figure 3.3. Note that $0 \le \Delta x_k \le \Delta x$. The uncertainty elements (with a superscript ^) only take on two values (per element of vector) each. Since all the extra constraints from (3.20) when compared to (3.11) can be incorporated into the constraints in (3.11), from now on the rest of this chapter will assume that (3.11) has had a thorough sample of points such that within each interval of uncertainty, the constraint functions are quasiconvex.

To speed up computation of the proposed approach, gradient-based optimization algorithms (Bazaraa et al., 1993) are used as opposed to population-based optimization ones (such as Genetic Algorithms or Simulated Annealing) (Davis, 1987). In particular, the nonlinear solvers CONOPT in GAMS (GAMS, 2010) which are gradient-based were used for all test problems except the last one on heat exchanger design. Since the code for the heat exchanger design problem was already available and coded in MATLAB, *fmincon* (MATLAB, 2008) was used for that particular problem.

The following describes the modified Benders decomposition algorithm where the constraint functions are quasiconvex within each interval of uncertainty.

Algorithm 3.3 (Heuristic Algorithm for Robust Optimization Problems with Nonlinear Constraint Functions):

<u>Step 0:</u> Proceed exactly as Algorithm 3.2 for problem (3.20)

Step 1: If at any iteration, $\frac{f(x_{it}^{sol},\hat{x}_{it}^{sol}) - f(x_{it-1}^{sol},\hat{x}_{it-1}^{sol})}{(\hat{x}_{it}^{sol})^T - (\hat{x}_{it-1}^{sol})^T} = 0$, add more[19] sample points in (3.20) by making a uniform partition. Return to Step 0.

Note that, theoretically, convergence is not guaranteed in Algorithm 3.3. But the numerical results suggest that Algorithm 3.3 is applicable to a wide variety of problems. Now some computational costs based on numerical evidence will be provided. Consider a single-objective, robust optimization problem with V variables, J constraints, P parameters, and N uncertainty variables. One function call[20] is defined as any instance where the solver calls an objective function, constraint, or other value or assignment in the optimization problem. Table 3.1 gives the maximum number of function calls possible through one iteration of the modified Benders method. This analysis is only based on numerical evidence that the method finds a locally optimal robust solution through Algorithm 3.3.

---

[19] In practice, double the number of points as was done for the numerical examples in this dissertation.

[20] Due to difference in software, the GAMS method of function calls was used. Therefore, function calls for both deterministic and robust cases are provided so the reader can compare solutions. Computational times have also been provided. For a discussion on function calls, please refer to Appendix B.

Table 3.1: Analysis of function calls for one iteration

Operation	Number of Assignments	Function Calls
Objective Function	1	1
Iteration Counter	1	1
Constraints	*J+1* *(Extra 1 for obj. robust*)	*N(J+1)*
Fixing Uncertainty at Lower Level	*N*	*N*
Slope of Modified Benders Cut	*N*	*N(J+1)*
Sample Points for Nonlinear	*2K*	*2KN + 4KN-2N*
Total Maximum Expected	-	*2 + N + JN + 6KN*

For most of the numerical problems in this chapter, *K* is at most 10. Hence, the number of binary variables introduced to the formulation is *NJK*. Note that this is the maximum theoretically possible function calls and actual function calls (as shown by examples) are much less. For example, in the heat exchanger problem (Section 3.5.3), N = 8, J = 17, K = 10 and 6 iterations were used to solve the problem so the maximum expected number of function calls is 6(2 + 8 + (17)(8) + 6(10)(8)) = 3,756. According to the results, it took only 984 to solve this particular problem.

3.4. Numerical Results

The following numerical and engineering examples serve to demonstrate applicability of the algorithm, compare the proposed algorithm to a previous algorithm from the literature, and show the different types of problems that can be solved. The first example is a simple quadratic program to show the algorithm steps in detail. The next three examples then increase the complexity (number of variables, nonlinearity) of this quadratic program and show how the number of function calls changes. The next two examples are similar linear programs except that Example 6 has been shown to need a significantly higher number of function calls to solve for the locally robust optimal solution than Example 5 (Li et al., 2011). Examples 7 and 8 are robust optimization problems with quasiconvex constraints which the modified Benders method is shown to solve exactly. The next four problems are scalable versions of an engineering example with quasiconvex constraints. These examples show that the modified Benders method is scalable and can be applied to large problems without a drastic increase in function calls. The final two examples are from engineering design and are nonlinear (non-convex) programs. Of the two engineering examples, the first one (Welded Beam Design) considers objective robustness and the second (Heat Exchanger Design) considers feasibility robustness. All optimization problems correspond to minimizing a single objective function with a set of constraints. Problems labeled as "self" have been designed by the author to use as test problems (Siddiqui et al., 2011a); detailed formulations as well as further characteristics of the solution are in Appendix A. Solutions were checked by a simple uniform discretization of the uncertainty range (each point separated by 0.01) to see if any of the constraints were violated under uncertainty. Tolerance (*tol*) was set to 0.00001 for all examples.

3.4.1. Numerical Example (Example 1) to Show Methodology Step-by-Step

A simple numerical example is presented to show how Benders algorithm is modified to obtain robust solutions using heuristic Algorithm 3.2. In the following robust problem, uncertainty is only in the constraint (without loss of generality).

$$\begin{aligned}
&\min_x \left((x_1-0.6)^2+(x_2-0.6)^2\right)\\
&s.t.\\
&(-1+\hat{x})+x_1+x_2\le 0\\
&-x_1\le 0\\
&-x_2\le 0\\
&where\\
&\forall \hat{x}\in[-\Delta x,\Delta x]
\end{aligned} \tag{3.21}$$

The robust solution to this problem (verified as unique algebraically) is $x_1=0.45, x_2=0.45$ with the globally interval-optimal value $\hat{x}=0.1$. Since the constraint functions are linear, hence quasiconvex within the uncertainty interval, reformulate as in (3.11) to have uncertainty variables only have endpoint values.

$$\begin{aligned}
&\min_x \left((x_1-0.6)^2+(x_2-0.6)^2\right)\\
&s.t.\\
&(-1+\hat{x})+x_1+x_2\le 0\\
&-x_1\le 0\\
&-x_2\le 0\\
&where\\
&\forall \hat{x}\in\{-\Delta x,\Delta x\}
\end{aligned} \tag{3.22}$$

Proceed according to the modified Benders method described in Section 3.3.2.

Step 1: The master problem is the following:

$$\begin{aligned}
&\max_{\alpha,\hat{x}} \alpha \\
&0 \le \hat{x} \le 1 \\
&\alpha \le 1000
\end{aligned} \tag{3.23}$$

The upper bound on α is chosen to be large enough to not interfere with given the form of (3.22). Solving the above problem gives $\alpha = 1000$ and $\hat{x}$ = -0.1. Here

$$\begin{aligned}
&\alpha = \min_{x}\left((x_1 - 0.6)^2 + (x_2 - 0.6)^2\right) \\
&s.t. \\
&(-1+\hat{x}) + x_1 + x_2 \le 0 \\
&-x_1 \le 0 \\
&-x_2 \le 0 \\
&\hat{x} \in \{-0.1, 0.1\}
\end{aligned} \tag{3.24}$$

Step 2: Fix $\hat{x}$ and then solve the following subproblem:

$$\begin{aligned}
&w = \min_{x}\left((x_1 - 0.6)^2 + (x_2 - 0.6)^2\right) \\
&s.t. \\
&(-1+\hat{x}) + x_1 + x_2 \le 0 \\
&-x_1 \le 0 \\
&-x_2 \le 0 \\
&\hat{x} = -0.1
\end{aligned} \tag{3.25}$$

This gives: $x_1 = 0.55, x_2 = 0.55$.

Step 3: Check for convergence with $z_{up} - z_{low} = 1000 - 0.005 = 999.995$ where $z_{low} = \left((0.55-0.6)^2 + (0.55-0.6)^2\right)$ and $z_{up} = \alpha$. Since this is not good enough for convergence when compared with the preselected tolerance, a modified (robust) Benders cut is added.

Step 4: Add the following *robust Benders cut*.

$$0.005 + \frac{1000 - 0.005}{-0.1 - (0.1)} (\hat{x} - (-0.1)) \geq \alpha \tag{3.26}$$

Step 1 (returned): Solve the following master problem after adding the robust Benders cut:

$$\begin{aligned}
&\min_{\hat{x}} \alpha \\
&s.t. \\
&-0.1 \leq \hat{x} \leq 0.1 \\
&\alpha \leq 1000 \\
&0.005 + \frac{1000 - 0.005}{-0.1 - (0.1)} (\hat{x} - (-0.1)) \geq \alpha
\end{aligned} \tag{3.27}$$

Solving the above problem gives $\alpha = 1000$ and $\hat{x}$ = 0.1. Then go back to Step 2 and solve the subproblem with $\hat{x}$ = 0.1 fixed. A new modified Benders cut will now be added.

The following graph (Figure 3.4) shows what happens when this cut is added. The standard Benders decomposition method would have taken a cut that would have forced $\hat{x}$ = -0.1 and that would have given the constraint with the dashed line. However, the robust Benders cut generates the cut signified by the dotted line, which is in fact the constraint that forms the border of the robust feasible region.

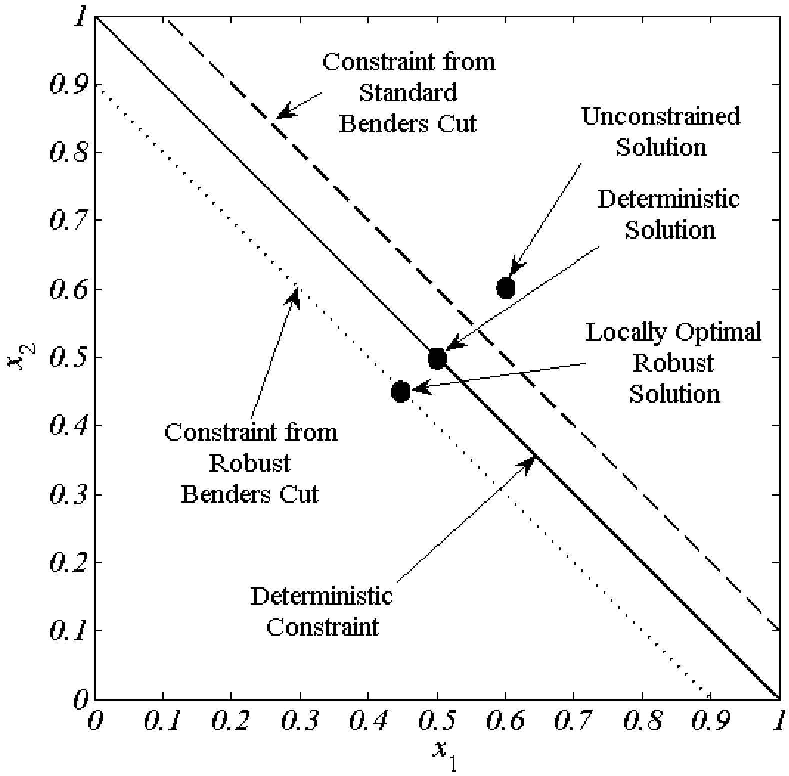

Figure 3.4: Adding a modified (Robust) Benders Cut

The algorithm proceeds in this manner until convergence. Table 3.2 summarizes these results.

Table 3.2: Solution Steps for Modified Benders Approach

Iteration	$\hat{x}$	x_1	x_2	z_{low}	z_{up}
1	-0.1	0.55	0.55	0.005	1000
2	0.1	0.45	0.45	0.045	1000
3	0.1	0.45	0.45	0.045	0.045

This simple problem was solved in three iterations. The final row corresponds to the globally-optimal robust solution, which can be verified algebraically to be globally robust optimal. The details are shown in the following table (Table 3.3).

Table 3.3: Detailed Solution for Simple Problem

Information	Nominal Solution	Robust Solution
x_1	0.5	0.45
x_2	0.5	0.45
$f(x)$	0.02	0.045
Function Calls	5	11

3.4.2. Numerical Results

Table 3.4 describes the results obtained from the numerical test problems. The first two examples have only uncertainty in the parameters while the rest have uncertainty in the parameters and the decision variables.

The same test problems were solved using Li et al.'s (Li et al., 2006) method for robust optimization and the results are displayed in Table 3.4. Not only does Li et al.'s (Li et al., 2006) method use a lot of function calls, but often the robust function value is higher than the modified Benders value. Once the number of function calls exceeded 10^9, the run was stopped. These solutions have not been reported in Table 3.4 as well as in the rest of the chapter as they were infeasible in the solver (which can imply that the problem was too computationally intensive for the solver).

Table 3.4: Description of Test Problems

Source	Determ. Optimal Function Value	Robust Optimal Function Value	Li et al. (2006) Function Value	# of Function Calls (Determ.)	# of Function Calls (Robust)	Li et al. (2006) Function Calls
Example 1 (Self)	0.02	0.045	0.045	5	11	540
Example 2 (Self)	9.02	9.145	9.268	7	19	2,592
Example 3 (Self)	9.02	9.145	9.268	7	21	2,808
Example 4 (Self)	9.77	9.885	9.920	7	21	2,916
Example 5 (Self)	-23.00	-21.50	-20.75	5	17	7,856
Example 6 (Self)	-31.21	-29.79	-28.36	5	17	11,099
Hock 100 (Hock, 1980)	680.6	692.4	-	7	19	$>10^9$
Hock 106 (Hock, 1980)	7049	7219	-	5	17	$>10^9$

3.5. Engineering Design and Other Applications

3.5.1. Fleury's Weight Minimization

This is a modified example from the literature (Groenwold & Etman, 2010) so that interval uncertainty is present in all the decision variables. This example supports the approach for feasibility robustness as well as corroborates the fact that the modified Benders method is able to tackle problems with large number of variables and constraints without being computationally expensive. For *N* variables, the problem is as follows:

$$
\begin{aligned}
&\min_{x} \; f(x) = \sum_{i=1}^{N} x_i \\
&s.t. \\
&\sum_{i=1}^{0.95N} \frac{1}{x_i + \hat{x}_i} + \frac{1}{N^2} \sum_{i=0.95N+1}^{N} \frac{1}{x_i + \hat{x}_i} - N \le 0 \\
&\sum_{i=1}^{0.95N} \frac{1}{x_i + \hat{x}_i} - \frac{1}{N^2} \sum_{i=0.95N+1}^{N} \frac{1}{x_i + \hat{x}_i} - 0.9N \le 0 \\
&\frac{1}{N^2} \le x_i + \hat{x}_i \le N^2 \quad i = 1,2,\ldots,N \\
&\hat{x}_i \in \left[-0.1,+0.1\right]
\end{aligned}
\tag{3.28}
$$

The modified Benders method solved this problem with the results shown in Table 3.5 for N = 10^2, 10^3, 10^4, and 10^5. Note that the number of function calls increases linearly with the complexity of the problem. *N* represents the number of variables in the problem and all of them have uncertainty. Again, this example was compared to (Li et al., 2006) as shown in Table 3.5. However, the results for (Li et al., 2006) are not reported as the problem was stopped after a certain number of function calls given in Table 3.5. Here, Li et al.'s (Li et al., 2006) method could, conceptually, solve this problem but would have taken a lot of computation time.

However, the modified Benders method solved all cases and produced only a linear increase in computational effort.

Table 3.5: Results for Fleury's Weight Minimization Like Problem

Number of Variables	$N = 10^2$	$N = 10^3$	$N = 10^4$	$N = 10^5$
Tolerance	10^{-6}	10^{-8}	10^{-10}	10^{-12}
x_1 to $x_{0.95N}$ (Determ.)	1	1	1	1
x_1 to $x_{0.95N}$ (Robust)	1.1556	1.1556	1.1556	1.1556
$x_{0.95N+1}$ to x_N (Determ.)	10^{-2}	10^{-3}	10^{-4}	10^{-5}
$x_{0.95N+1}$ to x_N (Robust)	$0.1+10^{-2}$	$0.1+10^{-3}$	$0.1+10^{-4}$	$0.1+10^{-5}$
Function Value (Determ.)	95.00005	950.0005	9500.005	95000.05
Function Value (Robust)	110.2820	1102.820	11028.20	110282.0
Function Calls (Determ.)	506	2.0×10^3	2.0×10^4	2.0×10^5
Function Calls (Robust)	744	2.3×10^4	1.9×10^5	1.9×10^6
Fn. Calls (Li et al., 2006)	$>10^9$	$>10^9$	$>10^{12}$	$>10^{12}$

3.5.2. Design of a Welded Beam

This example is a well-known welded beam problem from (Ragsdell & Phillips, 1976). In this problem, a beam A is to be welded to a rigid support member B. The beam has a rectangular cross-section and is to be made out of steel. The beam is designed to support a force F = 6000 LBF acting at the tip of the beam, and there are constraints on the shear stress, normal stress, deflection, and buckling load on the beam. The problem has four continuous design variables, and they are: thickness of the weld (h), length of the weld (l), thickness of the beam (t), and width of the beam (*b*). All variables are in inches. The objective of the problem is to minimize the total cost *f*(*x*) of making such an assembly. For complete formulation of the robust optimization problem including specific values of the parameters, please refer to (Gunawan & Azarm, 2004). Figure 3.5 shows the structure of the beam.

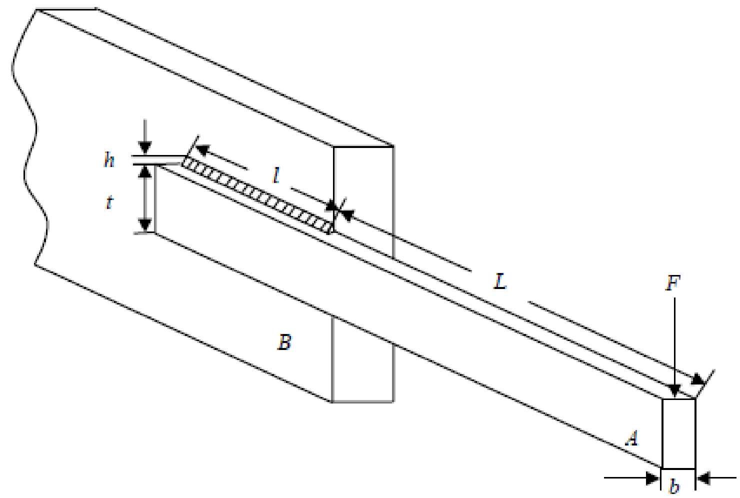

Figure 3.5: Design of a Welded Beam (Gunawan & Azarm, 2004)

The following is the formulation for the welded beam as outlined in (Gunawan & Azarm, 2004). The objective function is given by

$$\min\ f_{\text{cost}} = (1 + c_3)h^2 l + c_4 tb(L + l) \tag{3.29}$$

The constraints and other equations are described below.

$$\left|\left((1+\tilde{c}_3)h^2 l + c_4 tb(L+l) + 0.2\tilde{c}_5 l\right) - \left((1+c_3)h^2 l + c_4 tb(L+l)\right)\right| \le 0.3$$

$$g_1 \equiv \frac{\tau}{\tau_d} - 1 \le 0 \qquad g_2 \equiv \frac{\sigma}{\sigma_d} - 1 \le 0$$

$$g_3 \equiv \frac{\delta}{0.25} - 1 \le 0 \qquad g_4 \equiv \frac{F}{P_c} - 1 \le 0$$

$$g_5 \equiv \frac{h}{b} - 1 \le 0 \qquad g_2 \equiv \frac{0.125}{h} - 1 \le 0$$

$$0.1 \le h \le 2.0 \qquad 0.1 \le l \le 2.0$$

$$0.1 \le t \le 2.0 \qquad 0.1 \le b \le 2.0$$

$$\tilde{c}_3 \in [0.0547, 0.1547]$$

$$\tilde{c}_5 \in [0, 0.25]$$

where

c_3 = cost of weld material (\$0.1047/inch^3)

c_4 = cost of weld material (\$0.0481/inch^3)

τ = maximum shear stress in weld (psi)

τ_d = allowable shear stress in weld (13,600 psi)

σ = maximum normal stress in beam (psi)

σ_d = allowable normal stress in beam (30,000 psi)

δ = deflection at beam end (inch)

P_c = allowable buckling load (LBF)

L = Length of unwelded beam (14 inch)

$G = 12 \times 10^6$ psi

$E = 30 \times 10^6$ psi

The following equations are used to calculate the above variables.

$$\tau = \sqrt{(\tau')^2 + 2\tau'\tau''\cos\theta + (\tau'')^2} \tag{3.30}$$

$$\tau' = \frac{F}{\sqrt{2}hl}; \quad \tau'' = \frac{MR}{J}; \quad \cos\theta = \frac{l}{2R}; \tag{3.31}$$

$$M = F\left(L + \frac{l}{2}\right); \quad R = \sqrt{\frac{l^2}{4} + \left(\frac{h+t}{2}\right)^2} \tag{3.32}$$

$$J = 2\left\{0.707hl\left(\frac{l^2}{12} + \left(\frac{h+t}{2}\right)^2\right)\right\} \tag{3.33}$$

$$\sigma = \frac{6FL}{bt^2}; \quad \delta = \frac{4FL^3}{Et^3b} \tag{3.34}$$

$$P_c = \left(\frac{4.013\sqrt{EI\alpha}}{L^2}\right)\left(1 - \frac{t}{2L}\sqrt{\frac{EI}{\alpha}}\right) \tag{3.35}$$

$$I = \frac{1}{12}tb^2; \quad \alpha = \frac{1}{3}Gtb^3 \tag{3.36}$$

The solution from the modified Benders method is different from (Gunawan & Azarm, 2004). First, the modified Benders method's nominal solution is closer to an actual solution from an earlier paper by (Ragsdell & Phillips, 1976) who provided an optimal objective function value of f = 2.38 while (Gunawan & Azarm, 2004) provided f = 2.39. Second, the robust solution is also lower in function value but still feasible. The robust solution is also feasible for all realizations of uncertainty, hence it is better than (Gunawan & Azarm, 2004) reported solution.

This example highlights the strength of the modified Benders method over previous methods. Gunawan and Azarm's (Gunawan & Azarm, 2004) method involves a backward mapping approach, which is known to omit solutions. The modified Benders method, while giving a better solution (lower in function value), is also computationally less expensive. The solution from the modified Benders method was checked for robustness using a genetic algorithm. The solution is displayed in Table 3.6.

Table 3.6: Results of Welded Beam Example

Info	Gunawan and Azarm (2004) Nominal Sol.	Modified Benders Nominal Sol.	Gunawan and Azarm (2004) Robust Sol.	Modified Benders Robust Sol.
h	0.241	0.2444	0.246	0.2392
l	6.158	6.2186	5.461	5.6753
t	8.5	8.2915	9.138	9.1225
b	0.243	0.2444	0.248	0.2392
$f(x)$	2.39	2.3807	2.48	2.4236
Function Calls	N/A	8	250	38

3.5.3. Heat Exchanger Design

The energy balance on a heat exchanger can be written as

$$Q = UAF\Delta T_m = (mc_p)_h(T_{h1} - T_{h2}) = (mc_p)_c(T_{c2} - T_{c1}) \tag{3.37}$$

Several equations govern the above heat transfer. The above equation (3.37) will be used as an objective function that is to be maximized, as well as constraints that restrict the structure, in particular constraints on the pressure drop on the tube side (Δp_t) and shell side (Δp_s). Subscript 1 denotes the fluid entering while subscript 2 denotes it leaving; c denotes the cold fluid and h the hot fluid. In this example, cold water is in the tubes and hot water is on the shell side and the problem has been set up in a counterflow arrangement for a 124 tubes and two-pass heat exchanger. The following lists the important variables and parameters considered in this design.

Symbol	Units	Description	Value
c_p	J/(kg K)	Specific heat at constant pressure	Variable
d_0	m	Tube outside diameter	Variable
f		Tube flow friction factor	Variable
f_s		Friction factor shell side	Variable
h_0	W/(m^2 K)	Heat-transfer coefficient outside tube	Variable
h_i	W/(m^2 K)	Heat-transfer coefficient inside tube	Variable
k	W/(m K)	Thermal conductivity of fluids	Variable
Δp_s	Pa	Shell-side pressure drop	Variable
Δp_t	Pa	Tube-side pressure drop	Variable
u_t	M/s	Mean axial velocity of fluid in tube	Variable
A_0	m^2	Tube outside surface area per pass	Variable
A_i	m^2	Tube inside surface	Variable
A_s	m^2	Cross-flow area at or near shell centerline	Variable
A_t	m^2	Total cross-sectional area of tubes per pass	Variable
B	m	Baffle spacing	0.5
C	m	Clearance between adjacent tubes	Variable
C_L		Tube layout constant	1 (for 90^0)
C_{TP}		Tube count calculation constant	0.90
D_e	m	Equivalent diameter of shell	Variable
F		LMTD correction factor	Variable
L	m	Tube length	Variable
N_b		Number of Baffles (Integer = B/L)	4
N_T		Number of Tubes	124
N_p		Number of Tube passes	2
P_P	W	Pumping power of fluid in tubes	Variable
Pr		Prandtl Number	Variable
Q	W	Heat-transfer rate	Variable
R_{fo}	(m^2 K)/W	Fouling resistance on outside of tube	0.00015
R_{fi}	(m^2 K)/W	Fouling resistance on inside of tube	0.00015
Re_b		Reynolds number at T_b	Variable
Re_s		Shell-side Reynolds number at T_b	Variable
ΔT_m	K	LMTD	Variable
T_{h2}	K	Outlet temperature of hot fluid	Variable
T_{c2}	K	Outlet temperature of cold fluid	315
T_b	K	Bulk temperature	Variable
T_w	K	Wall temperature	Variable
U	W/(m^2 K)	Average overall heat transfer coefficient based on A	Variable
φ_s		Viscosity correction factor	Variable
μ	kg/(s m)	Dynamic Viscosity	Variable
μ_b	kg/(s m)	Dynamic Viscosity at T_b	Variable
μ_w	kg/(s m)	Dynamic Viscosity at T_w	Variable
ρ	kg/m^3	Density	Variable

The following equations are the ones coded into MATLAB and are selected from the whole formulation to provide further insight. For the complete formulation and all equations used, please refer to (Magrab et al., 2004).

$$T_{h2} = T_{h1} - \frac{(\dot{m}c_p)_t}{(\dot{m}c_p)_s}\left(T_{c2} - T_{c1}\right) \tag{3.38}$$

$$\Delta p_t = C_0 \frac{f\dot{m}_t^{\ 3}}{UF\Delta T_m} \tag{3.39}$$

$$\Delta p_s = \frac{f_s \dot{m}_s^{\ 2}(N_b + 1)D_s}{2A_s^{\ 2}\rho D_e \phi_s} \tag{3.40}$$

$$L = \frac{(\dot{m}c_p)_t\left(T_{c2} - T_{c1}\right)}{\pi d_0 N_T F\Delta T_m} \tag{3.41}$$

$$C_0 = \frac{N_p (c_p)_t\left(T_{c2} - T_{c1}\right)}{2\pi d_i d_0 N_T \rho_t A_t^{\ 2}} \tag{3.42}$$

$$A_t = \frac{\pi d_i^{\ 2} N_T}{4N_P} \tag{3.43}$$

Table 3.7 lists the design variables and parameters with uncertainty.

Table 3.7: Design Variables and Parameters with Uncertainty

Symbol	Unit	Description	Uncertainty
d_i	m	Tube inside diameter (Variable)	$d_i \pm 0.001$
m_t	kg/s	Tube-side mass flow rate (Variable)	$m_t \pm 1$
m_s	kg/s	Shell-side mass flow rate (Variable)	$m_s \pm 1$
D_s	m	Shell inside diameter (Variable)	$D_s \pm 0.01$
P_T	m	Pitch size (Variable)	$P_T \pm 0.01$
T_{h1}	K	Inlet temperature of hot fluid	65 ± 1
T_{c1}	K	Inlet temperature of cold fluid	18 ± 1
ktube	W/(m K)	Thermal conductivity of tubes	60 ± 1

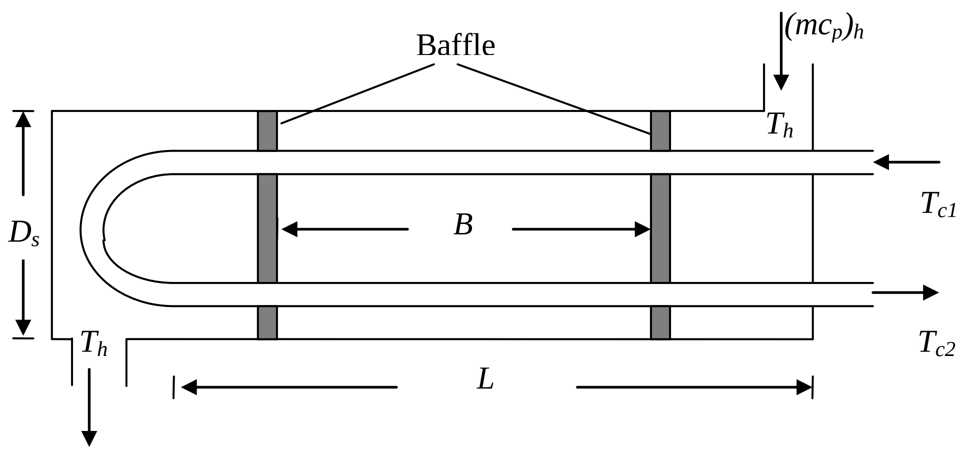

Figure 3.6: Heat Exchanger Schematic (Magrab et al., 2004)

The optimization problem is the following:

$$
\begin{aligned}
&\max Q \\
&s.t. \\
&\Delta p_s \le 35000 \\
&\Delta p_t \le 50000 \\
&T_{h2} \le 60 \\
&N_P A_T \le A_S \\
&-C \le 0 \\
&L \le 40 \\
&7 \le m_t + \hat{m}_t \le 10 \\
&13 \le m_s + \hat{m}_s \le 16 \\
&0.001 \le D_s + \hat{D}_s \le 2 \\
&0.002 \le P_T + \hat{P}_T \le 0.1 \\
&0.0001 \le d_i + \hat{d}_i \le 0.1
\end{aligned} \tag{3.44}
$$

The following table shows the results.

Table 3.8: Results for Heat Exchanger Design

Variables	Nominal Solution	Robust Solution
Q	1006.77	906.09
d_i	0.0160	0.0149
m_t	10	9
m_s	14	14
D_s	0.3900	0.3900
P_T	0.0240	0.0311
Function Calls	49	984

The thing to note about this example is that with less than an average of 1% uncertainty, the objective function value decreases by almost 10%. Hence, in the design of any model, it is important to consider the uncertainty in the problem, which can lead to different designs as well. This problem was also tried with Li et al.'s (2006) method however after 10^9 function calls without convergence, the approach was stopped.

For completeness, Table 3.9 displays actual computational time for each test problem as well.

Table 3.9: Number of Iterations and CPU Time to Solve Problems

Test Problem	Number of Iterations	CPU (2.0 GHz, 4GB RAM) Time (s)
Example 1	3	0.560
Example 2	3	0.787
Example 3	3	1.654
Example 4	3	1.435
Example 5	3	3.821
Example 6	3	3.494
Hock 100	4	30.552
Hock 106	4	25.645
Fleury ($N=10^2$)	6	9.328
Fleury ($N=10^3$)	9	324.532
Fleury ($N=10^4$)	12	886.321
Fleury ($N=10^5$)	15	2123.453
Welded Beam	4	1.606
Heat Exchanger	6	45.234

3.5.4. Building Energy Intensive Infrastructure

This example takes a problem of a decision maker to decide whether to build energy intensive infrastructure at intensity *H* when there is uncertainty in future carbon tax and retrofit cost. This example is a modified version of the formulation from the paper (Strand et al., 2011). Investments in large, long-lasting, energy-intensive infrastructure that use fossil fuels increase longer-term energy use and greenhouse gas emissions, unless the plant is shut down early or undergoes costly retrofit later. These investments will depend on expectations of retrofit costs and future energy costs, including energy cost increases from tighter controls on carbon emissions.

Consider a decision maker in a world with two periods. Infrastructure investment is made at the start of period 1, and can be "retrofitted" at the start of period 2. As long as it is operated and not retrofitted, a given infrastructure gives rise to a given energy consumption per unit of time, determined at the time of initial investment. Energy supply costs and environmental/climate-related costs are uncertain at the time of establishment in period 1, but are revealed at the start of period 2. Assume both periods have the same length, and there is no discounting within the periods. The problem of a decision maker is given by

$$\begin{aligned}
&\max_{H,\alpha} U(H) - t_1 H + (U(H) - t_2(1-\alpha)H - r\alpha H) \\
&s.t. \\
&0 \le \alpha \le 1 \\
&0 \le H \le H^{\max} \\
&t_2(1-\alpha)H + r\alpha H \le U(H) - t_1 H
\end{aligned} \tag{3.45}$$

Here, *U*(*H*) is the utility of the decision maker when selecting an energy investment intensity *H*. The costs (carbon tax, for example) in the first period for this energy are t_1 and in the second period are t_2. The ratio α is the amount of energy investment that is retrofitted, and r is the cost of that retrofit. In (3.45), the decision maker aims to maximize utility in the two periods (without any discounting).

As in (Strand et al., 2011), the uncertainty is present in the values of t_2 and r. The last constraint in the above formulation makes sure that the cost for retrofitting and paying carbon tax in the second period is below the excess utility achieved in the first period. For the numerical study, the following parameters were chosen.

- $U(H) = 8H - H^2$
- $H^{\max} = 4$
- $t_1 = 1$
- $t_2 \in [4 - \Delta t_2, 4 + \Delta t_2]$
- $r \in [5.5, 6.5]$

Note that in this first case, we have assumed that t_2 has uncertainty of magnitude $\Delta\, t_2$ while uncertainty in retrofit cost r is given as above. The goal is to see what happens as this uncertainty range is increased. The robust optimization problem to solve is

$$\begin{aligned}
&\max_{H,\alpha} U(H) - t_1 H + (U(H) - t_2(1-\alpha)H - r\alpha H) \\
&s.t. \\
&0 \le \alpha \le 1 \\
&0 \le H \le H^{\max} \\
&\tilde{t}_2(1-\alpha)H + \tilde{r}\alpha H \le U(H) - t_1 H \\
&\forall \tilde{t}_2 \in [4 - \Delta t_2, 4 + \Delta t_2] \\
&\forall \tilde{r} \in [5.5, 6.5]
\end{aligned} \tag{3.46}$$

The modified Benders method is applied to solve this problem. Table 3.10 shows what happens to energy intensity H and selection of retrofit vs. not retrofit as Δt_2 increases.

Table 3.10: Results for Increasing Uncertainty in t_2

Value of Δt_2	Energy Intensity (H)	Retrofit? (Value of α)
0	3	No ($\alpha = 0$)
1	2	No ($\alpha = 0$)
2	1	No ($\alpha = 0$)
3	0.5	Yes ($\alpha = 1$)
4	0.5	Yes ($\alpha = 1$)

Note that with increasing uncertainty, the decision maker chooses less energy intensive infrastructure. This is at odds with the probability-based analysis done in (Strand et al., 2011). Since (Strand et al., 2011) assumed probability distributions for t_2 and r, an increase in uncertainty meant a good chance that t_2 would offer a low tax in the future as well. Hence, an increase in uncertainty brought an increase in energy intensive investment. One of the main reasons this answer is different is that robust optimization considers a worst-case analysis. Hence, with increasing uncertainty, the extremely risk-averse robust optimizer chooses a progressively safer option, to avoid any chance of not being able to afford retrofit or tax in the future. Hence, robust optimization gives us an alternative way to analyze this problem.

3.6. Summary

This chapter presents an efficient robust optimization approach to solve problems that have parameters and/or decision variables with interval uncertainty. The proposed modified Benders method obtains robust optimal solutions to linear programming, quadratic programming, convex and non-convex programming problems. The approach is computationally tractable and is tested with 14 numerical and engineering examples with the most general being nonlinear (non-convex) objective function and nonlinear (non-convex) constraint robust optimization problems. The modified Benders method provides an approximate locally optimal robust solution to general nonlinear robust optimization problems, with a way to improve this approximation if desired.

The test examples show the strength of this method when compared to two previous approaches. Not only is the method computationally efficient, but also obtains better solutions when compared to these previous methods. The method is scalable, that is, number of function calls increases at most linearly with an increase in number of variables for the problems tested.

Chapter 4: Solving Mathematical Programs and Equilibrium Programs with Equilibrium Constraints

4.1. Introduction

This chapter describes a new algorithm to solve mathematical programs and equilibrium programs with equilibrium constraints. Numerical examples are provided in each case, along with a test of computational time with disjunctive constraints. An application of the method to a large-scale North American gas market model is also provided.

The motivation behind developing an algorithm for MPECs was to find an alternative to traditional techniques, in particular disjunctive constraints. This was necessitated by the need to solve large-scale MPECs representing natural gas markets. A North American gas model was developed from the larger World Gas Model (Gabriel et al., 2011c). Various techniques[21] were employed to solve this North American gas model, but the only successful one was the application of Algorithm 4.1 presented later in this chapter.

Hence, for this research thrust, the application drove the theory. In this chapter, the current state of the theory is presented. As these ideas are still being

[21] The fine tuning of solvers was needed to be able to solve the North American gas model. In particular, the SBB solver (GAMS, 2010) was used in conjunction with CONOPT (GAMS, 2010). The iteration limits for the first search of SBB needed to be increased. For particular sections of the branch and bound tree, a breadth-first approach was employed.

developed, there is great room for further development, including the EPEC solution techniques.

The chapter starts with a general theory of the new MPEC and EPEC technique followed by examples. Finally, an application of the technique is presented for the North American gas market model. A portion of this chapter has been presented in (Siddiqui & Gabriel, 2011b) and (Gabriel et al., 2011c).

4.2. Solving Mathematical Programs with Equilibrium Constraints

4.2.1. Changing the Formulation of the Lower-Level Problem

Recall that approximating

$$y^T g(x,y) = 0 \tag{4.1}$$

was one of the hurdles in solving MPECs. If *g* is a linear term, approximating the left-hand side of (4.1) often involves specialized techniques, one of which happens to be Schur's decomposition followed with an approximation by linear functions (Gabriel et al., 2006). Moreover, results in this dissertation corroborate this fact by using the same idea for a vector-valued linear function *g* if linear constraints are included. First, using Schur's decomposition, vectors *u* and *v* (dependent on *x* and *y*) are used to rewrite the original MPEC (2.7) as

$$
\begin{aligned}
&\min\ f(x,y)\\
&\text{s.t.}(x,y)\in\Omega\\
&y\geq 0\\
&g(x,y)\geq 0\\
&u^T u - v^T v = 0\\
&u = \frac{y+g(x,y)}{2}\\
&v = \frac{y-g(x,y)}{2}
\end{aligned}
\tag{4.2}
$$

Now, the optimization problem does not contain any bilinear terms. In fact

$$u^T u - v^T v = 0 \tag{4.3}$$

can be readily approximated using SOS type 2 (Beale, 1975)[22] variables to create a piecewise-linear function. However, realizing that the complementarity conditions force $y \geq 0$ and $g(x,y) \geq 0$, shows that only the positive square root of u^2 will give a feasible solution to the problem. Hence, (4.2) can be reformulated as (4.4) below

$$
\begin{aligned}
&\min\ f(x,y)\\
&\text{s.t.}(x,y)\in\Omega\\
&y\geq 0\\
&g(x,y)\geq 0\\
&u - |v| = 0\\
&u = \frac{y+g(x,y)}{2}\\
&v = \frac{y-g(x,y)}{2}
\end{aligned}
\tag{4.4}
$$

The next theorem shows that the solution sets to (2.5), (2.16), and (4.4) are the same.

[22] Special ordered sets of type 2 (SOS type 2) variables are defined as a set of positive variables of which at most two can be non-zero, and if two are non-zero then they need to be next to each other.

Theorem 4.1. Let the solution set to formulation (2.5) be given by S_1, the solution set to formulation (2.16) be given by S_2 and the solution set to formulation (4.4) be given by S_3. Then, given a large enough value[23] of ***K*, $S_1 = S_2 = S_3$.**

Proof. Realize that all three formulations (2.5), (2.16), and (4.4) have the same objective function. Hence, it is sufficient to show that all three formulations have the same feasible region. Hence, assume that S_1, S_2, S_3 represent the feasible regions of formulation (2.5), (2.16), and (4.4), respectively. We will show these feasible regions are equivalent by showing $S_1 \subseteq S_2 \subseteq S_3 \subseteq S_1$. The subscript *i* will denote vector element computation.

Pick a point $(x^1, y^1) \in S_1$. We want to show that there exists a value of *r* such that $(x^1, y^1) \in S_2$. Then, for all *i*, either $y_i^1 = 0$, or $g_i(x^1, y^1) = 0$ or both. Suppose $y_i^1 = 0$. Then, in formulation (2.16), let r_i = 1, which implies $y_i^1 = 0$ in formulation (2.16) as well. If $g_i(x^1, y^1) = 0$, then choose r_i = 0, which implies $g_i(x^1, y^1) = 0$ in formulation (2.16) as well. If both are zero, choose r_i = 1 (or r_i = 0), which will ensure that $y_i^1 = 0$ and that $g_i(x^1, y^1) = 0$ is within the feasible region of (2.16). Since *K* is chosen to be large enough, these arguments imply that the solution set to (2.5) is contained in the solution set to (2.16), i.e., $S_1 \subseteq S_2$.

Next, pick $(x^2, y^2, r^2) \in S_2 \times \{0,1\}^{n_y}$ which is a solution to (2.16). Consider any vector element *i*. Suppose that $r_i^2 = 0$. This implies $g_i(x^2, y^2) = 0$, which implies

---

[23] So that Disjunctive Constraints provides the same solution set as (2.16).

$u_i^2 = \frac{y_i^2}{2}$ and $v_i^2 = \frac{y_i^2}{2}$. Hence, this implies $u_i^2 - v_i^2 = 0$, and in particular $u_i^2 - \left|v_i^2\right| = 0$.

On the other hand, $r_i^2 = 1$ implies $y_i^2 = 0$, $u_i^2 = \frac{g_i(x^2, y^2)}{2}$, and $v_i^2 = -\frac{g_i(x^2, y^2)}{2}$.

Hence, this case also implies that $u_i^2 - \left|v_i^2\right| = 0$. Therefore, $(x^2, y^2) \in S_3$ and $S_2 \subseteq S_3$.

Now pick any solution $(x^3, y^3) \in S_3$. For this solution $u_i^3 - \left|v_i^3\right| = 0$ for each *i*.

Hence, this implies $\left(u_i^3\right)^2 - \left(\left|v_i^3\right|\right)^2 = 0$ and, in particular $\left(u_i^{\,3}\right)^2 - \left(v_i^{\,3}\right)^2 = 0$. Then, the following argument shows that $S_3 \subseteq S_1$.

$$\left(u_i^{\,3}\right)^2 - \left(v_i^{\,3}\right)^2 = 0$$

$$\Leftrightarrow$$

$$\frac{\left(y_i^{\,3}\right)^2 + 2\cdot\left(y_i^{\,3}\right)\cdot g_i(x^3, y^3) + (g_i(x^3, y^3))^2}{4} - \frac{\left(y_i^{\,3}\right)^2 - 2\cdot\left(y_i^{\,3}\right)\cdot g_i(x^3, y^3) + (g_i(x^3, y^3))^2}{4} = 0$$

$$\Leftrightarrow$$

$$y_i^{\,3}\cdot g_i(x^3, y^3) = 0.$$

Hence, S_1, S_2, and S_3 are subsets of each other so they are equivalent.■

4.2.2. Approximating The Absolute Value Function Using Special Ordered Sets of Type 1 Variables

The previous proof shows that using the absolute value function can be a substitute for using disjunctive constraints. However, the absolute value function is also a nonlinear function which can provide computational difficulty to optimization solvers (Steffensen & Ulbrich, 2010). Hence, a reformulation is required.

The absolute value can be reformulated as in (4.5) using Special ordered sets of type 1 (SOS1) variables (Beale & Tomlin, 1970). SOS1 variables are defined as sets of non-negative variables of which at most one can be non-zero.

$$
\begin{aligned}
&\min\ f(x,y) \\
&\text{s.t.}(x,y) \in \Omega \\
&y \geq 0 \\
&g(x,y) \geq 0 \\
&u - (v^+ + v^-) = 0 \\
&u = \frac{y + g(x,y)}{2} \\
&(v^+ - v^-) = \frac{y - g(x,y)}{2} \\
&\text{where } v^+, v^- \text{ are SOS 1 variables}
\end{aligned}
\tag{4.5}
$$

Lemma 4.1. Let S_4 be the solution set to (4.5). The solution sets to formulation (4.5) and formulation (4.4), S_4 and S_3 respectively, are equivalent. That is, S_4 = S_3.

Proof. Again, since the objective functions for both formulations are the same, it is sufficient to show that both formulations have the same feasible region. Hence, assume that S_4 and S_3 represent the feasible regions of formulation (4.5) and (4.4), respectively. Set $v^+ - v^- = v$. Then, for all i, either $(v^+)_i = 0$, or $(v^-)_i = 0$ or both because $(v^+)_i, (v^-)_i$ is a set of SOS1 variables where at most one can be nonzero. This implies that $v^+ + v^- = |v|$ (componentwise absolute value). Hence, we can substitute v in for $v^+ - v^-$ in formulation (9) and $|v|$ for $v^+ + v^-$ in formulation (4.5) to get formulation (4.4). The substitution the other way works as well, hence S_4 = S_3. ■

4.2.3. Approximating Absolute Value Function Using a Penalty Method

The SOS1 approach at times can numerically fail for more complex problems, as SOS1 variables also require binary variables to be formulated within the solver (GAMS, 2010). For example, the North American Gas model could not be solved using the SOS1 formulation and instead required a better starting point as described in Section 4.4. The North American Gas model was eventually solved using Algorithm 4.1 described in Section 4.2.4. Several other alternatives were explored to approximate this absolute value function. In particular, Steffensen and Ulbrich (Steffensen & Ulbrich, 2010) provide a smooth function approximation to the absolute value function. However, their methodology did not work when applied to the example (U.S. version of the World Gas Model (Gabriel et al., 2011c)) in this chapter. An alternative way to approximate the absolute value function is the penalty method (Bazaraa et al., 1993), which works well for finding solutions to MPECs.

$$
\begin{aligned}
&\min\ f(x,y)+\sum_{i=1}^{n_y} L_i(v_i^{+}+v_i^{-})\\
&\text{s.t.}(x,y)\in\Omega\\
&y\geq 0\\
&g(x,y)\geq 0\\
&u-(v^{+}+v^{-})=0\\
&u=\frac{y+g(x,y)}{2}\\
&(v^{+}-v^{-})=\frac{y-g(x,y)}{2}\\
&\text{where } v^{+},v^{-} \text{ are non-negative variables}
\end{aligned}
\tag{4.6}
$$

Theorem 4.2. Assume that the Karush-Kuhn-Tucker conditions are both necessary and sufficient for the optimization problem (4.6). If formulation (4.5)

has a solution, then for any $L_i > 0$ and for each i, at most one of $(v^+)_i$ and $(v^-)_i$ is nonzero in formulation (4.6).

Proof. We will show this by contradiction. Suppose that there exists a $L_i > 0$ such that a solution to (4.6) gives an index i where both $\left(v^+\right)_i > 0$ and $\left(v^-\right)_i > 0$. Let the following be the slightly altered form of (4.6) considered where the Lagrange multipliers are included in parentheses and $\{(x, y)$ s.t. $C(x, y) \leq 0\}$ defines the set of constraints that define Ω.

$$
\begin{aligned}
&\min\ f(x,y) + \sum_{i=1}^{n_y} L_i (v_i^{\ +} + v_i^{\ -}) && \\
&C(x,y) \leq 0 && (\lambda_1) \\
&-y \leq 0 && (\lambda_2) \\
&-g(x,y) \leq 0 && (\lambda_3) \\
&(v^+ + v^-) - \frac{y + g(x,y)}{2} = 0 && (\lambda_4) \\
&(v^+ - v^-) - \frac{y - g(x,y)}{2} = 0 && (\lambda_5) \\
&\text{where } v^+, v^- \text{ are non-negative variables} &&
\end{aligned}
\tag{4.7}
$$

Then, taking the first-order Karush-Kuhn-Tucker conditions (Bazaraa et al., 1993), respectively, for $\left(v^+\right)_i$ and $\left(v^-\right)_i$ gives $L_i + \left(\lambda_4\right)_i + \left(\lambda_5\right)_i = 0$ and $L_i + \left(\lambda_4\right)_i - \left(\lambda_5\right)_i = 0$ since both $\left(v^+\right)_i > 0$ and $\left(v^-\right)_i > 0$. Together these two conditions imply $\left(\lambda_5\right)_i = 0$ and $L_i = -\left(\lambda_4\right)_i$. This implies that the Lagrangian (Λ) can equivalently be expressed as

$$\Lambda = f(x,y) + \sum_{j=1, j\neq i}^{n_y} L_j (v_j^{\ +} + v_j^{\ -}) - (\lambda_4)_i (v_i^{\ +} + v_i^{\ -}) + (\lambda_1)^T C(x,y) + (\lambda_2)^T (-y)$$
$$+ (\lambda_3)^T (-g(x,y)) + (\lambda_4)^T \left((v^+ + v^-) - \frac{y + g(x,y)}{2} \right) + (\lambda_5)^T \left((v^+ - v^-) - \frac{y - g(x,y)}{2} \right).$$

Realizing that λ_4 now appears in two terms, we can factor this out and realize that the following optimization problem will give the same solution as formulation (4.7) above.

$$\begin{array}{ll}
\min\ f(x,y) + \sum_{j=1, j\neq i}^{n_y} L_j (v_j^{\ +} + v_j^{\ -}) & \\
C(x,y) \leq 0 & (\lambda_1) \\
-y \leq 0 & (\lambda_2) \\
-g(x,y) \leq 0 & (\lambda_3) \\
(v_{j\neq i}^{\ +} + v_{j\neq i}^{\ -}) - \frac{y_{j\neq i} + g_{j\neq i}(x,y)}{2} = 0 & (\lambda_4)_{j\neq i} \\
-\frac{y_i + g_i(x,y)}{2} = 0 & (\lambda_4)_i \\
(v^+ - v^-) - \frac{y - g(x,y)}{2} = 0 & (\lambda_5) \\
\text{where } v^+, v^- \text{ are non-negative variables} &
\end{array} \tag{4.8}$$

But since $L_i > 0$, this implies $(\lambda_4)_i < 0$ and since the above formulation satisfies necessary and sufficient[24] conditions for the Karush-Kuhn-Tucker conditions, formulation (4.8) indicates that $\frac{y_i + g_i(x,y)}{2} = 0$. Since both y and g are constrained

[24] Necessary conditions are needed to go from formulation (4.7) to the Karush-Kuhn-Tucker conditions and sufficient conditions to go from Karush-Kuhn-Tucker conditions of (4.7) to optimization formulation (4.8). Also, it can be argued that the constraint associated with $(\lambda_4)_i$ need not be an equality constraint. Hence, we include the fact that $(\lambda_4)_i < 0$ to ensure that we get equality for the associated constraint.

to be nonzero, this implies that $(v^+)_i = (v^-)_i = 0$ for the index i in (4.7). This is a contradiction. Hence, for all $L_i > 0$, the formulation (4.6) gives a solution where for each i, at most one of $(v^+)_i$ and $(v^-)_i$ is nonzero.■

From this point on, $L = \max\{L_i\}$ will be the constant for each variation of (4.6). The value of the constant L should be chosen to be small enough so it does not interfere with the solution. It is not known at this time if there always exists a value of L for which an exact solution is achieved but numerical results suggest there are multiple values of L for which a solution to the MPEC can be obtained. By Theorem 4.2 for all positive L, (4.6) provides a solution that is always a feasible solution to (4.5) but not necessarily optimal for large values of L. Therefore, L can be chosen to be machine epsilon[25]. Numerical results validate that as L approaches zero, the optimal objective function value of (4.6) approaches the optimal objective function value of (2.5). At times, solvers will fail to solve MPECs by finding an infeasible solution where there exists an i, for which both of $(v^+)_i$ and $(v^-)_i$ are nonzero. An alternative to this is provided by the following (heuristic) Algorithm 1.

4.2.4. Algorithm 4.1 to Solve Mathematical Programs with Equilibrium Constraints

<u>Step 0:</u> Pick a tolerance t.

[25] Machine epsilon is defined as the smallest positive number specific to the computer, in this case 10^{-17}.

Step 1: Solve the problem using the penalty method formulation (4.6) with $L = t$.

Step 2: Check for any pairs of variables v^+ and v^- that are both non-zero. If yes, go to Step 3. If not, skip to Step 6.

Step 3: Reformulate those particular variables as SOS1 variables as in formulation (4.5).

Step 4: Solve the problem again using the solution from Step 1 as an initial starting point.

Step 5: Go to Step 2.

Step 6: Check solution by changing value of L in formulation. Decrease L until value for objective function stays the same. Then stop.

4.2.5. Numerical Results

Consider the following sample MPEC where three firms compete to sell natural gas in the market. Assume linear demand and a quadratic cost function. This MPEC is modeled as a Stackelberg game (Gibbons, 1996), where the firms choose quantities to produce. In this context, a Stackelberg game is relevant under the assumption that a shale gas producing firm can exert market power in the North American natural gas market. This assumption can be interpreted in various ways. One way is that other players wait for the shale producing firm to make its production decision before deciding on their own production values. Another interpretation is that the shale producing firm can influence market dynamics so that the other players become reactionary. The Stackelberg leader, “Shale Firm,” has market power and gets to move first while the other two firms are followers.

Table 4.1: Definition of terms for simple example

Parameters	Shale Firm	Firm 1	Firm 2
Intercept and Slope of Linear Demand	a,b	a,b	a,b
Marginal cost	C	c_1	c_2
Positive Constants Used to Replace Complementarities by Disjunctive Constraints	N/A	K_1	K_2

Variables	Shale Firm	Firm 1	Firm 2
Quantity Natural Gas Sold[26]	Q	q_1	q_2
Binary Variables Used to Replace Complementarities by Disjunctive Constraints	N/A	r_1	r_2

Outputs	Shale Firm	Firm 1	Firm 2
Market Price[27]	P	P	P
Profits	*ProfitShale*	*Profit1*	*Profit2*

Shale Firm solves a constrained maximization problem where it maximizes its own profits. This is the upper-level problem:

$$\max_{Q \geq 0} \{(a - b(q_1 + q_2 + Q))Q - CQ\} \tag{4.9}$$

[26] These quantities are constrained to be nonnegative.

[27] We assume a linear demand with $P = a - b(q_1 + q_2 + Q)$.

The firms i = 1,2 at the lower-level solves the following problem where they take quantity Q as given and try to maximize profits while in Nash-Cournot competition with the other Stackelberg follower firm j.

$$\max_{q_i \geq 0} \left\{ \left(a - b\left(q_i + q_j + Q \right) \right) q_i - c_i q_i \right\} \tag{4.10}$$

This lower-level Nash-Cournot game can be expressed as a (linear) complementarity problem given as follows:

$$\begin{aligned} &0 \leq -a + c_1 + 2bq_1 + bq_2 + bQ \perp q_1 \geq 0 \\ &0 \leq -a + c_2 + 2bq_2 + bq_1 + bQ \perp q_2 \geq 0 \end{aligned} \tag{4.11}$$

To solve the problem using disjunctive constraints, the KKT conditions are added to the constraint set in (4.9) to form one overall problem. By having sufficiently large positive constants K_1 and K_2, the complementarity problem (4.11) is reformulated as follows:

$$\begin{aligned} &0 \leq -a + c_1 + 2bq_1 + bq_2 + bQ \leq K_1 r_1 \\ &0 \leq q_1 \leq K_1(1 - r_1) \\ &0 \leq -a + c_2 + 2bq_2 + bq_1 + bQ \leq K_2 r_2 \\ &0 \leq q_2 \leq K_2(1 - r_2) \end{aligned} \tag{4.12}$$

where r_1 are r_2 are binary variables. Let $K_1 = K_2$ be the maximum of the x-intercept, y-intercept of the demand function, and the capacity restrictions, i.e. $K_1 = K_2 = \max\{a/b, a\}$. This provided a lower bound on K_1 and K_2 so that there isn't a computational error (Gabriel & Leuthold, 2010).

Finally, replacing the original complementarity problem with the disjunctive constraints and combining with the upper-level problem, the following mixed-integer nonlinear program formulation is expressed in disjunctive form:

$$
\begin{aligned}
&\max_{Q,q_1,q_2} \{(a-b(q_1+q_2+Q))Q - CQ\} \\
&0 \le -a + c_1 + 2bq_1 + bq_2 + bQ \le K_1 r_1 \\
&0 \le q_1 \le K_1(1-r_1) \\
&0 \le -a + c_2 + 2bq_2 + bq_1 + bQ \le K_2 r_2 \\
&0 \le q_2 \le K_2(1-r_2) \\
&r_1, r_2 \in \{0,1\}
\end{aligned}
\tag{4.13}
$$

The goal is to use (4.13) as a benchmark for comparison to the proposed method. Using (4.9) to (4.11), the MPEC under consideration is reformulated to demonstrate the SOS1 and penalty methods.

$$
\begin{aligned}
&z_1 = -a + c_1 + 2bq_1 + bq_2 + bQ \\
&z_2 = -a + c_2 + 2bq_2 + bq_1 + bQ \\
&z_1 \ge 0 \\
&z_2 \ge 0 \\
&z_1 q_1 = 0 \\
&z_2 q_2 = 0
\end{aligned}
\tag{4.14}
$$

Now, for i = 1, 2, $z_i q_i = u_i - |v_i|$ where $u_i = \dfrac{q_i + z_i}{2}$ and $v_i = \dfrac{q_i - z_i}{2}$ by Schur's decomposition. So the eventual formulation using SOS type 1 variables is:

$$
\begin{aligned}
&\max_{Q,q_1,q_2} \left\{ \left(a - b\left(q_1 + q_2 + Q \right) \right) Q - CQ \right\} \\
&z_1 = -a + c_1 + 2bq_1 + bq_2 + bQ \\
&z_2 = -a + c_2 + 2bq_2 + bq_1 + bQ \\
&z_1 \geq 0 \\
&z_2 \geq 0 \\
&u_1 = \frac{q_1 + z_1}{2} \\
&v_1^+ - v_1^- = \frac{q_1 - z_1}{2} \\
&u_2 = \frac{q_2 + z_2}{2} \\
&v_2^+ - v_2^- = \frac{q_2 - z_2}{2} \\
&u_1 - (v_1^+ + v_1^-) = 0 \\
&u_2 - (v_2^+ + v_2^-) = 0 \\
&\text{where } v_i^+, v_i^- \text{ are SOS type1 variables}
\end{aligned}
\tag{4.15}
$$

Similarly, the formulation for the penalty method is given by

$$
\begin{aligned}
&\max_{Q,q_1,q_2} \left\{ \left(a - b\left(q_1 + q_2 + Q\right)\right)Q - CQ - L\left(\sum_{i=1}^{2} v_i^+ + v_i^-\right) \right\} \\
&z_1 = -a + c_1 + 2bq_1 + bq_2 + bQ \\
&z_2 = -a + c_2 + 2bq_2 + bq_1 + bQ \\
&z_1 \geq 0 \\
&z_2 \geq 0 \\
&u_1 = \frac{q_1 + z_1}{2} \\
&v_1^+ - v_1^- = \frac{q_1 - z_1}{2} \\
&u_2 = \frac{q_2 + z_2}{2} \\
&v_2^+ - v_2^- = \frac{q_2 - z_2}{2} \\
&u_1 - (v_1^+ + v_1^-) = 0 \\
&u_2 - (v_2^+ + v_2^-) = 0 \\
&\text{where } v_i^+, v_i^- \text{ are non-negative variables}
\end{aligned}
\tag{4.16}
$$

The computational objective was to compare the results for the methods of disjunctive constraints (4.13), SOS1 variables (4.15) and the penalty method (4.16). This was accomplished using GAMS (GAMS, 2010) with CONOPT being used as the nonlinear solver and SBB the mixed integer nonlinear solver.

The capacity constraints for production quantities are omitted for this first example, i.e., maximum production is only limited by rising costs. The following Table 4.2 shows the datasets used:

Table 4.2: Different Datasets to Compare (4.13), (4.15), and (4.16)

Dataset Parameters	Dataset 1	Dataset 2	Dataset 3
a	13	13	13
b	1	0.1	0.1
$c_1 = c_2 = C$	1	1	2

For the disjunctive constraints formulation (4.16), $K_1 = K_2 = \max\{b/a, a\} = 13$ consistent with (Gabriel & Leuthold, 2010) for all the datasets while for the penalty method approximation (4.16), two different values of L were chosen to show how a lower value of L gives a better answer as shown below in Table 4.3.

Table 4.3: Different Cases to Compare Solutions to (4.16)

	Case 1	Case 2
Value of L	0.0001	1

The following Tables 4.4-4.6 report the results. The true answer[28] can be easily verified algebraically as unique and is shown in the third column of the tables. Note that disjunctive constraints obtained the correct answer for Dataset 1, implying that a correct value of K was chosen.

[28] It is simple algebra to show that this is the unique solution since there are no constraints and all objective functions are quadratic.

Table 4.4: Results for Dataset 1

Results	Disj Cons	True Answer	SOS	Case1	Case 2
$q_1 = q_2$	2.000	2.000	2.000	2.000	1.833
Q	6.000	6.000	6.000	6.000	6.500
Price	3.000	3.000	3.000	3.000	2.833
Profit shale	12.000	12.000	12.000	12.000	11.917
Profit 1=2	4.000	4.000	4.000	4.000	3.361

Table 4.5: Results for Dataset 2

Results	Disj Cons	True Answer	SOS	Case1	Case 2
$q_1 = q_2$	13.000	20.000	20.000	20.000	18.333
Q	81.000	60.000	60.000	60.000	65.000
Price	2.300	3.000	3.000	3.000	2.833
Profit shale	105.300	120.000	120.000	120.000	119.167
Profit 1=2	16.900	40.000	40.000	40.000	33.611

Table 4.6: Results for Dataset 3

Results	Disj Cons	True Answer	SOS	Case1	Case 2
$q_1 = q_2$	13.000	18.333	18.333	18.333	16.667
Q	71.000	55.000	55.000	55.000	60.000
Price	3.300	3.833	3.833	3.833	3.667
Profit shale	92.300	100.833	100.833	100.833	100.00
Profit 1=2	16.900	33.611	33.611	33.611	27.778

If the methodology to choose K as outlined in the literature (Gabriel & Leuthold, 2010) is used, disjunctive constraints do not provide the solutions in datasets 2 and 3[29]. These results point out a big weakness with disjunctive constraints that the solution can be very far from the true answer and the given solution can be extremely sensitive to the value of K if appropriate problem specific values are not selected.

Choosing the correct K can make the disjunctive constraint method (4.13) accurate. Choosing a correct L makes (4.16) accurate as well. The next set of numerical results were done with Dataset 3 with K =10000 and[30] $L = 10^{-16}$ where these values were reached after numerical and algebraic verification of the test problem. The test problem was changed so that now instead of two players at the

[29] The method in (Gabriel & Leuthold, 2010) gives a correct value of K whenever maximum production (capacity constraints for production) is included in the problem formulation. Our goal was to give a very simple counterexample where the disjunctive constraints approach didn't work.

[30] This is machine-ε.

lower level, there were *M* players with similar costs and parameters. The number of players was increased to test the computation time taken for disjunctive constraints (4.13), SOS1 (4.15), and the penalty method (4.16). The results are shown in the following Figure 4.1. All methods were able to obtain the correct solutions.

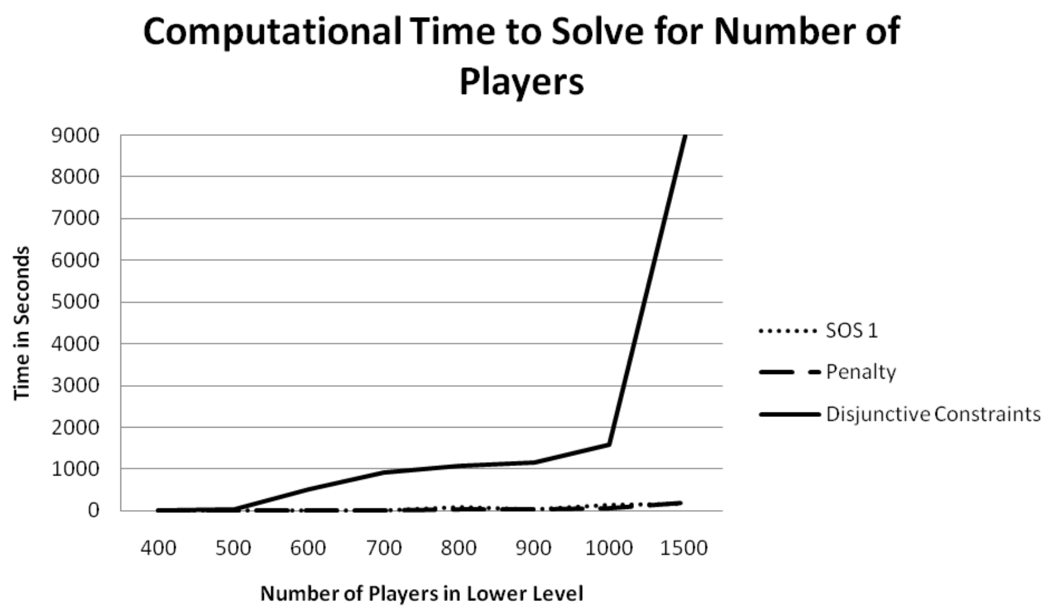

Figure 4.1: Computational Time for Solving Problem

Clearly, the disjunctive constraint method becomes extremely computationally expensive when number of players is increased. Note that the graphs for the penalty and SOS1 methods are overlapping.

4.3. Solving Equilibrium Programs with Equilibrium Constraints

4.3.1. Extending Algorithm 4.1 to Equilibrium Programs with Equilibrium Constraints

Note that a variation of the above formulation (4.6) can also be used to solve EPECs. An EPEC is defined as a game between N players at the top level where each top-level player solves an optimization problem of the form of an MPEC. Hence, an EPEC with a common lower-level for each of the N upper-level players typical of Stackelberg leaders in energy production with the rest of the market represented by the lower-level problem is given by

$$\begin{aligned}
&\min\ f_j(x,y) \qquad\qquad j=1,\ldots,N \\
&\text{s.t.}(x,y)\in\Omega \\
&y\geq 0 \\
&g(x,y)\geq 0 \\
&y^T g(x,y)=0
\end{aligned} \tag{4.17}$$

The formulation (4.17) with $\Omega=\{(x,y)\,|\,C(x,y)\leq 0\}$ can be rewritten as

$$\begin{aligned}
&\min\ f_j(x,y)+L\sum_{i=1}^{n_y}(v_i^{+}+v_i^{-}) && j=1,\ldots,N \\
&C(x,y)\leq 0 && (\lambda_1) \\
&-y\leq 0 && (\lambda_2) \\
&-g(x,y)\leq 0 && (\lambda_3) \\
&(v^{+}+v^{-})-\frac{y+g(x,y)}{2}=0 && (\lambda_4) \\
&(v^{+}-v^{-})-\frac{y-g(x,y)}{2}=0 && (\lambda_5) \\
&\text{where } v^{+}, v^{-} \text{ are non-negative variables}
\end{aligned} \tag{4.18}$$

By Theorem 4.2, choosing a positive L will ensure that the SOS1 constraints hold for each pair v^{+}, v^{-} for each individual top-level player's optimization problem and choosing a small enough L will ensure that the correct solution is achieved. Formulation (4.18) can then be solved as a Nash game among N players, and can be formulated as a complementarity problem by taking the Karush-Kuhn-Tucker conditions as in (4.19).

$$
\begin{array}{c}
0 \le \nabla_x f_j(x,y) + \lambda_1 \nabla_x C(x,y) - \lambda_3 \nabla_x g(x,y) - \frac{\lambda_4}{2} \nabla_x g(x,y) + \frac{\lambda_5}{2} \nabla_x g(x,y) \perp x \ge 0 \\
0 \le \nabla_y f_j(x,y) + \lambda_1 \nabla_y C(x,y) - \lambda_3 \nabla_y g(x,y) - \frac{\lambda_4}{2} (1 + \nabla_y g(x,y)) - \frac{\lambda_5}{2} (1 - \nabla_y g(x,y)) \perp y \ge 0 \\
0 \le L v_i^+ + \lambda_4 v_i^+ + \lambda_5 v_i^+ \perp v_i^+ \ge 0 \\
0 \le L v_i^- + \lambda_4 v_i^- - \lambda_5 v_i^- \perp v_i^- \ge 0 \\
i = 1,\ldots,n_y,\ j = 1,\ldots,N \\
0 \le -C(x,y) \perp \lambda_1 \ge 0 \\
0 \le g(x,y) \perp \lambda_3 \ge 0 \\
(v^+ + v^-) - \frac{y + g(x,y)}{2} = 0 \qquad (\lambda_4 \quad free) \\
(v^+ - v^-) - \frac{y - g(x,y)}{2} = 0 \qquad (\lambda_5 \quad free)
\end{array}
\tag{4.19}
$$

Note that as described in detail in (Ehrenmann, 2004), (4.19) is not a square system. The reason for this is that the same set of lower-level variables are shared among all top-level players in the component MPECs of the EPEC. Hence, solutions to (4.19) cannot be computed using solvers in GAMS. Many workarounds are available for this (e.g. penalization methods (Ehrenmann, 2004)), and for the case of the specific EPEC considered in this chapter, we introduce a “balancing agent.” All lower-level variables and constraints that are common to all top-level player optimization problems are treated as separate variables for each top-level player. For example, the variables y above will be treated as separate variables y_j for each of the top-level player MPECs. Then, (4.19) combined with the Karush-Kuhn-Tucker conditions[31] of (4.20) below will be a square system. The balancing agent solves the following problem

$$\min_{b_j^+, b_j^-} \sum_{j=1}^{N-1} b_j^+ + b_j^-$$
$$y_j - y_{j+1} - b_j^+ + b_j^- = 0 \qquad (\pi_j) \tag{4.20}$$
$$b_j^+, b_j^- \geq 0$$
$$j = 1, \ldots, N-1$$

A loose interpretation of the economic role of the balancing agent is the following. Without such an agent, Stackelberg leader j communicates with only the jth partition of the lower-level market, represented by y_j. The balancing agent tries at minimal cost to couple the partitions into one integrated market, which is more realistic. The Karush-Kuhn-Tucker conditions of the above problem can then be added to (4.19) to ensure that the values of values of y_j are the same. The following algorithm shows a method, then, to solve EPECs.

4.3.2. Algorithm 4.2 to Solve Equilibrium Problems with Equilibrium Constraints (Heuristic)

(Solving EPECs Using (4.19) and Karush-Kuhn-Tucker conditions of (4.20))

<u>Step 0:</u> Pick a tolerance t.

---

[31] Since (4.20) is a linear program, the Karush-Kuhn-Tucker conditions are both necessary and sufficient.

Step 1: Solve the problem using the formulation (4.19) with $L = t$ and the KKT conditions to (4.20).

Step 2: Check solution by changing value of L in formulation. Decrease L until value for objective function stays the same. Check with setting L = machine-ε. Then stop.

4.3.3. Numerical Results for Equilibrium Programs with Equilibrium Constraints

A corresponding EPEC where two players are at the top-level can also be formulated and solved by extending the MPEC method above. The formulation for the bottom level remains the same, and for the upper level, there are now two producers who determine quantities Q_1 and Q_2 whose objective functions are given as

$$\left\{\left(a - b\left(q_1 + q_2 + Q_1 + Q_2\right)\right)Q_j - C_j Q_j\right\} \quad j = 1,2 \tag{4.21}$$

Using the same datasets, let[32] $C_1 = C_2 = C$. Then, the EPEC can be formulated as

[32] This also works with other data, which was verified numerically as well.

$$
\begin{aligned}
&\max_{Q_j, q_1, q_2} \left\{ \left(a - b\left(q_1 + q_2 + Q_1 + Q_2\right)\right) Q_j - C_j Q_j - L\left(\sum_{i=1}^{2} v_i^+ + v_i^-\right) \right\} \quad j = 1,2 \\
&-a + c_1 + 2bq_1 + bq_2 + bQ_1 + bQ_2 \geq 0 \qquad (\lambda_1) \\
&-a + c_2 + 2bq_2 + bq_1 + bQ_1 + bQ_2 \geq 0 \qquad (\lambda_2) \\
&v_1^+ + v_1^- - \frac{q_1 + (-a + c_1 + 2bq_1 + bq_2 + bQ_1 + bQ_2)}{2} = 0 \qquad (\lambda_3) \\
&v_1^+ - v_1^- - \frac{q_1 - (-a + c_1 + 2bq_1 + bq_2 + bQ_1 + bQ_2)}{2} = 0 \qquad (\lambda_4) \\
&v_2^+ + v_2^- - \frac{q_2 + (-a + c_2 + 2bq_2 + bq_1 + bQ_1 + bQ_2)}{2} = 0 \qquad (\lambda_5) \\
&v_2^+ - v_2^- - \frac{q_2 - (-a + c_2 + 2bq_2 + bq_1 + bQ_1 + bQ_2)}{2} = 0 \qquad (\lambda_6) \\
&\text{where } v_i^+, v_i^- \text{ are non-negative variables}
\end{aligned}
\tag{4.22}
$$

The constraints in (4.22) are the KKT conditions of the lower-level problem that have been reformulated as in (4.6). As described in Section 4.3.1, this problem can be expressed and solved as a complementarity problem using Algorithm 4.2 and adding a balancing agent:

$$
\begin{aligned}
&0 \le -a + C_j + bq_1 + bq_2 + bQ_1 + bQ_2 + bQ_j - \frac{\lambda_{3,j} b}{2} + \frac{\lambda_{4,j} b}{2} - \frac{\lambda_{5,j} b}{2} + \frac{\lambda_{6,j} b}{2} \perp Q_j \ge 0 \\
&0 \le bQ_j - 2b\lambda_{1,j} - b\lambda_{2,j} + \lambda_{3,j}\left(-\frac{1+2b}{2}\right) + \lambda_{4,j}\left(-\frac{1-2b}{2}\right) + \lambda_{5,j}\left(-\frac{b}{2}\right) + \lambda_{6,j}\left(-\frac{-b}{2}\right) \perp q_{1,j} \ge 0 \\
&0 \le bQ_j - b\lambda_{1,j} - 2b\lambda_{2,j} + \lambda_{5,j}\left(-\frac{1+2b}{2}\right) + \lambda_{6,j}\left(-\frac{1-2b}{2}\right) + \lambda_{3,j}\left(-\frac{b}{2}\right) + \lambda_{4,j}\left(-\frac{-b}{2}\right) \perp q_{2,j} \ge 0 \\
&0 \le L + \lambda_{2i+1,j} + \lambda_{2i+2,j} \perp v^+_{i,j} \ge 0 \\
&0 \le L + \lambda_{2i+1,j} - \lambda_{2i+2,j} \perp v^-_{i,j} \ge 0 \\
&0 \le -a + c_1 + 2bq_{1,j} + bq_{2,j} + bQ_1 + bQ_2 \perp \lambda_{1,j} \ge 0 \\
&0 \le -a + c_2 + 2bq_{2,j} + bq_{1,j} + bQ_1 + bQ_2 \perp \lambda_{2,j} \ge 0 \qquad (4.23) \\
&v^+_{1,j} + v^-_{1,j} - \frac{q_{1,j} + (-a + c_1 + 2bq_{1,j} + bq_{2,j} + bQ_1 + bQ_2)}{2} = 0 \qquad (\lambda_{3,j} \quad free) \\
&v^+_{1,j} - v^-_{1,j} - \frac{q_{1,j} - (-a + c_1 + 2bq_{1,j} + bq_{2,j} + bQ_1 + bQ_2)}{2} = 0 \qquad (\lambda_{4,j} \quad free) \\
&v^+_{2,j} + v^-_{2,j} - \frac{q_{2,j} + (-a + c_2 + 2bq_{2,j} + bq_{1,j} + bQ_1 + bQ_2)}{2} = 0 \qquad (\lambda_{5,j} \quad free) \\
&v^+_{2,j} - v^-_{2,j} - \frac{q_{2,j} - (-a + c_2 + 2bq_{2,j} + bq_{1,j} + bQ_1 + bQ_2)}{2} = 0 \qquad (\lambda_{6,j} \quad free) \\
&0 \le 1 - \pi_j \perp b^+_j \ge 0 \\
&0 \le 1 + \pi_j \perp b^-_j \ge 0 \\
&q_{i,j} - q_{i,3-j} - b^+_j + b^-_j = 0 \qquad (\eta_j \quad free) \\
&b^+_j, b^-_j \ge 0 \\
&j = 1,2, \quad i = 1,2
\end{aligned}
$$

The following tables (4.7-4.9) give the solutions under different datasets and cases[33]. Simple algebra can show that there exists a solution, and hence a true answer is also given in the table.

[33] Many different cases with different costs were also solved successfully, but only the ones corresponding to the previous MPEC example are presented. Please refer to Section 4.2.5.

Table 4.7: Results for Dataset 1

Results	**A Solution**	**Case 1**	**Case 2**	***L* = machine-ε**
$q_1 = q_2$	1.333	1.333	1.111	1.333
$Q_1 = Q_2$	4.000	4.000	4.333	4.000
Price	2.333	2.333	2.111	2.333
Profit Top	5.333	5.333	4.815	5.333
Profit Bottom	1.778	1.778	1.235	1.778

Table 4.8: Results for Dataset 2

Results	A Solution	Case 1	Case 2	L = machine-ε
$q_1 = q_2$	13.333	13.333	11.111	13.333
$Q_1 = Q_2$	40.000	40.000	43.333	40.000
Price	2.333	2.333	2.111	2.333
Profit Top[34]	53.333	53.333	48.148	53.333
Profit Bottom	17.778	17.778	12.346	17.778

Table 4.9: Results for Dataset 3

Results	A Solution	Case 1	Case 2	L = machine-ε
$q_1 = q_2$	12.222	12.222	10.000	12.222
$Q_1 = Q_2$	36.667	36.667	40.000	36.667
Price	3.222	3.222	3.000	3.222
Profit Top	44.815	44.815	40.000	44.815
Profit Bottom	14.938	14.938	10.000	14.938

Again, in all three datasets, the Case 2 choice of L could not give an optimal solution, i.e., Nash equilibria at the top and bottom. Hence, all these datasets required a very small choice of L. Choosing L = machine-ε is a good option. Note that the EPEC has an extra player when compared to the MPEC; hence profits for all firms are lower in the EPEC study. Moreover, prices are also lower in the EPEC case, as

[34] Due to the deliberate selection of similar data, the profits for both top-level players are the same. Hence, only one player's profit is reported as the second player's profits are the same.

expected with an extra Stackelberg leader with same marginal cost in the EPEC. These results can be seen when comparing Tables (4.4-4.6) with Tables (4.7-4.9).

4.4. The North American Gas Model

4.4.1. Introduction

The advent of rising oil prices along with attitudes about decreasing greenhouse gas emissions in multiple sectors has lead to an interest in natural gas production for the future. The role of unconventional gas[35], in particular, has greatly increased due to engineering advances such as hydraulic fracturing and horizontal drilling (NPC, 2007). The projected role of shale gas in particular, especially in the United States but also elsewhere (Skagen, 2010) has lately been a major force in the increasing prominence of unconventional gas. In 2008, Cambridge Energy Research Associates indicated that this unconventional gas production could help delay by a decade the United States' need for substantial LNG imports (Economist, 2008). More recently, others gauge the U.S. shale gas impact in even more dramatic terms with estimates of up to 100 years of reserves. [36] Indeed, the Potential Gas Committee has concluded that the U.S. proved reserves of gas increased from 2006 to 2008 by a huge 35.4% from 1532.0 trillion cubic feet to 2074.1 (PGC, 2010). Others such as the petroleum

[35] Unconventional gas is defined as gas from tight sands, coalbed methane, and shale gas, and covers more low-permeability reservoirs that produce mostly natural gas (no associated hydrocarbon liquids) (NPC, 2007).

[36] Keith O. Rattie, CEO of Questar, a natural gas and pipeline company, cited in "Awash in Fossil Fuels," George F. Will, The Washington Post, November 22, 2009.

engineer Art Berman are more cautious about the ultimate supply due to the economics of producing shale gas (low gas prices in the U.S. recently) (Cohen, 2009) or steeper decline rates for shale wells (Steffy, 2009). Shale gas in the U.S. will be modeled using the World Gas Model (Gabriel et al., 2011c), restricted to North American nodes.

The World Gas Model[37] (WGM) is a long-term, game theoretic model of global gas markets with representation of Cournot market power originally based on a North American version of the mode (Gabriel et al., 2005a), (Gabriel et al., 2005b) and eventually extended to a global version (Egging et al., 2009) for which the most recent version is (Gabriel et al., 2011c). For the United States, the forecasts presented in the Annual Energy Outlook (April 2009 ARRA version) were used for the current study. For the rest of North America, the World Energy Outlook (IEA, 2008) was used. The WGM was then extensively calibrated to match these multiple sources for all countries/aggregated countries and years considered (2005, 2010, 2015, 2020, 2025, 2030).[38]

The most interesting change due to the presence of shale gas occurs in census region 7 (WGM node 7) where Haynesville and Barnett plays are present. This node

---

[37] National Science Foundation (DMS, Award #0106880), Principal Investigator ,S.A. Gabriel, “Computational Methods for Equilibrium Problems with Micro-Level Data,” 09/01/2001-08/31/2005 and National Science Foundation (DMS, Award #0408943), Principal Investigator, S.A. Gabriel, “Methods and Models for Stochastic Energy Market Equilibria, “08/01/2004-07/31/2008.

[38] See (Gabriel et al., 2011c) for details on the countries and regions included as well as other relevant geographic or nodal information.

is used as a test example for the new MPEC solution technique. The formulation proposed is that the producer of shale at node 7 will be the first mover in the Stackelberg game (Gibbons, 1996). The entire lower level will be the World Gas Model restricted to North American nodes including other shale nodes. This formulation assumes that the shale producer at node 7 has market power over all other players. While there is some arbitrariness about this assumption, i.e. it might give the shale producer at node 7 too much market power, it nevertheless is an interesting market formulation to study because there is a chance for this scenario to play out in the future. Furthermore, this formulation can be used for bounding purposes when considering a wide variety of market dynamics. Other MPEC formulations might consider a trader, producer, or even the government at the upper-level.

4.4.2. Shale Gas in the United States

The shale gas data were provided by the U.S. Department of Energy in the Annual Energy Outlook (2010) with shale gas production and Lower 48 onshore natural gas production datasets. [39] As compared to the version of the model from (Gabriel et al., 2011c), the World Gas Model was modified to contain three production nodes for each census region of the United States: conventional gas, shale gas, and non-shale unconventional gas.

A ‘Golombek’ production cost function (Golombek & Gjelsvik, 1995)

$$C(q) = (\alpha - \gamma)q + \frac{1}{2}\beta q^2 + \gamma(Q - q)\ln\left(\frac{Q - q}{Q}\right) \tag{4.28}$$

[39] For a table relating the WGM nodes to the shale plays in the US, please refer to the Appendix.

was used for which the marginal supply cost curve is:

$$C'(q) = \alpha + \beta q + \gamma \ln\left(\frac{Q-q}{Q}\right). \tag{4.29}$$

Here, Q is the production capacity, $\alpha > 0$ is the minimum per unit cost, $\beta > 0$ is the per unit linearly increasing cost term, and $\gamma \le 0$ is a term that induces high marginal costs when production is close to full capacity.

Skagen (Skagen, 2010) indicates that recent research has led to predicting a lower value of α for the cost function of shale gas when compared to conventional gas. Figure 4.2 shows that shale gas is now understood to have a lower price of extraction in the beginning.

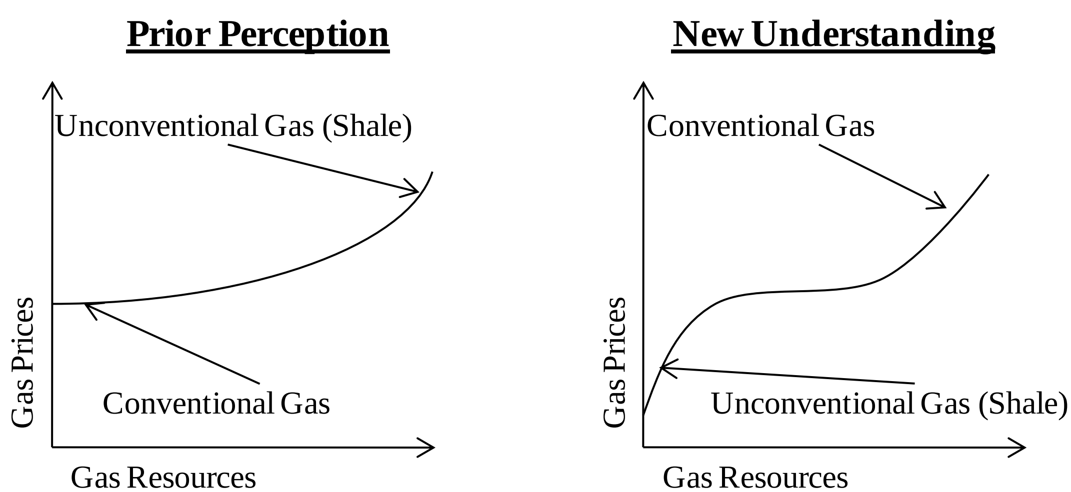

Figure 4.2: A Marginal Cost Structure for Shale Gas (Skagen, 2010)

Alternatively, others believe that initial positive results from shale gas extraction wells might not be sustainable in the long run (Cohen, 2009). In particular, geologist Art Berman claims that decline rates will be much higher than expected, and while shale appears to be a good resource right now, steep decline rates mean that higher extraction will lead to higher costs quickly (Cohen, 2009).

In the modification of the WGM, the shale gas cost curve has α (the *y*-intercept of the marginal cost curve) lower and β (the slope of the marginal cost curve) higher than for conventional gas. The current debate about shale gas has been incorporated. While the lower initial cost of extraction is consistent with Skagen's observation, a higher marginal cost increase and higher marginal costs at higher quantities is consistent with Berman's claim that decline rates of shale wells will be higher. Hence, shale gas has a lower initial cost of extraction than conventional gas but a higher rate of increase for marginal cost. It is important to note that this marginal cost curve for shale gas is by no means the final word but just one perspective developed for our modeling needs.

The other initial condition placed is that total production costs should be the same, so the integral of the marginal cost curve should be the same for both functions (conventional and shale gas). This will ensure a positive production of both types of gas, which can be calibrated to real data. A comparison is provided with two other cases with higher total costs for shale production. Another reason why the total costs would be equal in the reference case is that producers drilling in the same region would encounter similar terrain, similar taxes, similar hurdles etc. Hence, α was reduced by 20% of the value of conventional gas based on Skagen (2010) and β was increased by an amount so that the integral of the marginal cost curve remains the same. An explanation of this is shown in Figure 4.3 below. Note that the values of γ are kept the same for shale and conventional gas, so Figure 4.3 only shows the linear portion of the marginal cost curve. The production cost data for conventional and unconventional (non-shale) gas was obtained as described in (Gabriel et al., 2011c).

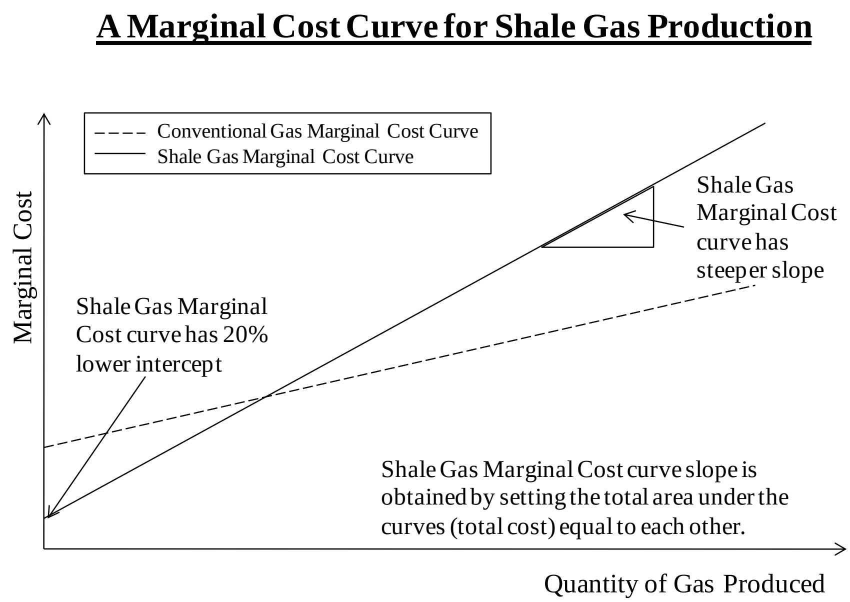

Figure 4.3: A Marginal Cost Structure for Shale Gas

The following table provides the coverage of states and shale basins in the world gas model.

Table 4.10: World Gas Model Nodes: Coverage of States and Shale Basins

Shale Basin Name	States							WGM Nodes
Mancos	Utah							US_8
Hilliard-Baxter Mancos	Wyoming	Colorado						US_8
Niobrara	Colorado	Nebraska	Kansas					US_4, US_8
Cody	Montana							US_8
Mowry	Wyoming							US_8
Gammon	Montana	North Dakota	South Dakota					US_4, US_8
Excello-Mulky	Kansas	Oklahoma						US_4, US_7
New Albany	Illinois	Indiana	Kentucky					US_3, US_6
Antrim	Michigan	Indiana	Ohio					US_3
Utica	New York							US_2
Marcellus	New York	Pennsylvania	Ohio	West Virginia	Maryland	Virginia	Tennessee	US_2, US_3, US_5, US_6
Devonian	Ohio	Kentucky	West Virginia	Virginia	Tennessee	Alabama	Georgia	US_3, US_5, US_6
Chattanooga	Kentucky	Virginia	Tennessee	Alabama	Georgia			US_5, US_6
Conasauga	Alabama	Georgia						US_5, US_6
Floyd-Neal	Mississippi	Alabama						US_6
Fayetteville	Arkansas							US_7
Hayneville/ Bossier	Louisiana	Texas						US_7
Woodford/ Caney	Oklahoma							US_7
Barnett	Texas							US_7
Pearsall	Texas							US_7
Woodford	Oklahoma	Texas						US_7
Barnett and Woodford	New Mexico	Texas						US_7, US_8
Bend	Texas							US_7
Pierre	New Mexico	Colorado						US_8
Lewis	New Mexico	Colorado						US_8
Hermosa	Utah							US_8

4.4.3. Scenario Results

This section describes numerical examples to solve the WGM (North American nodes only) using the new MPEC approach outlined above. The point is to demonstrate that even on large MPECs, this Algorithm 4.1 works well. Five different cases were run, which are described below. The results from these cases are consistent with economic theory, and are presented in graphical form.

Computational results show that a solution exists for the lower-level complementarity problem (which includes the shale producer at node 7). This means that a feasible solution for the MPEC exists as well, as the lower-level complementary problem contains both the complementary restrictions as well as constraints for the upper-level player of the MPEC. The method of disjunctive constraints did not provide a feasible solution for this problem with the solvers SBB and CONOPT (GAMS, 2010).

The WGM restricted to the North American nodes has 30 producers, of which seven are for shale gas and seven for unconventional gas production in the United States. The rest produce conventional gas. There are a total of 15 production nodes, of which nine correspond to the census regions for the lower-48 states. There are also three traders (one each for United States, Canada, and Mexico, the three countries in the model), along with eight periods from 2005-2040 (the last two five-year periods are not reported to avoid end-of-horizon bias), and two seasons (high and low demand) in each period. The decision variables are operating levels (production, storage injection, etc.) as well as investment levels (pipeline, liquefaction capacity, etc.). Prices are set to 2005 US$. The whole complementarity model has about 9456 variables and takes 243.2 seconds to solve on a 2.0 GHz processor with 2 GB memory.

The MPEC version of the WGM restricted to North America was formulated with the shale gas producer in census region 7 as the top-level player. Census region 7 contains both the Barnett and Haynesville shale plays, two of the most important ones in the United States[40]. The MPEC version was solved using Algorithm 4.1. The algorithm solved the problem in approximately three hours on the same computer described above, though the time was different for each case.

The following five cases were considered, with the first (Base Case) modeled as a complementarity problem and the rest as MPECs for purposes of comparison:

1) **Base**: The Base Case for the WGM restricted to North America formulated as a complementarity problem and calibrated according to the Annual Energy Outlook (April 2009 ARRA version) and the World Energy Outlook (IEA, 2008).
2) **MPEC**: The MPEC version of the Base Case. The shale producer in census region 7 was placed at the upper level and all other players at the lower level.
3) **MoreShale**: A higher production of shale gas was considered by increasing the daily capacity available, with a 10% increase for 2015, 2020; a 15% increase for 2025, 2030; and a 20% increase for 2035, 2040. These numbers are approximations of increases given by the Annual Energy Outlook between the 2008 and 2009 reports' predictions. While the 2010 reports did not show such an increase, for our purposes this case was developed to show what might happen if a similar increase took place after 2015. This case is modeled as an MPEC.

[40] Refer to www.eia.gov for more information.

4) **ShaleTax**: All shale-producing firms are taxed $0.39/MCF (39 cents for every thousand cubic feet of natural gas produced) from 2015 to 2040. This is in line with the tax proposed for Pennsylvania shale production in the Marcellus shale play, which was later overturned (Barnes, 2010). No other value for a shale tax has so far been found from any legislature. This case is modeled as an MPEC.
5) **AllTax**: All natural gas is taxed at $0.39/MCF from 2015 to 2040. This case will help see if the shale players, especially the one in census region 7, have any comparative advantage when everyone is taxed. Modeled as an MPEC.

The results are presented below. The MPEC case produces lower average prices (e.g., $6.74/MMBTU vs. $6.94/MMBTU in 2025) and higher total production (e.g., 844.2 BCM vs. 830.2 BCM in 2025) and consumption (Gibbons, 1996) when compared to the Base Case for all years. Moreover, as expected, the MoreShale case showed an overall increase in shale production when compared to the Base Case (e.g., 111.5 BCM vs. 89.4 BCM in 2025) and for the shale producer in census region 7, proved to be the most profitable. The profits at node 7 increase by more than three times in 2025 when compared to the Base Case. This shows the advantage of being the Stackelberg leader and allowing collection of more profits and also serves as a cautionary numerical result for market regulators and other interested parties.

The MoreShale case shows that it will be advantageous for producers as well as consumers (with prices dropping in nodes with large amounts of shale). However, the fact that total production doesn't change much with the invoking of the tax (shale or otherwise) shows that it will not be detrimental to the producers. This is corroborated by looking at producer profits as well, where the imposition of tax barely changes total profit. Since Node 8 has a relatively abundant supply of conventional, unconventional, and shale gas, it can change production around depending on the demands. Hence, nodes 8 and 9 remain relatively unchanged with the imposition of tax. Moreover, the production for shale producers is as expected, and the imposition of tax does less to harm any production, and overall profits remain relatively stable. Also, this might be a policy argument for saying that the tax will barely harm producers, but produce revenue for the state.

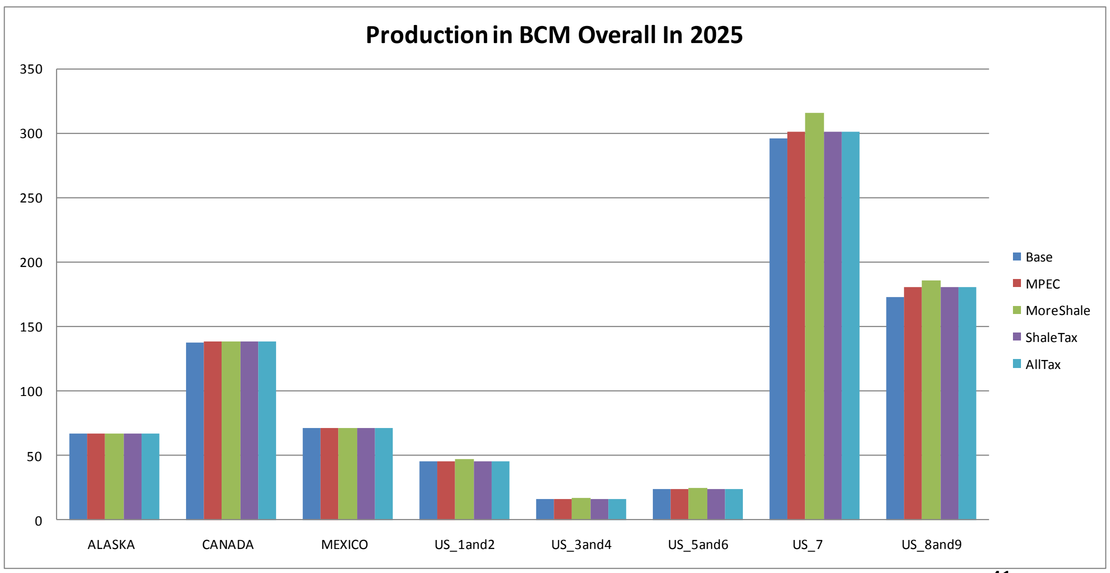

Figure 4.4: Overall Production in 2025 as Predicted by the Model[41]

[41] US_1and2, for example, gives data for US census regions 1 and 2 combined.

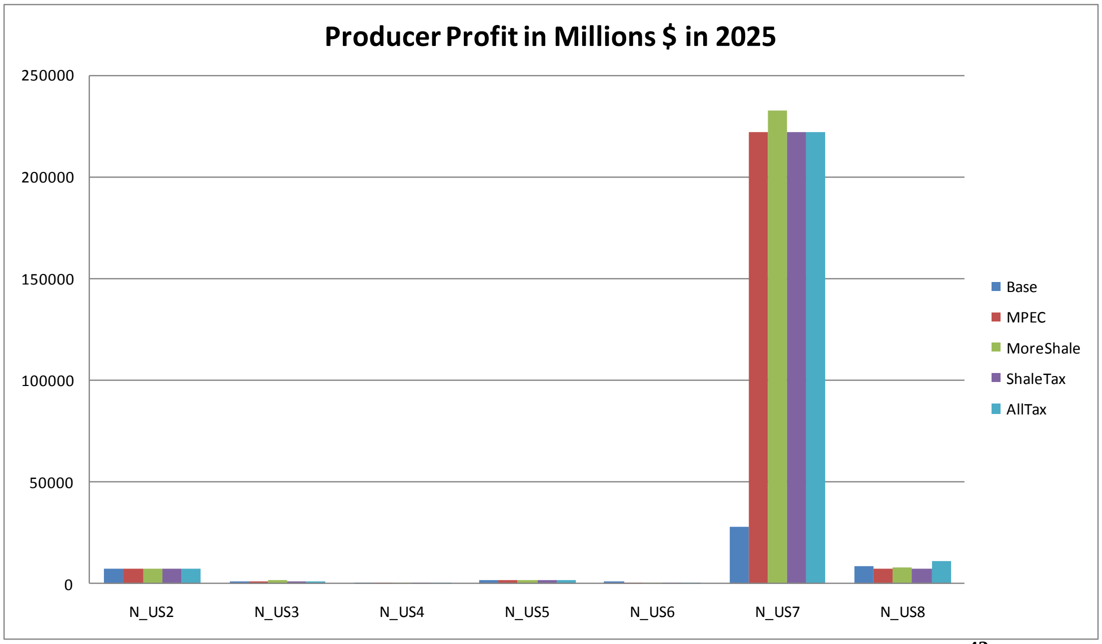

Figure 4.5: Producer Profit in 2025 as Predicted by the Model [42]

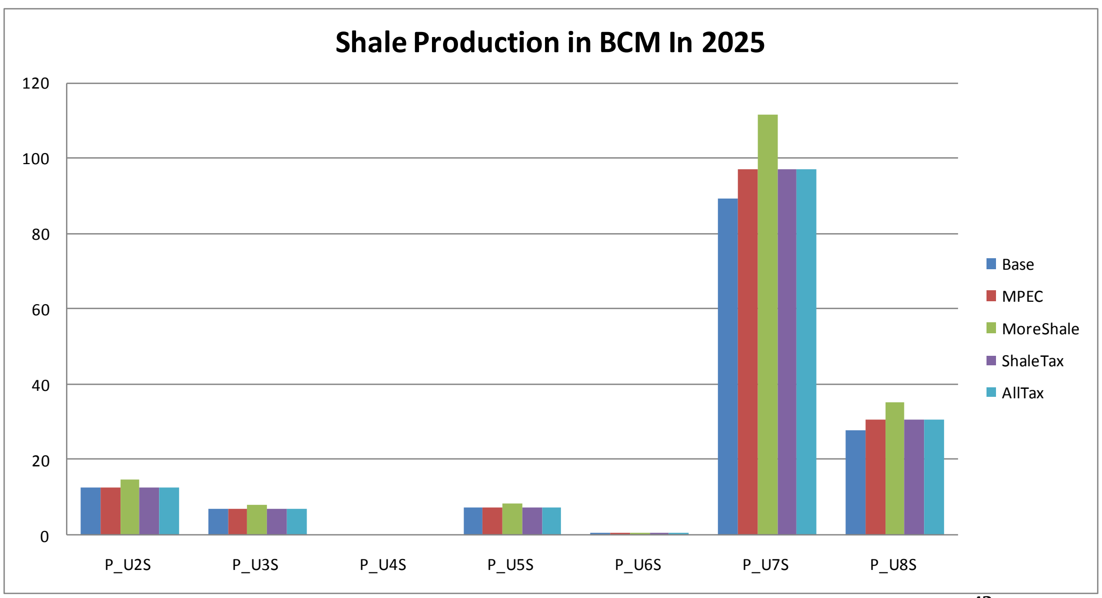

Figure 4.6: Shale Producers in 2025 as Predicted by the Model [43]

[42] N_US3, for example, gives profit at the node for US census region 3.

[43] P_U5S, for example, gives the production at US node 5 for shale gas.

Data for consumption and prices (Figure 4.7 and Table 4.11, respectively), however, show that the producers will pass most of the tax onto the consumers. This also shows the strength of the World Gas Model, by predicting which areas will show a change in prices. Nodes 5, and 6, will take on the burden of the tax with prices going slightly up ($7.14/MMBTU vs. 7.07 $/MMBTU) and consumption relatively unchanged when compared to the MPEC case. Nodes 1 and 2 contain a majority of the Marcellus shale play; hence prices there go up with the imposition of a tax on shale gas. Moreover, US nodes 7 and 8 have high production, and it's profitable for these producers to sell at a lower price in their own market and at a higher price to the other nodes. However, imposing a tax on US Node 7 increases prices at that particular node in 2025 when compared to the MPEC case. Since the shale producer at node 7 is the Stackelberg leader, in this case it can derive more profits by passing the tax onto its own consumption node. Note that the prices under the two tax cases at node 7 (4.94 $/MMBTU and 5.13 $/MMBTU in the ShaleTax and AllTax case, respectively) are still lower than the price for the Base Case (5.72 $/MMBTU, when the shale producer at node 7 is not a Stackelberg player).

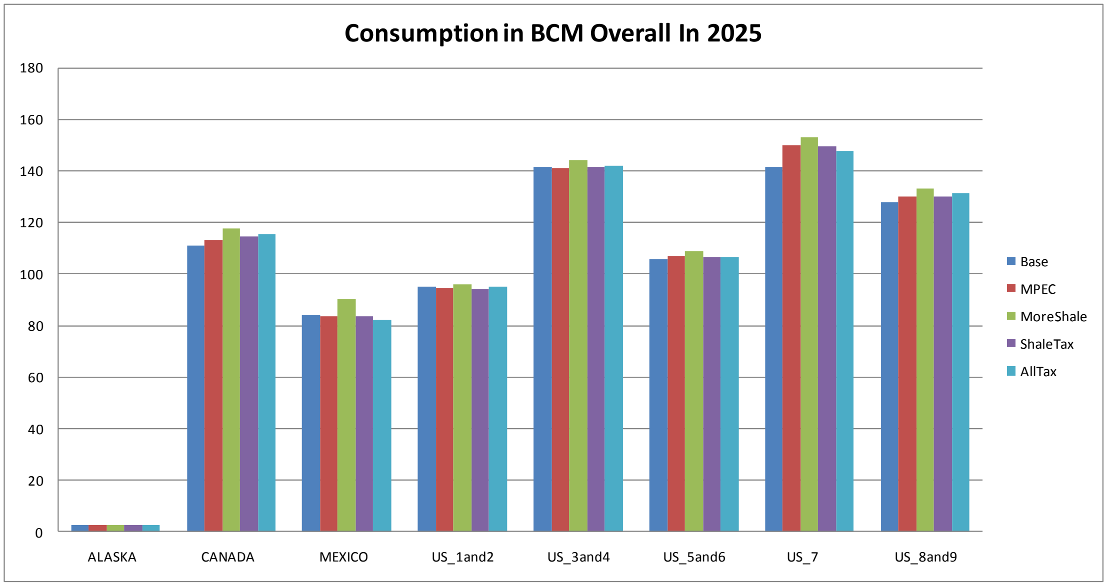

Figure 4.7: Consumption in 2025 as Predicted by the Model

Table 4.11: Average Prices in $/MMBTU in 2025

Region	Base	MPEC	MoreShale	ShaleTax	AllTax
Alaska	6.17	6.17	6.17	6.17	6.17
Canada	6.94	6.69	6.02	6.50	6.34
Mexico	6.66	6.65	5.63	6.70	6.92
US Nodes 1 & 2	8.85	8.91	8.59	8.98	8.88
US Nodes 3 & 4	7.48	7.52	7.02	7.48	7.37
US Nodes 5 & 6	7.36	7.07	6.76	7.14	7.14
US Node 7	5.72	4.88	4.59	4.94	5.13
US Nodes 8 & 9	6.31	6.03	5.65	6.01	5.88

4.5. Summary

This chapter provides a novel way to solve mathematical programs with equilibrium constraints. The new method has been shown to be computationally tractable, and able to solve MPECs where the lower level is a complementarity problem. An extension to solve EPECs is also presented.

The method was first applied to numerical examples for MPECs. It outperformed the method of disjunctive constraints in two ways. First, the selection of the constant L for Algorithm 4.1 did not prove as difficult as the selection of the constant K in disjunctive constraints. Second, with numerical tests the method proved to be computationally quicker than the method of disjunctive constraints. The method was also shown to be able to solve a numerical example of an EPEC, but extensive numerical and theoretical results will be part of future work.

The method was applied to an example of a shale gas producer in the US natural gas market acting as a dominant player. The results show that in the case of a Stackelberg structure, the profits of the producer are not negatively affected with the current proposals for taxes. However, with this structure the producers are able to pass the tax onto the consumer, as profits do not decrease with the implementation of tax but prices do go up. Moreover, if more resources are present, the producer is able to take advantage of them. While in actuality the Stackelberg player might not have such an advantage, this setup helps show how under this scenario, producers can manipulate the market to make decent decisions.

Chapter 5: Solving Discretely-Constrained Mixed Linear Complementarity Problems

5.1. Introduction

This chapter provides solution techniques for DC-MLCPs. In particular, this chapter will consider discretely-constrained Nash games (DC-Nash), where some of the decision variables are constrained to be integer-valued. These games have been formulated as complementary problems (Cottle et al., 2009) in the literature. However, the discretely-constrained versions have often been needed to be solved using inspection; for example, a bimatrix game table (Gibbons, 1996) which has finite, discrete choices to choose from.

When solving DC-MLCPs, it is important to realize that a particular instance might not have integer solutions. A compromise would be to get solutions that are as close to integer as possible. While this chapter does not provide theoretical arguments for the near-integer solutions, the numerical results presented corroborate that the new technique helps achieve integer solutions where appropriate. The method presented in this chapter is shown to be better than the method of (Gabriel et al., 2011a), (Gabriel et al., 2011b) in computational effort and because the method in this dissertation does not require the selection of a constant while the method in (Gabriel et al., 2011a), (Gabriel et al., 2011b) requires the selection of a specified constant.

First, a description of the two relaxation conditions will be given. Then, a general formulation to solve these problems will be provided, based on the work of

(Gabriel et al., 2011a), (Gabriel et al., 2011b). Finally, the methods from Chapter 4 will be applied to solve this resulting two-level problem formulation for numerical examples that are DC-Nash games and discretely-constrained network problems.

A portion of this chapter has been presented in (Gabriel et al., 2011a), (Gabriel et al., 2011b). However, a new way to solve these problems is presented in this dissertation which was not used in the aforementioned papers. Both papers (Gabriel et al., 2011a), (Gabriel et al., 2011b) used disjunctive constraints to solve the DC-MLCPs, but this dissertation uses the technique of SOS Type 1 variables explained in Chapter 4. The examples taken from (Gabriel et al., 2011a), (Gabriel et al., 2011b) are exactly the same as in the papers but the solution technique is different. An extra example with computational time is provided to further support the use of Chapter 4 techniques as opposed to disjunctive constraints which was not discussed in the two papers (Gabriel et al., 2011a), (Gabriel et al., 2011b). Hence, all the examples in this chapter were solved by the solution technique developed in Chapter 4, which was original work that is part of this dissertation. But the problem formulation presented in this chapter and the theory behind the formulation was developed in two papers (Gabriel et al., 2011a), (Gabriel et al., 2011b) and cannot be regarded as original work. The solution technique of Chapter 4, however, proves to be computationally superior for the examples presented.

5.2. Discretely-Constrained Mixed Linear Complementarity Problems

Recall from Chapter 2 that a general, discretely-constrained mixed linear complementarity problem is given the vector $q=(q_1,q_2)^T$ and matrix $A=\begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$, find $z=(z_1,z_2)^T \in \Re^{n_1}\times\Re^{n_2}$ such that

$$
\begin{aligned}
&0 \le q_1 + \begin{pmatrix} A_{11} & A_{12} \end{pmatrix}\begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \perp z_1 \ge 0 \\
&0 = q_2 + \begin{pmatrix} A_{21} & A_{22} \end{pmatrix}\begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \perp z_2, z_2 \text{ free} \\
&(z_1)_c \in \Re_+, c \in C_1, (z_1)_d \in Z_+, d \in D_1 \\
&(z_2)_c \in \Re_+, c \in C_2, (z_2)_d \in Z_+, d \in D_2
\end{aligned}
\tag{5.1}
$$

From this formulation, if $(z_1)_d$ and $(z_2)_d$ were continuous variables, the problem would simplify to a linear complementary problem. Since they are not, one obvious solution is to relax them to be continuous variables, and then solve the problem hoping for an (approximate) integer solution. However, close inspection shows that the complementary conditions also can be relaxed, giving another option for an approximate solution. The next two subsections show how this is done and follows the initial problem description from (Gabriel et al., 2011a), (Gabriel et al., 2011b).

5.2.1 Epsilon-Integrality

Consider the conditions $(z_1)_d \in Z_+, d \in D_1$ and $(z_2)_d \in Z_+, d \in D_2$ from (5.1). Assume these conditions are to be relaxed to make this problem easier. Without loss of generality, consider $(z_1)_d \in Z_+, d \in D_1$ as the arguments for $(z_2)_d \in Z_+, d \in D_2$ are similar. Then, consider a small deviation $\varepsilon_{1ri}, r \in D_1, i = 0,1,..., N$ through which this discrete variable is relaxed. Given any feasible set *M*, the problem becomes to minimize this deviation from integrality while still being in this feasible set. This is formulated by (5.2) below

$$\begin{aligned}
&\min \sum_{i=0}^{N} \sum_{r \in D_1} \left| \varepsilon_{1ri} \right| \\
&-N(1-w_{1ri}) \le (z_1)_r - i - \varepsilon_{1ri} \le N(1-w_{1ri}) \\
&\sum_{i=0}^{N} w_{1ri} = 1 \\
&(z_1)_r \in M \\
&r \in D_1, i = 0,1,..., N \\
&\text{where } w_{1ri} \text{ are SOS Type1 Variables}
\end{aligned} \tag{5.2}$$

In (5.2), the integer value *i* is selected that is closest to a continuous value in *M*. Note, however, that the objective function contains a nonlinear function. This absolute value function can be decomposed into its positive part and negative part so that the objective function is no longer nonlinear as in (5.3).

$$\begin{aligned}
&\min \sum_{i=0}^{N} \sum_{r \in D_1} (\varepsilon_{1ri})^+ + (\varepsilon_{1ri})^- \\
&-N(1-w_{1ri}) \le (z_1)_r - i - \varepsilon_{1ri} \le N(1-w_{1ri}) \\
&\varepsilon_{1ri} = (\varepsilon_{1ri})^+ - (\varepsilon_{1ri})^- \\
&\sum_{i=0}^{N} w_{1ri} = 1 \\
&(z_1)_r \in M \\
&r \in D_1, i = 0,1,..., N \\
&\text{where } w_{1ri} \text{ are SOS Type1 Variables}
\end{aligned} \tag{5.3}$$

This is one way to relax the DC-MLCP. From now on, this will be referred to as ε-integrality. This relaxation, along with another one described next, will be used to help solve DC-MLCPs.

5.2.2. Sigma-Complementarity

From (5.1), consider the complementary condition

$$0 \le q_1 + \begin{pmatrix} A_{11} & A_{12} \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \perp z_1 \ge 0 \tag{5.4}$$

This condition is equivalent to (5.5) below.

$$\begin{aligned} & q_1 + \begin{pmatrix} A_{11} & A_{12} \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \ge 0 \\ & z_1 \ge 0 \\ & \left(q_1 + \begin{pmatrix} A_{11} & A_{12} \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \right)^T z_1 = 0 \end{aligned} \tag{5.5}$$

The goal is to relax the last line equality condition in (5.5). To obtain a solution that is approximate, the last equality need not equal zero but can be very close to zero. In fact, a deviation similar to the one for integrality can be developed here. Consider the deviational vector σ such that the relaxed complementary problem is formulated below[44].

$$\begin{aligned} & q_1 + \begin{pmatrix} A_{11} & A_{12} \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \ge 0 \\ & z_1 \ge 0 \\ & \left(q_1 + \begin{pmatrix} A_{11} & A_{12} \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} - \sigma \right)^T (z_1 - \sigma) = 0 \end{aligned} \tag{5.6}$$

---

[44] Note that different σ's could be used in the last equation in (5.6) as

$$\left(q_1 + \begin{pmatrix} A_{11} & A_{12} \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} - \sigma_1 \right)^T (z_1 - \sigma_2) = 0 \cdot$$

However, this relaxation provides a same solution as the one given in (5.6) because as only one of the factors needs to be 0 for the product to be zero.

Again, the problem becomes to minimize this deviation from complementary whilst still being in this feasible set. This is formulated by (5.7) below

$$
\begin{aligned}
&\min 1^T \sigma \\
&q_1 + \begin{pmatrix} A_{11} & A_{12} \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \geq 0 \\
&z_1 \geq 0 \\
&\left(q_1 + \begin{pmatrix} A_{11} & A_{12} \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} - \sigma \right)^T (z_1 - \sigma) = 0 \\
&\sigma \geq 0
\end{aligned}
\tag{5.7}
$$

The nonlinear equality condition (a product) can be handled the same way the product was handled in Chapter 4. Hence, reformulated to be solved with SOS Type 1 variables, (5.7) becomes

$$
\begin{aligned}
&\min 1^T \sigma \\
&q_1 + \begin{pmatrix} A_{11} & A_{12} \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \geq 0 \\
&z_1 \geq 0 \\
&u - (v^+ + v^-) = 0 \\
&u = \frac{\left(q_1 + \begin{pmatrix} A_{11} & A_{12} \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} - \sigma \right) + (z_1 - \sigma)}{2} \\
&(v^+ - v^-) = \frac{\left(q_1 + \begin{pmatrix} A_{11} & A_{12} \end{pmatrix} \begin{pmatrix} z_1 \\ z_2 \end{pmatrix} - \sigma \right) - (z_1 - \sigma)}{2} \\
&\text{where } v^+, v^- \text{ are SOS 1 variables}
\end{aligned}
\tag{5.8}
$$

5.2.3. Complementarity, Integrality Trade-off

One of the advantages that relaxing both complementary and integrality gives is that we can figure out the tradeoff of relaxing one versus the other. In practice, different solutions can be achieved depending on which of the two (or both) is relaxed. Figure

5.1 shows the idea behind this tradeoff. This can be thought of as the Pareto frontier of a multiobjective programming problem (Cohon, 1978).

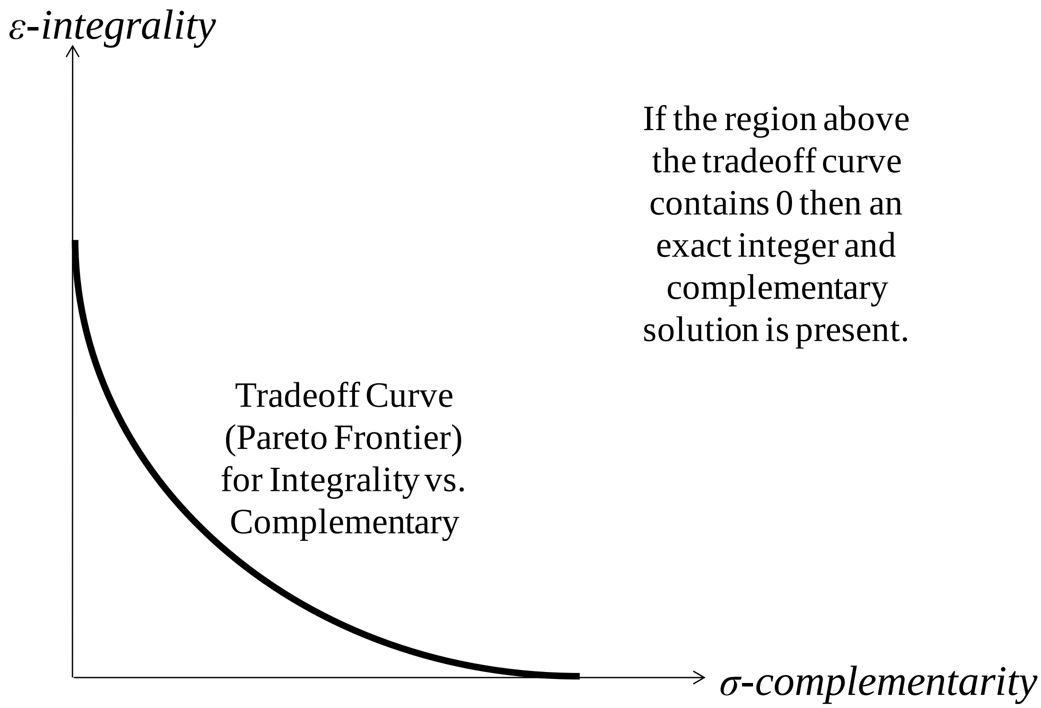

Figure 5.1: The Tradeoff Between Complementary and Integrality

Although the whole Pareto frontier might not have a smooth convex shape as shown in the figure, it is motivation enough to study different variations. Moreover, a point on the tradeoff does not necessarily need to correspond to an integer solution to the DC-MLCP. In the future sections, at least the endpoints of the tradeoff curve (the intersections of the curve with the axes) will be calculated to give an indication of the extent of the Pareto frontier.

5.2.4. Formulation to Solve Discretely-Constrained Mixed Linear Complementary problems

The following formulation is from (Gabriel et al., 2011a) for solving DC-MLCPs except the techniques from Chapter 4 have been used instead of disjunctive constraints[45].

[45] In the following formulation (5.8), $\{w_{1ri}\}_{\mathrm{i}}$ are SOS1 variables implies that $w_{1r1}, w_{1r2}, w_{1r3,\ldots}$ are SOS1 variables.

$$\min\left[\omega_1\left(\sum_{i=0}^{N}\sum_{r\in D_1}(\varepsilon_{1ri})^+ + (\varepsilon_{1ri})^- + \sum_{i=-N_1}^{N_2}\sum_{r\in D_2}(\varepsilon_{2ri})^+ + (\varepsilon_{2ri})^-\right) + \omega_2\left(1^T\sigma\right)\right]$$

$$q_1 + \begin{pmatrix} A_{11} & A_{12} \end{pmatrix}\begin{pmatrix} z_1 \\ z_2 \end{pmatrix} \geq 0$$

$$z_1 \geq 0$$

$$u - (v^+ + v^-) = 0$$

$$u = \frac{\left(q_1 + \begin{pmatrix} A_{11} & A_{12} \end{pmatrix}\begin{pmatrix} z_1 \\ z_2 \end{pmatrix} - 1\sigma^T\right) + \left(z_1 - 1\sigma^T\right)}{2}$$

$$(v^+ - v^-) = \frac{\left(q_1 + \begin{pmatrix} A_{11} & A_{12} \end{pmatrix}\begin{pmatrix} z_1 \\ z_2 \end{pmatrix} - 1\sigma^T\right) - \left(z_1 - 1\sigma^T\right)}{2}$$

$$0 = q_2 + \begin{pmatrix} A_{21} & A_{22} \end{pmatrix}\begin{pmatrix} z_1 \\ z_2 \end{pmatrix}$$

where v^+, v^- are SOS 1 variables (5.8)

$$-N(1-w_{1ri}) \leq (z_1)_r - i - \varepsilon_{1ri} \leq N(1-w_{1ri})$$

$$\varepsilon_{1ri} = (\varepsilon_{1ri})^+ - (\varepsilon_{1ri})^-$$

$$(\varepsilon_{1ri})^+, (\varepsilon_{1ri})^- \geq 0$$

$$\sum_{i=0}^{N} w_{1ri} = 1$$

$$r \in D_1, i = 0,1,\ldots,N$$

where $\{w_{1ri}\}_{\mathrm{i}}$ are SOS Type1 Variables

$$-N_1(1-w_{2ri}) \leq (z_2)_r - i - \varepsilon_{2ri} \leq N_2(1-w_{2ri})$$

$$\varepsilon_{2ri} = (\varepsilon_{2ri})^+ - (\varepsilon_{2ri})^-$$

$$(\varepsilon_{2ri})^+, (\varepsilon_{2ri})^- \geq 0$$

$$\sum_{i=-N_1}^{N_2} w_{2ri} = 1$$

$$r \in D_2, i = -N_1,\ldots,-1,0,1,\ldots,N$$

where $\{w_{2ri}\}_{\mathrm{i}}$ are SOS Type1 Variables

Here, ω_1, ω_2 are predetermined weights attached to ε-integrality and σ-complementarity, respectively. These weights can be used to determine the tradeoff decision between ε-integrality and σ-complementarity.

5.3. Discretely-Constrained Nash-Cournot Games

One way to solve Nash-Cournot games is to first convert them to complementary problems. This requires that the objective and constraint functions be differentiable and that the KKT conditions can be formulated. When some of the variables are integer-valued (e.g., binary yes/no, integer production), the KKT conditions are not valid because the functions are no longer continuous. This section shows an approach by (Gabriel et al., 2011b) that provides a compromise between complementarity and integrality. This is done by first relaxing the discretely-constrained variables to their continuous analogs and taking KKT conditions for this relaxed problem. Gabriel et al. (Gabriel et al., 2011b) converted these KKT conditions to disjunctive-constraints form (Fortuny-Amat & McCarl, 1981) and solved them along with the original integer restrictions re-inserted in a mixed-integer, linear program (MILP). The integer conditions were then further relaxed, but targeted using penalty terms in the objective function. This MILP by (Gabriel et al., 2011b) relaxes both complementarity and integrality but tries to find minimum deviations for both and as such is an example of bi-objective problem (Cohon, 1978). This section will follow the same methodology except the technique of SOS type 1 variables from Chapter 4 will be used instead of disjunctive constraints.

The advantage of the technique presented in this chapter over the formulation with disjunctive constraints is that a large constant, which is essential for the formulation outlined in (Gabriel et al., 2011b), does not need to be selected. Instead SOS Type 1 variables are used as in Chapter 4. This method is also shown to be computationally quicker than the method of disjunctive constraints, and numerical evidence is provided later in this chapter. Note that for all numerical tests presented in this dissertation, the solutions were the same as in (Gabriel et al., 2011a), (Gabriel et al., 2011b).

The next sections provide a formulation based on the one presented in section 5.2. Then, numerical results for two different discretely-constrained Nash games are presented. The first one has discrete restrictions on the production quantities while the second one has discrete startup/shutdown variables.

5.3.1. Formulation of a DC-Nash game by Gabriel et al. (2011b)

For the DC-Nash game, assume there are several Cournot power producers that maximize their profit simultaneously by choosing their optimal production quantities. Their objective function (profit) depends on the production of the competitors through the market demand curve (relationship between the total production and the market price) as well as their own marginal cost. Players p = 1,...,P seek optimal values for their decision vectors $\hat{x}^p \in X^p, p = 1,..., P$ by minimizing their cost functions (or negative profit functions) $f^p(\bullet, x^{-p})$ such that

$$f^p(\hat{x}^p, \hat{x}^{-p}) \le f^p(x^p, \hat{x}^{-p}), \forall x^p \in X^p \tag{5.9}$$

Here $x^p \in \Re^{n_p}$ represents the variables under player p's control with x^{-p} the remaining variables for other players. Also, $\hat{x}^p$ means an equilibrium value to x^p and $X^p = C^p \cap Z_+^{n_p}$ where

$$C^p = \left\{x^p \mid g_j^p(x^p) \le 0,\ j = 1,\dots,I_p; h_k^p(x^p) = 0, k = 1,\dots,E_p; x_q^p \ge 0, q \in S_p\right\} \tag{5.10}$$

and $Z_+^{n_p}$ is the set of nonnegative, integer-valued variables , i.e., $x_r^p \in Z_+, r \in \{1,\dots,n_p\} \setminus S_p$. Here S_p represents those indices for x^p that relate to continuous variables. A continuous relaxation would then be to replace X^p by C^p, i.e., find $\hat{x}^p$, $p = 1,\dots, P$ such that

$$f^p(\hat{x}^p, \hat{x}^{-p}) \le f^p(x^p, \hat{x}^{-p}), \forall x^p \in C^p \tag{5.11}$$

or equivalently find $\hat{x}^p$ that solves

$$\begin{aligned}
&\min_{x^p} f^p(x^p, \hat{x}^{-p})\\
&s.t.\\
&g_j^p(x^p) \le 0\\
&h_k^p(x^p) = 0\\
&j = 1,\dots, I_p, k = 1,\dots, E_p\\
&x_q^p \ge 0\\
&q \in S^p
\end{aligned} \tag{5.12}$$

For the Karush-Kuhn-Tucker (KKT) conditions of (5.12) to be equivalent to solving that optimization problem, the assumption that the functions $f^p(\bullet, x^{-p})$ are convex and a constraint qualification (see (Bazaraa et al., 1993) for generalization of these assumptions that will also lead to KKT conditions being sufficient for optimality) holds (e.g. $g_j^p(x^p), h_k^p(x^p)$ linear) is needed. The KKT conditions for player p's relaxed problem (5.12) are to find $x^p \in \Re^{n_p}, \lambda_j^p \in \Re^{I_p}, \gamma_k^p \in \Re^{E_p}$ such that

$$
\begin{aligned}
&0 \le \nabla_{x^p} f^p(x^p, x^{-p}) + \sum_{j \in I^p} \nabla g_j^p(x^p)\lambda_j^p + \sum_{k \in E^p} \nabla h_k^p(x^p)\gamma_k^p \perp x^p \ge 0 \\
&0 \le -g_j^p(x^p) \perp \lambda_j^p \ge 0 \\
&j = 1,\ldots, I_p \\
&0 = h_k^p(x^p), \gamma_k^p \text{ free} \\
&k = 1,\ldots, E_p
\end{aligned}
\tag{5.13}
$$

Gabriel et al. (2011b) showed that the solution to (5.13) with the discrete restrictions inserted back is the same as the solution to (5.9).

To be able to end up with a linear, mixed-integer program the payoff function $f^p(x^p, x^{-p}) = \frac{1}{2}\begin{pmatrix} x^p \\ x^{-p} \end{pmatrix}^T \begin{pmatrix} N_1^p & N_2^p \\ N_2^p & N_3^p \end{pmatrix}\begin{pmatrix} x^p \\ x^{-p} \end{pmatrix} + (c^p)^T x^p$ is restricted to be quadratic and the constraint functions to be linear. The KKT conditions are

$$
\begin{aligned}
&0 \le \nabla_{x^p} f^p(x^p, x^{-p}) + \sum_{j \in I^p} \nabla g_j^p(x^p)\lambda_j^p + \sum_{k \in E^p} \nabla h_k^p(x^p)\gamma_k^p \perp x^p \ge 0 \\
&g_j^p(x^p) = \left(d_j^p\right)^T x^p - \kappa^p \le 0 \\
&j = 1,\ldots, I_p \\
&h_k^p(x^p) = \left(e_k^p\right)^T x^p - \delta^p = 0 \\
&k = 1,\ldots, E_p
\end{aligned}
\tag{5.14}
$$

Gabriel et al. (2011b) reformulate the continuous relaxation of the original problem (5.9) by using the complementarity problem form of the Nash problem suitably relaxed as in (5.13). These KKT conditions are equivalent to a set of disjunctive constraints of the form:

$$
\begin{aligned}
&0 \le \nabla_{x^p} f^p(x^p, x^{-p}) + \sum_{j \in I^p} \nabla g_j^p(x^p)\lambda_j^p + \sum_{k \in E^p} \nabla h_k^p(x^p)\gamma_k^p \le K_1^p u_1^p \\
&0 \le x^p \le K_1^p(1-u_1^p) \\
&0 \le -g_j^p(x^p) = K_2^p u_{2,j}^p \\
&0 \le \lambda_j^p(x^p) = K_2^p(1-u_{2,j}^p) \\
&j = 1,\ldots, I_p \\
&0 = h_k^p(x^p), \gamma_k^p \text{ free} \\
&k = 1,\ldots, E_p \\
&u_1^p \in \{0,1\}^{n_p} \\
&u_2^p \in \{0,1\}^{I_p}
\end{aligned}
\tag{5.15}
$$

for suitably large values of K_1^p and K_2^p that can be computed as described in (Gabriel et al., 2011b).

An alternative method is to use SOS Type 1 variables as described in Chapter 4, which will be used here so the suitably large values of K_1^p and K_2^p do not need to be computed. This is described below

$$
\begin{aligned}
&0 \le \nabla_{x^p} f^p(x^p, x^{-p}) + \sum_{j \in I^p} \nabla g_j^p(x^p)\lambda_j^p + \sum_{k \in E^p} \nabla h_k^p(x^p)\gamma_k^p \\
&0 \le x^p \\
&u_1^p = \frac{\nabla_{x^p} f^p(x^p, x^{-p}) + \sum_{j \in I^p} \nabla g_j^p(x^p)\lambda_j^p + \sum_{k \in E^p} \nabla h_k^p(x^p)\gamma_k^p + x^p}{2} \\
&\left(v_1^p\right)^+ - \left(v_1^p\right)^- = \frac{\nabla_{x^p} f^p(x^p, x^{-p}) + \sum_{j \in I^p} \nabla g_j^p(x^p)\lambda_j^p + \sum_{k \in E^p} \nabla h_k^p(x^p)\gamma_k^p - x^p}{2} \\
&u_1^p - \left(v_1^p\right)^+ - \left(v_1^p\right)^- = 0 \\
&\text{where} \left(v_1^p\right)^+ - \left(v_1^p\right)^- \text{ are SOS 1 variables} \\
&0 \le -g_j^p(x^p) \\
&0 \le \lambda_j^p(x^p) \\
&u_{2,j}^p = \frac{-g_j^p(x^p) + \lambda_j^p(x^p)}{2} \\
&\left(v_{2,j}^p\right)^+ - \left(v_{2,j}^p\right)^- = \frac{-g_j^p(x^p) - \lambda_j^p(x^p)}{2} \\
&u_{2,j}^p - \left(v_{2,j}^p\right)^+ - \left(v_{2,j}^p\right)^- = 0 \\
&j = 1, \ldots, I_p \\
&0 = h_k^p(x^p), \gamma_k^p \text{ free} \\
&k = 1, \ldots, E_p
\end{aligned}
\tag{5.16}
$$

Using the quadratic form of f^p and the linear forms of g^p and h^p from above, results in the following linear, mixed-integer (with SOS1 variables, that are defined using integers) program with arbitrary objective function $\sum_{p=1}^{P} \left(z^p\right)^T x^p$ and the integer restrictions added back:

$$
\begin{aligned}
&\min_{x^p} \sum_{p=1}^{P} \left(z^p\right)^T x^p \\
&s.t. \text{ for all } p = 1,\ldots,P \\
&0 \le \frac{1}{2}\left(N_1^p + N_1^{p^T}\right)x^p + \frac{1}{2}\left(N_1^p + N_1^{p^T}\right)x^{-p} + c^p + \sum_{j\in I^p} d_j^p \lambda_j^p + \sum_{k\in E^p} e_k^p \gamma_k^p \\
&0 \le x^p \\
&u_1^p = \frac{\frac{1}{2}\left(N_1^p + N_1^{p^T}\right)x^p + \frac{1}{2}\left(N_1^p + N_1^{p^T}\right)x^{-p} + c^p + \sum_{j\in I^p} d_j^p \lambda_j^p + \sum_{k\in E^p} e_k^p \gamma_k^p + x^p}{2} \\
&\left(v_1^p\right)^+ - \left(v_1^p\right)^- = \frac{\frac{1}{2}\left(N_1^p + N_1^{p^T}\right)x^p + \frac{1}{2}\left(N_1^p + N_1^{p^T}\right)x^{-p} + c^p + \sum_{j\in I^p} d_j^p \lambda_j^p + \sum_{k\in E^p} e_k^p \gamma_k^p - x^p}{2} \\
&u_1^p - \left(v_1^p\right)^+ - \left(v_1^p\right)^- = 0 \\
&\text{where } \left(v_1^p\right)^+, \left(v_1^p\right)^- \text{ are SOS 1 variables} \\
&0 \le -\left(\left(d_j^p\right)^T x^p - \kappa^p\right) \\
&0 \le \lambda_j^p (x^p) \\
&u_{2,j}^p = \frac{-\left(\left(d_j^p\right)^T x^p - \kappa^p\right) + \lambda_j^p (x^p)}{2} \\
&\left(v_{2,j}^p\right)^+ - \left(v_{2,j}^p\right)^- = \frac{-\left(\left(d_j^p\right)^T x^p - \kappa^p\right) - \lambda_j^p (x^p)}{2} \\
&u_{2,j}^p - \left(v_{2,j}^p\right)^+ - \left(v_{2,j}^p\right)^- = 0 \\
&\text{where } \left(v_{2,j}^p\right)^+, \left(v_{2,j}^p\right)^- \text{ are SOS 1 variables} \\
&j = 1,\ldots,I_p \\
&0 = \left(e_k^p\right)^T x^p - \delta^p, \gamma_k^p \text{ free} \\
&k = 1,\ldots,E_p \\
&x_r^p \in Z_+, r \in \{1,\ldots,n_p\} \setminus S_p
\end{aligned}
\tag{5.17}
$$

Note that the above problem requires integral restrictions and complementary restrictions to hold at the same time, and may prove to be infeasible (Gabriel et al., 2011b). This is the crucial conversion to a two-level problem. In (5.17), the upper-level has an objective function that is arbitrary. Hence, (5.17) is essentially still a one-level problem, with only the Nash-Cournot game at the bottom level reformulated with SOS1 constraints being equivalent to solving a complementary problem.

The one-level complementary problem can be infeasible, so it needs to be relaxed. The relaxations introduced are the Epsilon-Integrality (Section 5.2.1) and Sigma-Complementary (Section 5.2.2) for the problem to be feasible. Minimizing these deviations can be put in the objective function, making this one-level problem a two-level problem. The lower-level solves a relaxed DC-Nash game while the upper-level minimizes the deviations from complementary and integrality. To ensure that the above reformulation does not have a conflict between complementarity and integrality [46], the following relaxed version of the problem is employed.

[46] We assume the relaxed continuous version of the problem is feasible.

$$
\begin{aligned}
&\min\left[\omega_1\left(\sum_{p=1}^{P}\sum_{i=0}^{N}\sum_{r\in\{1,\dots,n_p\}}\left((\varepsilon_{1ri})^p\right)^+ + \left((\varepsilon_{1ri})^p\right)^-\right) + \omega_2\left(1^T\left(\sigma_1^p + \sigma_2^p\right)\right)\right]\\
&s.t.\text{ for all } p = 1,\dots,P\\
&0 \le \frac{1}{2}\left(N_1^p + N_1^{p^T}\right)x^p + \frac{1}{2}\left(N_1^p + N_1^{p^T}\right)x^{-p} + c^p + \sum_{j\in I^p} d_j^p\lambda_j^p + \sum_{k\in E^p} e_k^p\gamma_k^p\\
&0 \le x^p\\
&u_1^p = \frac{\frac{1}{2}\left(N_1^p + N_1^{p^T}\right)x^p + \frac{1}{2}\left(N_1^p + N_1^{p^T}\right)x^{-p} + c^p + \sum_{j\in I^p} d_j^p\lambda_j^p + \sum_{k\in E^p} e_k^p\gamma_k^p - \sigma_1^p + x^p - \sigma_1^p}{2}\\
&\left(v_1^p\right)^+ - \left(v_1^p\right)^- = \frac{\frac{1}{2}\left(N_1^p + N_1^{p^T}\right)x^p + \frac{1}{2}\left(N_1^p + N_1^{p^T}\right)x^{-p} + c^p + \sum_{j\in I^p} d_j^p\lambda_j^p + \sum_{k\in E^p} e_k^p\gamma_k^p - \sigma_1^p - (x^p - \sigma_1^p)}{2}\\
&u_1^p - \left(v_1^p\right)^+ - \left(v_1^p\right)^- = 0\\
&\text{where } \left(v_1^p\right)^+, \left(v_1^p\right)^- \text{ are SOS 1 variables}\\
&0 \le -\left(\left(d_j^p\right)^T x^p - \kappa^p\right)\\
&0 \le \lambda_j^p(x^p)\\
&u_{2,j}^p = \frac{-\left(\left(d_j^p\right)^T x^p - \kappa^p\right) - \sigma_2^p + \lambda_j^p(x^p) - \sigma_2^p}{2}\\
&\left(v_{2,j}^p\right)^+ - \left(v_{2,j}^p\right)^- = \frac{-\left(\left(d_j^p\right)^T x^p - \kappa^p\right) - \sigma_2^p - \left(\lambda_j^p(x^p) - \sigma_2^p\right)}{2}\\
&u_{2,j}^p - \left(v_{2,j}^p\right)^+ - \left(v_{2,j}^p\right)^- = 0\\
&\text{where } \left(v_{2,j}^p\right)^+, \left(v_{2,j}^p\right)^- \text{ are SOS 1 variables}\\
&j = 1,\dots,I_p\\
&0 = \left(e_k^p\right)^T x^p - \delta^p, \gamma_k^p \text{ free}\\
&k = 1,\dots,E_p\\
&x_r^p \in Z_+, r \in \{1,\dots,n_p\}\setminus S_p\\
&\sigma_1^p, \sigma_2^p \ge 0
\end{aligned}
\tag{5.18}
$$

where the restrictions on the relaxation variables are the same as in (5.8) and given by (5.19)

$$
\begin{aligned}
& -N(1-w_{1ri}) \le \left(x^p\right)_r - i - (\varepsilon_{1ri})^p \le N(1-w_{1ri}) \\
& (\varepsilon_{1ri})^p = \left((\varepsilon_{1ri})^p\right)^+ - \left((\varepsilon_{1ri})^p\right)^- \\
& \left((\varepsilon_{1ri})^p\right)^+, \left((\varepsilon_{1ri})^p\right)^- \ge 0 \\
& \sum_{i=0}^{N} w_{1ri} = 1 \\
& r \in \{1,\ldots,n_p\}, i = 0,1,\ldots,N \\
& \text{where } \{w_{1ri}\}_{\text{i}} \text{ are SOS1 Variables}
\end{aligned}
\tag{5.19}
$$

In the above formulation (5.18)-(5.19), the $(\varepsilon_{1ri})^p$ are used to target the specified integer values (ε-integrality) and σ_1^p, σ_2^p are used to relax complementarity (σ-complementarity), both of which are minimized in the objective function weighting the two objective function parts with positive weights ω_1 and ω_2. Thus, minimizing these deviations helps find an optimal integer solution, as described in (Gabriel et al., 2011a).

5.3.2. First Numerical Example

This section presents the results of numerical examples for solving discretely-constrained Nash-Cournot games from the theory outlined in the previous subsection. The first example constrains the production quantities to be integer while the second example has continuous production quantities but binary startup/shutdown variables. In both examples, seven variations are considered. These variations go through different relaxation techniques and combinations of formulations to be described later. The problems selected can be shown to have unique solutions by simple algebra.

The results show that formulation (5.18)-(5.19) provides solutions to the original discretely-constrained problems. The variations also show that, as stated before, (5.17) can lead to an infeasible solution. Moreover, relaxing complementarity in (5.18)-(5.19) but keeping integer restrictions also leads to a discrete feasible solution. Both numerical examples show that relaxing complementarity is essential to obtaining discrete solutions. Enforcing discrete restrictions, even by integer relaxation, does not help obtain the integer solutions and relaxation of complementary conditions is necessary. A combination of both, as presented in (5.18)-(5.19) helps obtain the required solutions in both cases.

For ease of presentation and comparison but with no loss of generality, consider a Nash-Cournot game with two players (*p* = 1, 2). Given an inverse demand curve *Price* = *a* - *b*(*Quantity*), each player chooses $q_p \in Z_+$ to maximize their profit function

$$Profit_p = Price \times q_p - \left(\beta_p q_p^2 + \rho_p q_p\right) \tag{5.20}$$

where the term in parentheses denotes cost as a function of quantity selected i.e., q_p. The formulation of the game is the same as discussed in the previous subsection.

For the first example, let $a = 6$, $b = 1$, $\beta_1 = \beta_2 = 1$, and $\rho_1 = \rho_2 = 1$, as well as adding capacity constraints for both players of the form

$$q_p \le q_{\max} \tag{5.21}$$

where $q_{max} = 4$. Since only integer-valued production q_p is allowed, a bimatrix payoff table (assuming maximizing payoff) as shown below in Table 5.1 is employed to solve (5.9).

Table 5.1: Bimatrix Nash-Cournot Game, Profits(q_1/q_2)

	0	1	2	3	4
0	(0,0)	(0,3)	(0,2)	(0,-3)	(0,-12)
1	(3,0)	(2,2)	(1,0)	(0,-6)	(-1,-16)
2	(2,0)	(0,1)	(-2,-2)	(-4,-9)	(-6,-20)
3	(-3,0)	(-6,0)	(-9,-4)	(-12,-12)	(-15,-24)
4	(-12,0)	(-16,-1)	(-20,-6)	(-24,-15)	(-28,-28)

Clearly $q_1 = q_2 = 1$ is the unique Nash equilibrium in pure strategies. Another way to solve Nash-Cournot games is by simultaneously solving the problems

$$\begin{aligned}&\max_{q_p}\left[a - b(q_1 + q_2)\right]q_p - \left(\beta_p q_p^2 + \rho_p q_p\right)\\&s.t.\\&q_p \le q_{\max} \quad (\lambda_p \ dual)\\&q_p \ge 0\end{aligned} \tag{5.22}$$

for p = 1, 2. Since the slope of the inverse demand function $b > 0$ and $\beta_p > 0$, the KKT conditions are both necessary and sufficient for solving these problems. These conditions are to find $q_1,q_2,\lambda_1,\lambda_2$ that solve the following linear complementary problem (LCP):

$$\begin{aligned}&0 \le 2q_p(b + \beta_p) + bq_{-p} - (a - \rho_p) + \lambda_p \perp q_p \ge 0\\&0 \le q_{\max} - q_p \perp \lambda_p \ge 0\end{aligned} \tag{5.23}$$

for each p = 1, 2. However, the KKT conditions are only valid if q_p, p = 1, 2 are continuous-valued. Thus, the resulting LCP needs to avoid discrete restrictions on the q_p variables. In this particular example, solving the above LCP after assuming $q_p \in \Re_+$ results in the integer solution $q_1 = 1$, $q_2 = 1$ with *Price* = 4.

However, changing some of the data to a = 9 and ρ_2 = 3 results in a non-integer solution of q_1 = 1.733, q_2 = 1.067, and *Price* = 6.2. But the new bimatrix payoff table for the original discrete version of this game with these new data (Table 5.2), shown below, gives a unique discrete solution of q_1 = 2, q_2 = 1 with *Price* = 6.

Table 5.2: Nash-Cournot Game, Profits(q_1/q_2), (Only Adjustments a=9, ρ_2 = 3)

	0	1	2	3	4
0	(0,0)	(0,4)	(0,4)	(0,0)	(0,-8)
1	(6,0)	(5,3)	(4,2)	(3,-3)	(2,-12)
2	(8,0)	(6,2)	(4,0)	(2,-6)	(0,-16)
3	(6,0)	(3,1)	(0,-2)	(-3,-9)	(-6,-20)
4	(0,0)	(-4,0)	(-8,-4)	(-12,-12)	(-16,-24)

This example shows what can happen if the relaxed LCP does not provide integer-valued answers. Next, more numerical tests are described with the new data a = 9, b = 1, $\beta_1 = \beta_2$ = 1, ρ_1 = 1, and ρ_2 = 3.

The first variation is to solve the continuous version of the LCP (i.e., without any integer restrictions) relating to (5.9) ("MLCP"). Solving the original version of the problem with the integer restrictions relating to (5.9) is variation 2 ("Bimatrix") and is solved by examining the bimatrix payoff table. In the remaining variations to be described, there are two ways of forcing integrality of the solutions. First, the problem can be integer-constrained through the solver (variations 3 and 4) with variation 3 being (5.17) and variation 4 also relaxing complementarity (σ-complementary) in (5.17).

Second, in variation 5, complementarity can be relaxed without constraining the problem to have integer solutions, hence "continuous variables" for the problem description. Hence, we should not expect integer solutions. Finally, in variations 6 and 7, integers can be targeted using the ε deviational variables (5.18)-(5.19) (ε-integrality). In variation 6, no relaxation for complementarity is allowed. Variation 7 allows relaxation for both complementarity and integrality (σ-complementary and ε-integrality). Table 5.3 describes the various possible formulations considered.

Table 5.3: Description of Formulation Variations

Variation	σ-Complementary	ε-Integrality	Problem Description
1	No	No	MLCP
2	No	No	Bimatrix
3	No	No	Integer variables
4	Yes	No	Integer variables
5	Yes	No	Continuous variables
6	No	Yes	Continuous variables
7	Yes	Yes	Continuous variables

5.3.3. Results for First Numerical Example

Tables 5.4 and 5.5 below give the results for this first numerical example.

Table 5.4: Summary of Results (a = 9, b = 1, $\beta_1 = \beta_2 = 1, \rho_1 = 1, \rho_2 = 3$)

Variation	Solution (q_1,q_2)	*Price*	Profits (*P1*, *P2*)
1	(1.733,1.067)	6.2	(6.01, 2.28)
2	(2,1)	6	(6,2)
3	Infeasible	Infeasible	Infeasible
4	(2,1)	6	(6,2)
5	(1.733,1.067)	6.2	(6.01, 2.28)
6	(1.733,1.067)	6.2	(6.01, 2.28)
7	(2,1)	6	(6,2)

Table 5.5: Summary of Results (a = 9, b = 1, $\beta_1 = \beta_2 = 1, \rho_1 = 1, \rho_2 = 3$)

Variation	Sum ε	Sum σ
1	n/a	n/a
2	n/a	n/a
3	n/a	n/a
4	n/a	0.2
5	n/a	0
6	0.334	n/a
7	0	0.2

Table 5.4 shows that a solution to the integer-constrained Nash game is to have q_1 = 2, q_2 = 1 with a resulting price of 6 (variation 2). When the integer restrictions are removed, the solution is then q_1 = 1.733, q_2 = 1.067 with the new

price of 6.2 (variation 1). Solving the mixed integer programming (MIP) version of the problem but forcing exact complementarity and integrality results in an infeasible solution (variation 3) as would be expected. Interestingly, the original integer solution to the Nash problem can be obtained with the MIP approach as long as complementarity is relaxed (variation 4) or when integers are targeted using ε's (without enforcing integrality) along with the complementarity relaxation (variation 7).

It is interesting to note that variation 7 is a validation of the earlier discussion for obtaining integer solutions to DC-Nash. From the perspective of accuracy in attaining the original production values and price, the MIP approach is correct in this instance and thus provides an alternative, viable method for solving such problems. It is interesting to note the difference in results between variations 4 and 5. The former achieves the correct integer solution but directly forces the variables in GAMS to be integer-valued. The latter allows relaxation of complementarity but does not give integer solutions as expected. Furthermore, variation 6 also does not get the correct integer solution even though the using the ε deviational variables were included.

To compare computational time, the formulation of the numerical example above was expanded where the number of players P was increased but the marginal cost for half the players was set the same as player 1 and the other half the same as player 2 from the above example for variation 7. The following Figure 5.2 shows this result with an increase in the number of players for the method of this dissertation compared to the method by (Gabriel et al., 2011b). Again, the method of disjunctive constraints is computationally slower for this example when compared to the SOS1 method.

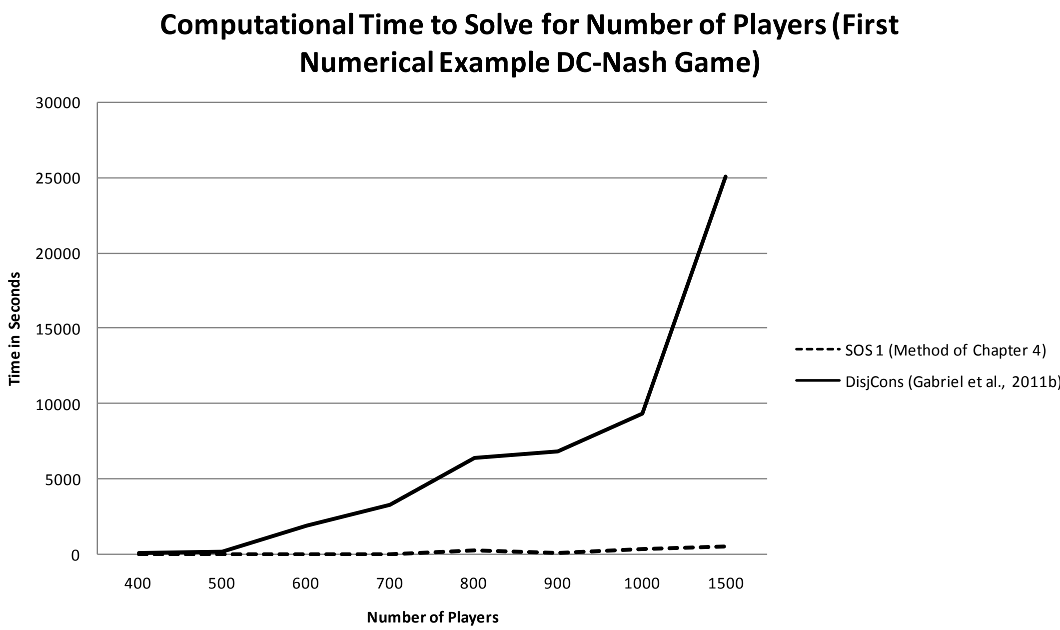

Figure 5.2: Computational Time for First Numerical Example

5.3.4. Numerical Example Relevant to Production Systems

In many applications, the quantities q_p are actually positive real numbers but there are also constraints of the form

$$s_p q_{\min} \leq q_p \leq s_p q_{\max} \tag{5.24}$$

where s_p is a binary variable that is 1 when the player p chooses to produce and 0 when player p chooses to not produce. Here the binary variable s_p might for example relate to the on/off status for a power generation unit. If on, then the minimum and maximum production quantities are in force. If off, then both the upper and lower bounds are equal to zero.

The original capacity constraint is replaced by the one above and the resulting Nash-Cournot game is then solved with $a = 9$, $b = 1$, $\beta_1 = \beta_2 = 1$, $\rho_1 = 1$, $\rho_2 = 3$, $q_{min} =$

1.5, and q_{max} = 4. The binary variables s_p are the ones targeted when complementarity and integrality are relaxed but still allowing for continuous generation variables. The following tables summarize the results.

Table 5.6: Summary of Results (Example Relevant to Production Systems)

Variation	Solution (q_1,q_2)	Binary(s_1, s_2)	Profits ($P1$, $P2$)
1	(1.733,1.067)	(0.347,0.213)	(6.01, 2.28)
2	(1.625,1.5)	(1,1)	(5.28, 2.06)
3	(1.625,1.5)	(1,1)	(5.28, 2.06)
4	(1.625,1.5)	(1,1)	(5.28, 2.06)
5	(1.733,1.067)	(0.347,0.711)	(6.01, 2.28)
6	(1.625,1.5)	(1,1)	(5.28, 2.06)
7	(1.625,1.5)	(1,1)	(5.28, 2.06)

Table 5.7: Summary of Results (Example Relevant to Production Systems)

Variation	*Price*	Sum ε	Sum σ
1	6.2	n/a	n/a
2	5.875	n/a	n/a
3	5.875	n/a	n/a
4	5.875	n/a	0
5	6.2	n/a	0
6	5.875	0	n/a
7	5.875	0	0

The solutions to this example are very different from the previous one. Variation 2 shows the true solution when the variables s_p, p = 1, 2 are forced to be binary. Namely, player 2 produces at the minimum level of 1.5 but player 1 chooses a value of 1.625, in between the minimum and maximum. The continuous relaxation (variation 1) achieves higher profits for both players as would be expected due to less restrictive constraints but does not end up with binary values for the s_p variables. Interestingly, all other variations on relaxation are able to achieve the correct production quantities (q_p) and binary production indicators (s_p) except for variation 5 when only complementarity is relaxed. For this particular problem, forcing integrality is key (through one of the two aforementioned methods) as variations 3, 4, 6, and 7 all give the correct binary solution for s_p, p = 1, 2.

Similar to the previous example, the number of players was increased so that half the players had the data for player 1 and the other half for player 2. The following Figure 5.3 shows this result with an increase in the number of players for the method of this dissertation compared to the method by (Gabriel et al., 2011b), both for variation 7. Again, the method of disjunctive constraints is computationally slower for this example when compared to the SOS1 method. However, this time the advantage of SOS1 is not as strong as for the first numerical example in the previous section. A reason for this can be that since the decision variables are binary, formulating as SOS1 might not have that much of an advantage. This contrasts with the first example where the decision variables were integer.

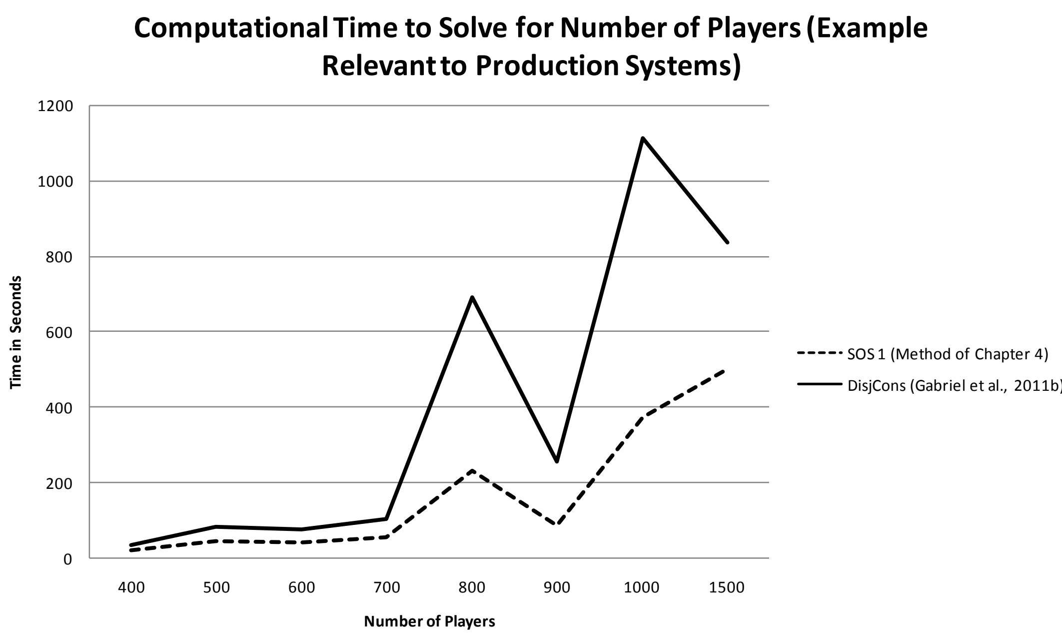

Figure 5.3: Computational Time for Example Relevant to Production Systems

5.4. Discretely-Constrained Network Problems

This section considers discretely-constrained network problems. Note that these problems can be cast as DC-Nash games as well (Cottle et al., 2009). However, it is instructive to look at these network examples separately as well, as extra intuition can be gained from considering transmission lines. The first example is a continuation form the previous section, with two producers. The second example has four producers over two nodes.

5.4.1. First Network Example

Consider a power market with two producers supplying to one demand node as shown in Figure 5.4. Producers 1 and 2 choose to produce quantities q_1 and q_2 respectively, and supply it to meet inelastic demand d, while there are transmission lines (with flow variables q_{12}, q_{13}, q_{23}) between the three nodes. There is a marginal utility of demand c_d and marginal costs c_1 and c_2 for producers 1 and 2, respectively. There is also a market operator who maximizes its own profits by buying from the producers and selling to the consumers.

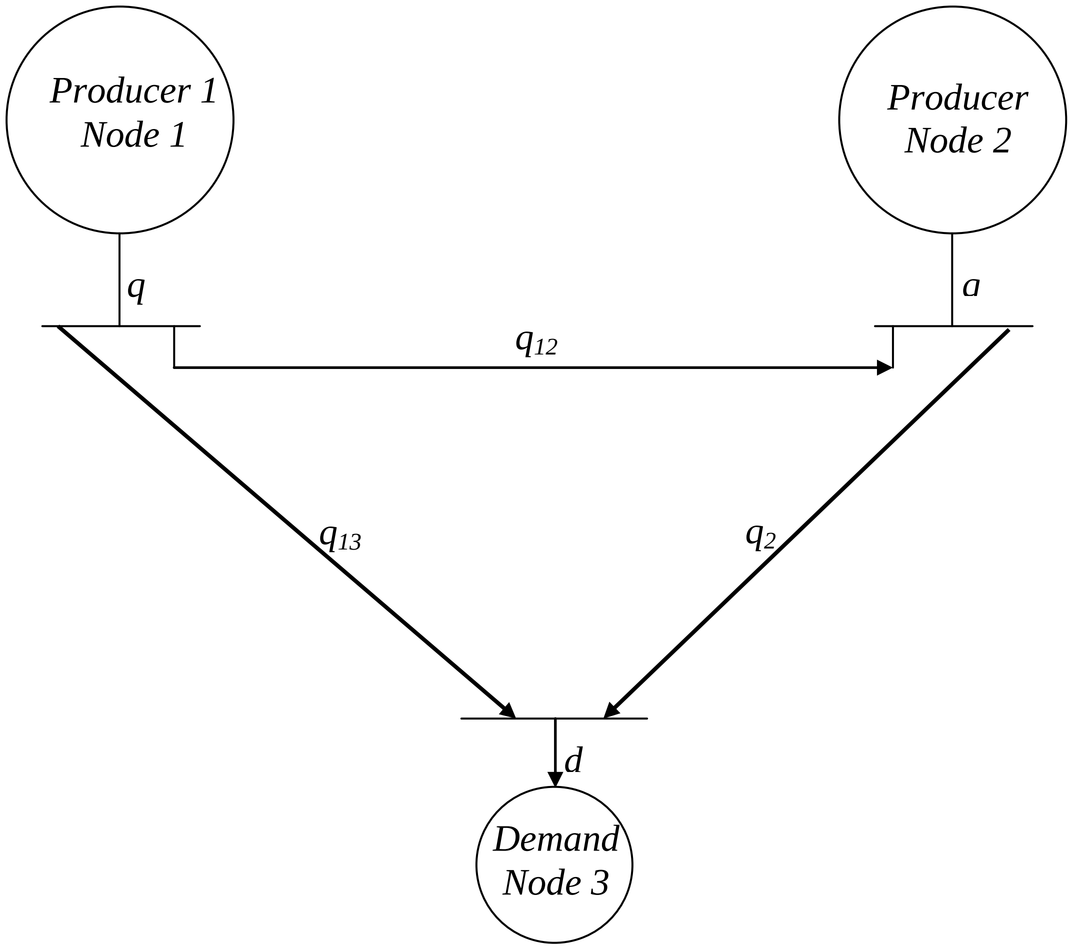

Figure 5.4: Diagram of First Network Example

The producer p (p = 1, 2) solves the following optimization problem

$$
\begin{aligned}
&\min_{q_p} \{ c_p q_p - \lambda_n q_p \} \\
&s.t. \\
&0 \le q_p \le q_{\max} \quad (dual\ \beta_p^{\max})
\end{aligned}
\tag{5.25}
$$

where λ_n is the (endogenous) price at node n. Note that the producer p is active at node $n = p$.

The market operator solves the following optimization problem (with $\bar{q}_p$ introduced to have a square system). The equality constraints set the power flow (q_{13} for example, signifies flow from node 1 to node 3) equal to the power produced and the inequality constraints give a bound on the maximum amount of flow allowed. Flow can be towards the opposite direction as well which is signified by a negative number (i.e., if q_{13} is negative, then the flow is from node 3 to node 1), so the inequalities contain a maximum negative flow as well.

$$
\begin{aligned}
&\min_{\bar{q}_1,\bar{q}_2 d,q_{13},q_{23},q_{12}} \left\{c_1\bar{q}_1 + c_2\bar{q}_2 - c_d d\right\} \\
&s.t. \\
&q_{13} + q_{12} - \bar{q}_1 = 0 \quad (\lambda_1) \\
&q_{23} + q_{12} - \bar{q}_2 = 0 \quad (\lambda_2) \\
&d - q_{13} + q_{23} = 0 \quad (\lambda_3) \\
&-q_{12}^{\min} \le q_{12} \le q_{12}^{\max} \quad (\beta_{12}^{\min}, \beta_{12}^{\max}) \\
&-q_{13}^{\min} \le q_{13} \le q_{13}^{\max} \quad (\beta_{13}^{\min}, \beta_{13}^{\max}) \\
&-q_{23}^{\min} \le q_{23} \le q_{23}^{\max} \quad (\beta_{23}^{\min}, \beta_{23}^{\max}) \\
&\bar{q}_1 = q_1 \\
&\bar{q}_2 = q_2
\end{aligned}
\tag{5.26}
$$

The above optimization problems can be combined to form an MCP, which gives a solution to the game. Our goal here is to see if we restricted the quantities produced and flows to be integer-valued, if we can come up with an equilibrium solution. The following Table 5.8 gives the values of the parameters used for solving this network problem.

Table 5.8: Parameter Values Used in First Network Example

$q_1^{\max}$	$q_2^{\max}$	$q_{12}^{\max}$	$q_{13}^{\max}$	$q_{23}^{\max}$	c_1	c_2	c_d
18	20.5	12	15	15	2	1	5

Hence, producer 2 has a lower marginal cost so will attempt to supply more units of q_2. We use the same process as in the previous section and formulate the problem according the variations in Table 5.9. Note that we are not considering the bimatrix game for this example, so there is no variation 2. Table 5.10 shows the results for the example under different variations.

Table 5.9: Description of Formulation Variations

Variation	σ-Complementary	ε-Integrality	Problem Description
1	No	No	MLCP
3	No	No	Integer variables
4	Yes	No	Integer variables
5	Yes	No	Continuous variables
6	No	Yes	Continuous variables
7	Yes	Yes	Continuous variables

Table 5.10: Solution to Power Market Example

Variations	1	3	4	5	6	7
q_1	9.5	Infeasible	10	9.5	9.5	10
q_2	20.5	Infeasible	20	20.5	20.5	20
q_{12}	-5.5	Infeasible	-5	-5.5	-5.5	-5
q_{13}	15	Infeasible	15	15	15	15
q_{23}	15	Infeasible	15	15	15	15
λ_1	2	Infeasible	2	2	2	2
λ_2	2	Infeasible	2	2	2	2
λ_3	5	Infeasible	5	5	5	5
d	30	Infeasible	30	30	30	30
Sum ε	n/a	n/a	n/a	n/a	1	0
Sum σ	n/a	n/a	0.5	0	n/a	0.5

Note that again, variation 7 gives an integer solution. Comparison to variation 4 is critical, as both of them give the same solution. However, variation 7 provides integer solutions but does not explicitly enforce integrality, while variation 4 requires imposing integer restrictions to get to the answer. Variation 3 proves to be infeasible, while variations 5 and 6 show that only including σ-complementarity or only including ε-integrality is not sufficient to achieve an integer solution for all the variables that are constrained as such. Note that prices at each node (λ_1, λ_2, λ_3) stay the same at each node, regardless of the variation. However, variation 3 did not provide any solution, so not only does variation 7 provide an integer solution; it does so without imposing integer restrictions and also delivering reasonable prices.

5.4.2. Second Network Example

The next example is from (Gabriel et al., 2011a) and depicts an equilibrium in an energy network (e.g., natural gas, electricity) where production, consumption, and transmission of the energy product are analyzed.

Four energy price-taking producers (*A*, *B*, *C*, *D*) are modeled with the first two located at node 1 and the latter two at node 2. The production levels are denoted as q_n^p where node $n \in \{1, 2\}$ and producer $p \in \{A, B, C, D\}$. Similarly, the sales levels are denoted as s_n^p. Lastly, at node 1, the two producers *A* and *B* have the additional option of sending energy to node 2 and f_{12}^A, f_{12}^B represents the associated amounts of flow. (Note that the producers at node 2 are not allowed to ship their product to node 1.)

Both producers *A* and *B* at node 1 have structurally a similar optimization problem shown below just for producer *A*. For node 2, the producers have an optimization that is almost the same as at node 1 with the exception that no flow variables (nor related terms) are included.

$$\begin{aligned}
&\max_{s_1^A, q_1^A, f_{12}^A} \left\{ \pi_1 s_1^A + \pi_2 f_{12}^A - c_1^A\left(q_1^A\right) - \left(\tau_{12}^{REG} + \tau_{12}\right) f_{12}^A \right\} \\
&s.t. \\
&q_1^A \le \bar{q}_1^A \quad (\lambda_1^A) \\
&s_1^A = q_1^A - f_{12}^A \quad (\delta_1^A) \\
&s_1^A \ge 0 \\
&q_1^A \ge 0 \\
&f_{12}^A \ge 0
\end{aligned} \tag{5.27}$$

where

- π_n is the producer price at node $n \in \{1, 2\}$
- $c_1^A\left(q_1^A\right)$ is the (marginal) production cost function assumed to be linear, i.e., $c_1^A\left(q_1^A\right) = \gamma_1^A q_1^A, \gamma_1^A > 0$.
- τ_{12}^{REG} represents the nonnegative, regulated tariff for using the network from node 1 to node 2; τ_{12}^{REG} is a fixed parameter.
- τ_{12} is the congestion tariff for using the network from node 1 to node 2 and a variable from another part of the equilibrium model
- $\overline{q}_1^A$ is the maximum production quantity[47]

Each producer is maximizing their profit (5.27) by choosing appropriate nonnegative levels of production, sales and flow variables subject to not exceeding production limits, and consistency between sales, production, and flow (5.27). The KKT conditions for each of the producers' problems are both necessary and sufficient (Bazaraa et al., 1993) given the functions chosen and these conditions for each of the producers (producer A at node 1, producer B at node 1, producer C at node 2, producer D at node 2) are as follows:

[47] All maximum values for primal variables denoted by an overbar are assumed to be positive as are cost coefficients γ_1^A, γ_1^B, γ_2^C, γ_2^D.

$$
\begin{aligned}
&0 \le -\pi_1 + \delta_1^A \perp s_1^A \ge 0\\
&0 \le \gamma_1^A + \lambda_1^A - \delta_1^A \perp q_1^A \ge 0\\
&0 \le -\pi_2 + \left(\tau_{12}^{REG} + \tau_{12}\right) + \delta_1^A \perp f_{12}^A \ge 0\\
&0 \le \overline{q}_1^A - q_1^A \perp \lambda_1^A \ge 0\\
&s_1^A = q_1^A - f_{12}^A, \delta_1^A \text{ free}
\end{aligned}
\tag{5.28}
$$

$$
\begin{aligned}
&0 \le -\pi_1 + \delta_1^B \perp s_1^B \ge 0\\
&0 \le \gamma_1^B + \lambda_1^B - \delta_1^B \perp q_1^B \ge 0\\
&0 \le -\pi_2 + \left(\tau_{12}^{REG} + \tau_{12}\right) + \delta_1^B \perp f_{12}^B \ge 0\\
&0 \le \overline{q}_1^B - q_1^B \perp \lambda_1^B \ge 0\\
&s_1^B = q_1^B - f_{12}^B, \delta_1^B \text{ free}
\end{aligned}
\tag{5.29}
$$

$$
\begin{aligned}
&0 \le -\pi_2 + \delta_2^C \perp s_2^C \ge 0\\
&0 \le \gamma_2^C + \lambda_2^C - \delta_2^C \perp q_2^C \ge 0\\
&0 \le \overline{q}_2^C - q_2^C \perp \lambda_2^C \ge 0\\
&s_2^C = q_2^C, \delta_2^C \text{ free}
\end{aligned}
\tag{5.30}
$$

$$
\begin{aligned}
&0 \le -\pi_2 + \delta_2^D \perp s_2^D \ge 0\\
&0 \le \gamma_2^D + \lambda_2^D - \delta_2^D \perp q_2^D \ge 0\\
&0 \le \overline{q}_2^D - q_2^D \perp \lambda_2^D \ge 0\\
&s_2^D = q_2^D, \delta_2^D \text{ free}
\end{aligned}
\tag{5.31}
$$

In addition to the KKT conditions for the four producer problems, there are market-clearing conditions that force supply to equal demand:

$$
\begin{aligned}
&0 = \left[s_1^A + s_1^B\right] - D_1(\pi_1), \quad \pi_1 \text{ free}\\
&0 = \left[s_2^C + s_2^C + f_{12}^A + f_{12}^B\right] - D_2(\pi_2), \quad \pi_2 \text{ free}
\end{aligned}
\tag{5.32}
$$

Note that the terms in square brackets are the net supply at each node (assuming no losses) and $D_n(\pi_n)$, n = 1, 2 are the nodal demand as a function of the price π_n. While the producers depicted above operate using the network, there is additional a transportation system operator (TSO) who manages the congestion and flows. The TSO's linear program is as follows (where other objectives are also possible):

$$\begin{aligned}
&\max_{g_{12}} \left(\tau_{12}^{REG} + \tau_{12}\right) g_{12} - c^{TSO}(g_{12}) \\
&s.t. \\
&g_{12} \le \bar{g}_{12} \quad (\varepsilon_{12}) \\
&g_{12} \ge 0
\end{aligned} \tag{5.33}$$

Here, the TSO controls the variable g_{12} which is the flow from node 1 to node 2, $c^{TSO}(g_{12})$ is a network operations cost function (assumed linear i.e., $c^{TSO}(g_{12}) = \gamma^{TSO} g_{12}, \gamma^{TSO} > 0$) and g_{12} is the capacity of the link between nodes 1 and 2. The KKT conditions for this problem are both necessary and sufficient and since it is a linear program and these conditions are the following:

$$\begin{aligned}
&0 \le -\tau_{12}^{REG} - \tau_{12} + \gamma^{TSO} + \varepsilon_{12} \perp g_{12} \ge 0 \\
&0 \le \bar{g}_{12} - g_{12} \perp \varepsilon_{12} \ge 0
\end{aligned} \tag{5.34}$$

The last part of the equilibrium problem is the market-clearing conditions that balance the flow controlled by the network operator and thus by producers A and B:

$$0 = g_{12} - \left[f_{12}^{A} + f_{12}^{B} \right], \tau_{12} \text{ free} \tag{5.35}$$

The LCP for this energy network problem is thus the KKT conditions of the producers: (5.28), (5.29), (5.30), (5.31), the nodal market-clearing conditions (5.32), the KKT conditions of the TSO (5.34) and the market-clearing conditions of the transportation market (5.35). Figure 5.5 below shows a diagrammatic representation of this network.

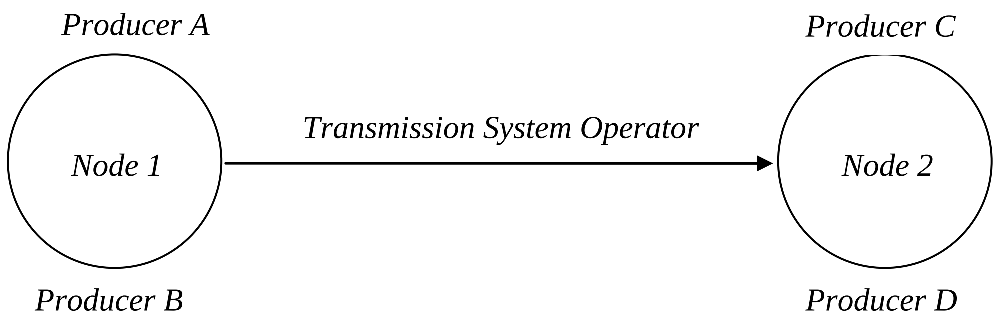

Figure 5.5: Representation of Second Network Example

In this problem, s_1^A, s_1^B, s_2^C, s_2^D, q_1^A, q_1^B, q_2^C, q_2^D are the variables that are integer-constrained in variations 3 and 4. The goal is to find a solution which has these variables as integers. Note that there are multiple integer solutions. The values for the input parameters as well as the six variations that were tested are shown in Table 5.11 below.

Table 5.11: Dataset Used in Second Network Example

Parameter	Value
τ_{12}^{REG}	0.5
γ_1^A	10
γ_1^B	12
γ_2^C	15
γ_2^D	18
a_1	20
b_1	1
a_2	40
b_2	2
$\bar{q}_1^A$	10
$\bar{q}_1^B$	10
$\bar{q}_2^C$	4.5
$\bar{q}_2^D$	5
g_{12}	15
γ^{TSO}	1

Table 5.12 reports the different variations considered. Again, using SOS Type 1 variables from Chapter 4, this DC-MLCP can be converted to a two-level problem and then solved. Note the variations are similar to the previous network example.

Table 5.12: Description of Formulation Variations for Second Network Example

Variation	σ-Complementary	ε-Integrality	Problem Description
1	No	No	MLCP
3	No	No	Integer variables
4	Yes	No	Integer variables
5	Yes	No	Continuous variables
6	No	Yes	Continuous variables
7	Yes	Yes	Continuous variables

As in the DC-Nash example, several numerical variations were done to see the change in solutions. Variation 1 was a mixed-complementary problem (MCP) without imposing integer restrictions. Variation 3 involved converting the MCP to a formulation with disjunctive constraints but restricting the variables of production and sales to be integer. The rest of the variations then go through the different combinations as in the DC-Nash example.

First, variations 4 and 7 give an integer solution. However, due to the presence of multiple equilibria, these solutions need not be unique as in the DC-NASH game. Multiple starting points were chosen, and, according to the numerical tests, the reported solution had the highest objective function value (along with some other equilibria not reported) when a feasible integer solution was desired. Hence, variations 4 and 7 can be used to obtain optimal, integer solutions that are feasible. Note that variation 6 targets integers through ε-complementarity, while variation 4 actually restricts solutions to integer values.

Similar to the previous example, variation 1 yielded a non-integer but optimal and feasible solution while variation 3 was infeasible. Again, this shows that σ-complementarity is essential to obtain a feasible integer solution (as in variations 4 and 7). However, only σ-complementarity is not enough to obtain integer solutions (variation 5) nor is only ε-complementarity (variation 6).

The extra advantage of using variations 4 and 7 is that values of dual variables can be obtained and interpreted. It is interesting to note that the dual variables change from the continuous to the integer case, which is what was expected. However, it also shows the differences in solutions with relaxation of integer variables to solve a problem and how it leads to solutions that can be very different from the market dynamics of an integer constrained problem. Tables 5.13 and 5.14 below display the results obtained from this network example.

Table 5.13: Results for Second Network Problem (Integer Variables)

Variations	1	3	4	5	6	7
s_1^A	7.440	Infeasible	8.000	8.000	8.000	8.000
s_1^B	0.560	Infeasible	0	0	0	0
s_2^C	4.500	Infeasible	4.000	4.500	4.500	4.000
s_2^D	0	Infeasible	0	0	0	0
q_1^A	10.000	Infeasible	10.000	10.000	10.000	10.000
q_1^B	3.000	Infeasible	3.000	3.000	3.000	3.000
q_2^C	4.500	Infeasible	4.000	4.500	4.500	4.000
q_2^D	0	Infeasible	0	0	0	0
Sum σ	n/a	n/a	0.5	0	n/a	0.5
Sum ε	n/a	n/a	n/a	n/a	1.000	0

Table 5.14: Results for Second Network Problem (Other Variables)

Variations	1	3	4	5	6	7
f_{12}^{A}	2.560	Infeasible	2.000	2.000	2.000	2.000
f_{12}^{B}	2.440	Infeasible	3.000	3.000	3.000	3.000
λ_1^{A}	2.000	Infeasible	2.000	2.000	2.000	2.000
λ_1^{B}	0	Infeasible	0	0	0	0
λ_2^{C}	0.250	Infeasible	0.500	0.250	0.250	0.500
λ_2^{D}	0	Infeasible	0	0	0	0
g_{12}	5.000	Infeasible	5.000	5.000	5.000	5.000
ε_{12}	2.250	Infeasible	2.500	2.250	2.250	2.500
δ_1^{A}	12.000	Infeasible	12.000	12.000	12.000	12.000
δ_1^{B}	12.000	Infeasible	12.000	12.000	12.000	12.000
δ_2^{C}	15.250	Infeasible	15.500	15.250	15.250	15.500
δ_2^{D}	17.581	Infeasible	18.000	18.000	18.000	18.000
π_1	12.000	Infeasible	12.000	12.000	12.000	12.000
π_2	15.250	Infeasible	15.500	15.250	15.250	15.500
τ_{12}	2.750	Infeasible	3.000	2.750	2.750	3.000

5.5. Summary

This chapter improves a methodology to solve discretely-constrained Nash games formulated as mixed complementarity problems. The discrete restrictions can lead to infeasible solutions, so a relaxation is needed. Along with providing both a complementary and integrality relaxation, this chapter uses the technique from Chapter 4 to solve the resulting DC-MLCP.

The two-level formulation proposed in this chapter gives a new way to look at an otherwise one-level problem. First, this formulation shows that there are actually two different sets of optimization problems hidden in this one-level problem. One set of minimization problems aims to minimize deviations from complementary and integrality. The other set of complementary problems aims to solve either a Nash-Cournot game or a network problem (as the two examples discussed in this chapter; there can be other applications). In this way, the two-level approach tackles the problem from a different perspective. This perspective, coupled with the solution technique from Chapter 4, ends up performing better than the single-stage method of (Gabriel et al., 2011a), (Gabriel et al., 2011b).

From the theoretical analysis carried out and the examples considered, several conclusions can be drawn. First, relaxing both integrality and complementarity in the lower-level problem while using the upper-level to minimize deviations enables the selection of an integer, equilibrium solution. Second, the method of SOS Type 1 variables from Chapter 4 proves to be computationally quicker than the method of disjunctive constraints for two of the numerical examples illustrated in this chapter. Third, different variations of relaxation, as shown by Variations 1 through 7 for each example, can lead to different solutions. It also helps analyze the importance of the technique to practical examples.

Chapter 6: Conclusions

This chapter provides a summary of the work done in the dissertation. The dissertation went through three different types of two-level problems, and provided novel solution techniques for each of them. Several applications of these techniques were provided to show the nuances of each method. This chapter will start off with concluding remarks about each two-level problem studied. Then, the main contributions of this dissertation will be listed. Finally, some proposals for future work as an extension of the work provided in this dissertation will be presented.

6.1. Concluding Remarks

6.1.1. Robust Optimization

Numerous robust optimization techniques exist in the literature, but the goal of this dissertation was to develop a technique which is numerically more efficient than previous techniques. Chapter 3 presented a method based on Benders decomposition, which was shown to be numerically more efficient when compared to a previous method (Li et al., 2006).

The modified Benders method was also shown to be computationally tractable, in that empirically the increase in the number of function calls was at most linear with an increase in variables. Diverse numerical examples were provided to show the applicability of the method to different types of problems.

Previous methods exist, which could easily solve linear and quadratic robust optimization problems efficiently. But many engineering design problems, as also stated in this dissertation, involve nonlinear constraints and objective functions. The modified Benders method was shown to be able to tackle these problems, and a sampling technique was provided so the user could choose the level of accuracy desired. In particular, the engineering design examples showed how the selection of an optimal design can vary with the presence of uncertainty. Moreover, the examples also showed how the presence of uncertainty degrades objective function performance.

A final example showed the importance of studying uncertainty to environmental market strategies. The future of a carbon tax and retrofitting technology is uncertain, and this uncertainty in future events can have important implications on decisions made today. As shown in the example, the uncertainty of a tax discourages energy intensive infrastructure for today, when the decision maker is extremely risk averse. Worries of a larger tax in the future encourages infrastructure to not be as energy intensive under a worst-case scenario.

The approach presented in this dissertation is designed for robust optimization problems with a goal to decrease computational time. One drawback of this approach is that there is no built-in verification that the solution is actually robust for nonlinear robust optimization problems. Other methods (Gunawan & Azarm, 2004), (Li et al., 2006) have optimization problems that verify the solution is robust within the approach. Secondly, this approach requires explicit objective and constraint functions to be able to work. Simulation or "black box" type problems will not be solved using the approach presented in this dissertation. Moreover, this dissertation only provides numerical evidence that the modified Benders method works for robust optimization problems with quasiconvex constraints. A mathematical proof would improve the validity of the method and is an option for future work. Another area of improvement is a better sampling technique than the one presented for nonlinear constraint functions. Improvements in sampling and theory might eventually lead to a solution technique for general nonlinear robust optimization problems.

6.1.2. MPECs and EPECs

Chapter 4 describes a new solution technique for MPECs and EPECs, which was developed to solve large-scale problems such as the North American gas model MPEC. The new technique was developed to be more computationally efficient than previous techniques for solving these problems. A lot of times optimization solvers return a solution as infeasible if they are unable to find one for complex problems. The aim was to develop a simple enough technique that could be applied to a wide variety of problems.

The two algorithms presented in Chapter 4 were shown to solve large MPECs with much less computational effort when compared to disjunctive constraints and also be applicable to complex problems (e.g., the North American gas model) where traditional methods had failed. The methods were restricted to be used in problems where the KKT conditions were necessary and sufficient, which decreases the applicability of various functional forms but still lends itself to different examples.

The focus of the examples was on different types of Stackelberg games and Nash-Cournot games with a modified structure. The theme of the examples was energy and natural gas production, outlined by the North American gas model.

Various scenarios for shale gas in the United States were developed using the North American gas model. The scenarios studied what would happen under a tax for shale production, a tax on all natural gas production, and the presence of more shale than predicted. One of the main conclusions was that in the presence of a tax, the producers pass the tax onto the consumers. Moreover, the top-level firm always makes the majority of the profits by manipulating the market. The reality of the market situation probably lies somewhere in between that of a Stackelberg game and a Nash-Cournot game, but being able to study this formulation was instructive.

6.1.3. Discretely-Constrained Mixed Linear Complementarity Problems

The technique presented in Chapter 5 was a new way to solve and think about discretely-constrained mixed linear complementary problems. The technique provided a way to solve a relaxed version of the problem in one stage, thus converting a two-level problem into one level. This conversion into one stage was initially achieved using disjunctive constraints, but this meant the solution would depend on a large constant. This dissertation used the techniques presented in Chapter 4 to not have to use disjunctive constraints when finding solutions to DC-MLCPs.

The first set of numerical examples studied discretely-constrained Nash-Cournot games. While the continuous versions of these games have been extensively studied, imposing discrete restrictions might lead to infeasibility. Hence, relaxing the integer restrictions as well as the complementary conditions, while targeting specific integer values, provided a solution to these games. The computational effort was also abated using SOS Type 1 variables as opposed to disjunctive constraints. In all examples, the payoffs for the players in the continuous relaxation were higher than with the integer restrictions. Two different variations were provided to be able to obtain integer solutions to DC-Nash games.

The second set of numerical examples studied discretely-constrained network problems. Network problems can also be expressed as complementary problems, and adding the discrete restrictions would yield the same problems of infeasibility as the discretely-constrained Nash games. Again, different variations were studied to see which one yielded an integer solution. Two variations, the same ones that worked for the discretely-constrained Nash games, worked for these network problems as well. Dual variables are often used in network problems to obtain shadow prices, and this technique helped obtaining these prices. However, the applicability of these prices is still a matter of debate as the complementary and integrality relaxations also factor into these prices.

Another advantage of using the technique outlined in this dissertation was that a tradeoff between complementary and integrality can be obtained. This was numerically shown by studying different variations in Chapter 5.

6.2. Main Contributions

This dissertation is focused on solving three specific types of two-level problems. However, these three types of problems have been chosen to be the ones that best encompass the class of two-level problems. First, robust optimization is a two-level problem where the lower-level can be thought of as checking the feasibility of an upper-level decision. In this way, the lower-level aims to check feasibility, but does not have an objective or goal for itself. For MPECs and EPECs, the lower-level is either a cooperative or noncooperative equilibrium may or may not conflict with the upper-level objective. While the lower-level alters the feasible space for the upper-level problem, the focus is on influencing the objective function of the upper-level problem. Thus, these two types of two-level problems encompass dealing with influence of the lower-level on constraints (robust optimization) and objective function (MPECs) of the upper-level directly, and indirectly the objective function (robust optimization) and constraints (MPECs) of the upper-level. Finally, the third type of problem is something which starts off with a one-level structure, but is converted to two levels to be able to solve more easily.

The first main contribution of this dissertation is the application of decomposition techniques to two-level problems, which helps convert them to a single one-level problem (as in the case of MPECs, EPECs, and DC-MLCPs), or a series of one-level problems that can be solved iteratively (as in the case of robust optimization). This use of decomposition techniques provides insight that could not be achieved through a two-level analysis, for example, the robust feasible region for robust optimization problems, the absolute value function equality in MPECs and EPECs, and obtaining shadow prices from DC-MLCPs. These decomposition techniques are presented in a way to take advantage of the problem structure, and obtain a solution that can relate to the original problem.

The second main contribution of this dissertation is to provide methods that greatly speed up computation time for two-level problems. Robust optimization problems have been traditionally solved using a nested inner-outer structure which takes a lot of computational effort. MPECs and EPECs have been solved using primarily disjunctive constraints which not only involve great computational effort because of the presence of binary variables, but also require the selection of a large constant which is not immediately obvious. DC-MLCPs have been solved either successively fixing and relaxing discrete variables or using disjunctive constraints, both of which are computationally more expensive than the methods provided in this dissertation. In particular, the method for MPECs can be applied to a host of other problems to speed up computation wherever a product of two terms resulting in a nonlinear function is present.

The third main contribution of this dissertation is applying the theory to an extremely diverse set of examples. The dissertation contains examples from environmental markets, energy markets, power systems, structural optimization, engineering design, networks, and game theory. These same examples can also be split into academic subjects of operations research, economics, mechanical engineering, and market design. A host of such examples serves the academic community well, as it outlines the importance of research into the theory of two-level problems.

6.3. Future Research

6.3.1. Multiobjective Mixed-Integer Robust Optimization

There are two natural ways to develop the ideas presented for robust optimization. These ideas arise out of the methods developed in the dissertation, and it is convenient that this direction is shared by current research as well.

Many engineering design applications involve multiobjective optimization. Thus, extending the modified Benders method to be applicable to multiobjective robust optimization problems would be useful. Since the modified Benders method has already converted the two-level problem into a single-level, combining it with traditional methods of multiobjective optimization would be natural. For example, if the robust feasible region is provided, any multiobjective method can be applied. Hence, each stage of the modified Benders decomposition method can involve solving a multiobjective problem. Since the modified Benders method is gradient-based, it would make sense to combine it with another gradient-based method such as Normal Boundary intersection or one of its variations (Siddiqui et al., 2011d). Each step of the Normal Boundary intersection method provides one point on the Pareto frontier. The modified Benders method would be used at each step to come up with one robust Pareto point, thus generating a robust Pareto frontier.

The second natural extension has to do with solving mixed-integer robust optimization problems. Standard Benders decomposition is already applicable to mixed-integer optimization problems. Thus, a variation can easily be considered which contains standard Benders cuts and modified Benders cuts to solve a robust mixed-integer optimization problem. These two ideas can then be combined to solve a mixed-integer robust optimization problems.

6.3.2. Solving Nonlinear MPECs and EPECs

The methods presented in this dissertation were only applicable to MPECs and EPECs which comprised of optimization problems where the KKT conditions were necessary and sufficient. However, to obtain local solutions to nonlinear programs an approximation scheme can be developed where the lower level problem is locally approximated. This can be done using SOS Type 2 variables, and the linear interpolation could have the KKT conditions necessary and sufficient within a specified interval.

An easier task would be to consider the case where the KKT conditions might just be necessary (or sufficient) and develop an approximation scheme from there. In particular, if the product of two terms becomes complicated, other approximation techniques may be studied.

6.3.3. Solving Large-Scale Mixed-Integer Complementary Problems

The relaxation techniques put into the DC-MLCPs in Chapter 5 were not put to the test on larger problems. There might be even better ways to approximate the relaxation of complementarity. For example, a nonlinear function describing the product might be added as a constraint.

In many cases, the formulation might yield a simple way to both add relaxations and approximate the lower-level product at the same time. This can then be tested on large mixed-integer complementarity problems, solutions to which can be very useful when studying market or network dynamics problems.

Appendices

Appendix A: Robust Optimization Test Problems

For examples 2 to 4: $q_1 = -1, q_2 = -1, \Delta q_{1`} = 0.1, \Delta q_{2`} = 0.1, \Delta x_{3`} = 0.1$.

(Example 2)

$$\begin{aligned}
&\min_x \left((x_1 - 0.6)^2 + (x_2 - 0.6)^2 - x_3 - x_4 + 10 \right) \\
&s.t. \\
&(q_1 + \hat{q}_1) + x_1 + x_2 \le 0 \\
&(q_2 + \hat{q}_2) + x_3 + x_4 \le 0 \\
&-x_1 \le 0, -x_2 \le 0, -x_3 \le 0, -x_4 \le 0 \\
&where: \ \forall \hat{q}_1 \in \{-\Delta q_1, \Delta q_1\}, \forall \hat{q}_2 \in \{-\Delta q_2, \Delta q_2\}
\end{aligned} \tag{A1}$$

Table A1: Solution to Example 2

Information	Nominal Solution	Robust Solution	Li et al.'s (2006) Solution
x_1	0.5	0.45	0.375
x_2	0.5	0.45	0.375
x_3	1	0.90	0.416
x_4	0	0	0.416
$f(x)$	9.02	9.145	9.268
Function Calls	7	19	2592

(Example 3)

$$\begin{aligned}
&\min_x \left(\left(x_1 - 0.6\right)^2 + \left(x_2 - 0.6\right)^2 - \tilde{x}_3 - x_4 + 10 \right) \\
&s.t. \\
&(q_1 + \hat{q}_1) + x_1 + x_2 \le 0 \\
&(q_2 + \hat{q}_2) + \tilde{x}_3 + x_4 \le 0 \\
&-x_1 \le 0, -x_2 \le 0, -\tilde{x}_3 \le 0, -x_4 \le 0 \\
&where: \ \forall \hat{q}_1 \in \left\{-\Delta q_1, \Delta q_1\right\}, \forall \hat{q}_2 \in \left\{-\Delta q_2, \Delta q_2\right\}, \forall \tilde{x}_3 \in \left\{x_3 - \Delta x_3, x_3 + \Delta x_3\right\}
\end{aligned} \tag{A2}$$

Table A2: Solution to Example 3

Information	Nominal Solution	Robust Solution	Li et al.'s (2006) Solution
x_1	0.5	0.45	0.375
x_2	0.5	0.45	0.375
x_3	1	0.10	0.416
x_4	0	0.70	0.416
$f(x)$	9.02	9.145	9.268
Function Calls	7	21	2808

(Example 4)

$$\begin{aligned}
&\min_x \left(\left(x_1 - 0.6\right)^2 + \left(x_2 - 0.6\right)^2 - \tilde{x}_3 x_4 + 10 \right) \\
&s.t. \\
&(q_1 + \hat{q}_1) + x_1 + x_2 \le 0 \\
&(q_2 + \hat{q}_2) + \tilde{x}_3 + x_4 \le 0 \\
&-x_1 \le 0, -x_2 \le 0, -\tilde{x}_3 \le 0, -x_4 \le 0 \\
&where \quad \forall \hat{q}_1 \in \left\{-\Delta q_1, \Delta q_1\right\}, \forall \hat{q}_2 \in \left\{-\Delta q_2, \Delta q_2\right\}, \forall \tilde{x}_3 \in \left\{x_3 - \Delta x_3, x_3 + \Delta x_3\right\}
\end{aligned} \tag{A3}$$

Table A3: Solution to Example 4

Information	Nominal Solution	Robust Solution	Li et al.'s (2006) Solution
x_1	0.5	0.45	0.40
x_2	0.5	0.45	0.40
x_3	1	0.90	0.40
x_4	0	0	0.40
$f(x)$	9.77	9.8850	9.92
Function Calls	7	21	2592

(Example 5)

$\Delta x = \Delta p = 0.1$

$$
\begin{aligned}
&\min_x \left(2x_1 + 3x_2 - 5x_3 - 2x_4 + 3x_5\right) \\
&s.t. \\
&\tilde{x}_1 + \tilde{x}_2 - \tilde{2}x_3 - x_4 + 3x_5 - 1 \geq 0 \\
&2\tilde{x}_1 - 2\tilde{x}_2 + 3x_3 - x_4 + x_5 - 3 \geq 0 \\
&-5 \leq \tilde{x}_i \leq 5, \quad i = 1,2 \\
&-5 \leq x_i \leq 5, \quad i = 3,4,5
\end{aligned}
\tag{A4}
$$

Table A4: Solution to Example 5

Information	Nominal Solution	Robust Solution	Li et al.'s (2006) Solution
x_1	-4.00	-3.40	1.04
x_2	-5.00	-4.90	-4.53
x_3	5.00	5.00	5.00
x_4	-5.00	-5.00	-0.37
x_5	5.00	5.00	5.00
$f(x)$	-23.00	-21.50	-20.75
Function Calls	5	17	7856

(Example 6)

$\Delta x = \Delta p = 0.1$

$$\begin{aligned}
&\min_{x}\left(2.1x_1 + 3.07x_2 - 5x_3 - 2x_4 + 2.4x_5\right)\\
&s.t.\\
&0.9\tilde{x}_1 + \tilde{x}_2 - \tilde{2}.2x_3 - 1.1x_4 + 3.5x_5 - 1.2 \geq 0\\
&2\tilde{x}_1 - 2\tilde{x}_2 + 3x_3 - x_4 + x_5 - 10 \geq 0\\
&-5 \leq \tilde{x}_i \leq 5, \quad i = 1,2\\
&-5 \leq x_i \leq 5, \quad i = 3,4,5
\end{aligned} \tag{A5}$$

Table A5: Solution to Example 6

Information	Nominal Solution	Robust Solution	Li et al.'s (2006) Solution
x_1	-5.00	-4.90	-4.23
x_2	-5.00	-4.90	-4.52
x_3	5.00	5.00	5.00
x_4	-3.82	-4.27	-3.67
x_5	5.00	5.00	5.00
$f(x)$	-31.21	-29.79	-28.36
Function Calls	5	17	11099

(Hock 100)

This is problem 100 modified from (Hock & Schittkowski, 1980). $\Delta x = \Delta p = 0.1$

$$\min_x \left((\tilde{x}_1 - 10)^2 + 5(\tilde{x}_2 - 12)^2 + x_3^{\ 4} + 3(x_4 - 11)^2 + 10x_5^{\ 6} + 7x_6^{\ 2} + x_7^{\ 4} - 4x_6 x_7 - 10x_6 - 8x_7 \right)$$
$$s.t.$$
$$1\tilde{2}7 - 2\tilde{x}_1^{\ 2} - 3\tilde{x}_2^{\ 4} - x_3 - 4x_4^{\ 2} - 5x_5 \geq 0$$
$$282 - 7\tilde{x}_1 - 3\tilde{x}_2 - 10x_3^{\ 2} - x_4 + x_5 \geq 0$$
$$196 - 23\tilde{x}_1 - \tilde{x}_2^{\ 2} - 6x_6^{\ 2} + 8x_7 \geq 0$$
$$-4\tilde{x}_1^{\ 2} - \tilde{x}_2^{\ 2} + 3\tilde{x}_1\tilde{x}_2 - 2x_3^{\ 2} - 5x_6 + 11x_7 \geq 0 \qquad \text{(A6)}$$

Table A6: Solution to Hock 100

Information	Nominal Solution	Robust Solution	Li et al.'s (2006) Solution
x_1	2.3304	2.2350	-
x_2	1.9514	1.8546	-
x_3	-0.4775	-0.4749	-
x_4	4.3657	4.3533	-
x_5	-0.6245	-0.6251	-
x_6	1.0381	1.0359	-
x_7	1.5942	1.5970	-
$f(x)$	680.6301	692.3847	-
Function Calls	7	19	$>10^9$

(Hock 106)

$\Delta \tilde{x} = \Delta p = 0.1$

$$
\begin{aligned}
&\min_{x}\left(\left(\tilde{x}_1\right)+\left(\tilde{x}_2\right)+(x_3)\right)\\
&s.t.\\
&\tilde{1}-0.0025(x_4+x_6)\geq 0\\
&1-0.0025(x_5+x_7-x_4)\geq 0\\
&1-0.01(x_8-x_5)\geq 0\\
&\tilde{x}_1 x_6-833.33252x_4-100\tilde{x}_1+83333.333\geq 0\\
&\tilde{x}_2 x_7-1250x_5-\tilde{x}_2 x_4+1250x_4\geq 0\\
&\tilde{x}_3 x_8-1250000-\tilde{x}_3 x_5+2500x_5\geq 0\\
&100\leq\tilde{x}_1\leq 10000, 1000\leq\tilde{x}_2\leq 10000, 1000\leq\tilde{x}_3\leq 10000, 10\leq x_i\leq 1000,\quad i=4,\ldots,8
\end{aligned}
\tag{A7}
$$

Table A7: Solution to Hock 106

Information	Nominal Solution	Robust Solution	Li et al.'s (2006) Solution
x_1	579.32	388.73	-
x_2	1359.94	1540.21	-
x_3	5110.07	5290.11	-
x_4	182.02	150.89	-
x_5	295.60	288.40	-
x_6	217.98	209.11	-
x_7	286.42	262.49	-
x_8	395.60	388.40	-
$f(x)$	7049.33	7219.06	-
Function Calls	5	17	$>10^9$

Appendix B: Discussion on Function Calls

One of the benchmarks of a useful algorithm is that it is uses less computational effort than other algorithms. One way to measure computational effort of an algorithm is a comparison of CPU time, i.e., how fast the algorithm can solve certain test problems when compared to others. However, CPU time can vary with the type of computer used, other programs running in the background, and other factors that are machine dependent.

Measuring the number of function calls is a measure of computational effort that is machine independent. Moreover, measuring computational efficiency in terms of function calls can better estimate how the algorithm will perform for different types of problems (e.g., black box or simulation-based design).

This dissertation defines function calls as any instances where the solver calls an objective function, constraint, or other value or assignment in the optimization problem. This is based on the definition of a statement execution in GAMS, which is defined as any instance where the solver calls an equation or other value or assignment in the optimization problem (GAMS, 2010). This definition was chosen in part because the modified Benders method was programmed and tested in GAMS.

This definition is also similar to the other definitions of function calls in the recent literature. The definition by (Hu et al., 2011) is that a "function call refers to one instance of calculating objective and constraint functions altogether (i.e., one call to the optimization problem)." The authors (Hu et al., 2011) have used MATLAB to solve their test problems, and their definition depends on their use of MATLAB (MATLAB, 2008). MATLAB does not have an internal explicit function call counter like GAMS, but the method of (Hu et al., 2011) entails putting the objective and constraint functions in one “m-file” and attaching a function call counter within this file. Note it is not possible to place a similar counter in GAMS because the GAMS file structure is different than MATLAB. Another definition is offered by (Li et al., 2011), who define function calls “equal to the number of points that have been evaluated during one run of the optimizer.” The authors in (Li et al., 2011) used the solver XPRESS (XPRESS, 2003) for their test problems.

Since all problems except one (Heat Exchanger Design in Section 3.5.3 was solved using MATLAB) were solved in GAMS, the following example provides a basis for comparison for function call counting in GAMS and MATLAB. This is the nominal version of Fleury’s weight minimization like problem (Section 3.5.1).

$$\begin{aligned}
&\min_x \ f(x) = \sum_{i=1}^{N} x_i \\
&s.t. \\
&\sum_{i=1}^{0.95N} \frac{1}{x_i} + \frac{1}{N^2} \sum_{i=0.95N+1}^{N} \frac{1}{x_i} - N \le 0 \\
&\sum_{i=1}^{0.95N} \frac{1}{x_i} - \frac{1}{N^2} \sum_{i=0.95N+1}^{N} \frac{1}{x_i} - 0.9N \le 0 \\
&\frac{1}{N^2} \le x_i \le N^2 \quad i = 1,2,\ldots,N
\end{aligned} \tag{B1}$$

This problem (B1) was solved using both GAMS and MATLAB. The values of N were changed to give an idea of computational effort for MATLAB and GAMS (and two different ways of measuring function calls). Table B1 shows these results.

Table B1: Comparing Function Calls Between GAMS and MATLAB

Number of Variables (*N*)	Number of Constraints	GAMS Function Calls	MATLAB Function Calls
100	102	506	5493
200	202	607	8210
300	302	913	12252
400	402	1201	17786
500	502	1403	20096

Clearly, even though we are careful in using the same definition for MATLAB and GAMS, there is a difference in counting function calls for these programs. Roughly, the function calls in GAMS are an order of magnitude or two lower for example (B1). This difference should be kept in mind when looking at the examples in this dissertation. Just to note, the function calls for the Heat Exchanger example (Section 3.5.3) were reported using the counting method for MATLAB. The results on maximum function calls presented in Table 3.1 are also applicable to both methods of counting since they talk about the maximum possible function calls, and are based on the definition of function call in this dissertation.

Bibliography

1. Balling, R., Free, J., & Parkinson, A. (1986). Consideration of Worst-Case Manufacturing Tolerances in Design Optimization. *Journal of Mechanisms, Transmissions, Automation in Design , 108*, 438-441.
2. Bard, J. (1983). An Efficient Point Algorithm for a Linear Two-Stage Optimization Problem. *Operations Research , 31* (4), 670-684.
3. Bard, J. (1988). Convex Two-Level Optimization. *Mathematical Programming , 40* (1), 15-27.
4. Bard, J., & Moore, J. (1990). A Branch and Bound Algorithm for the Bilevel Programming Problem. *SIAM Journal on Scientific and Statistical Computation , 11* (2), 281-292.
5. Bard, J., Plummer, J., & Sourie, J. (2000). A Bilevel Programming Approach to Determining Tax Credits for Biofuel Production. *European Journal of Operational Research , 120*, 30-46.
6. Barnes, T. (2010, 9 30). *House Passes Shale Gas Production Tax*. Retrieved 4 22, 2011, from Pittsburgh Post-Gazette: http://www.post-gazette.com/pg/10273/1091447-454.stm
7. Bazaraa, M., Sherali, H., & Shetty, C. (1993). *Nonlinear Programming: Theory and Algorithms (2nd Ed).* New York, NY: John Wiley & Sons.
8. Beale, E. (1975). Some Uses of Mathematical Programming Systems to Solve Problems that are Not Linear. *Operational Research Quarterly , 26*, 609–618.
9. Beale, E., & Tomlin, J. (1970). Special Facilities in a General Mathematical Programming System for Non-convex Problems using Ordered Sets of. In J.

Lawrence, *OR 69: Proc. Fifth Int. Conf. Oper. Res.* (pp. 447-454). London, U.K.: Tavistock Publications.

10. Benders, J. (1962). Partitioning Procedures for Solving Mixed-Variables Programming Problems. *Numerische Mathematik , 4*, 238-252.
11. Ben-Tal, A., & Nemirovski, A. (2002). Robust Optimization-Methodology and Applications. *Mathematical Programming, Series B , 92*, 453-480.
12. Ben-Tal, A., & Nemirovski, A. (2008). Robust Solutions to Conic Quadratic Problems and Their Applications. *Optimization and Engineering , 9*, 1-18.
13. Ben-Tal, A., El Ghaoui, L., & Nemirovski, A. (2009). *Robust Optimization.* Princeton, NJ: Princeton University Press.
14. Bertsimas, D., & Sim, M. (2006). Tractable Approximations to Robust Conic Optimization Problems. *Mathematical Programming, Series B , 107*, 5-36.
15. Bialas, W., & Karwan, M. (1982). On Two-Level Optimization. *IEEE Transactions on Automatic Control , AC-27* (1), 211-214.
16. Bjørndal, M., & Jörnsten, K. (2008). Equilibrium Prices Supported by Dual Price Functions in Markets with Non-Convexities. *European Journal of Operational Research , 190* (3), 768-789.
17. Cohen, D. (2009, 6 25). *A Shale Gas Boom?* Retrieved 4 22, 2011, from Energy Bulletin: http://www.energybulletin.net/node/49342
18. Cohon, J. (1978). *Multiobjective Programming and Planning.* New York, NY: Academic Press.

19. Conejo, A., Castillo, E., Minguez, R., & Garcia-Bertrand, R. (2006). *Decomposition Techniques in Mathematical Programming.* New York, NY: Springer.

20. Cottle, R., Pang, J., & Stone, R. (2009). *The Linear Complementarity Problem.* Philadelphia, PA: SIAM.

21. Davis, L. (1987). *Genetic Algorithms and Simulated Annealing.* Los Angeles, CA: Morgan Kaufman Publishers.

22. Economist, T. (2008, 1- 4). Liquefied Natural Gas a More Liquid Market. *The Economist* .

23. Egging, R., Holz, F., von Hirschhausen, C., & Gabriel, S. (2009). Representing GASPEC with the World Gas Model. *The Energy Journal , 30*, 97-118.

24. Ehrenmann, A. (2004). *Equilibrium Problems with Equilibrium Constraints and their Application to Electricity Markets.* Cambridge, U.K.: PhD Dissertation at Fitzwilliam College.

25. Faccinei, F., & Pang, J. (2003). *Finite-Dimensional Variational Inequalities and Complementarity Problems Volumes I and II.* New York, NY: Springer.

26. Fortuny-Amat, J., & McCarl, B. (1981). A Representation and Economic Interpretation of a Two-Level Programming Problem. *The Journal of the Operational Research Society , 32* (9), 783-792.

27. Fuller, D. (2010). Capacity Pricing in Unit Commitment. *Working Paper* , http://www.mansci.uwaterloo.ca/about/webpage.php?id=2&username=dfuller.

28. Fuller, D. (2008). Market Equilibrium Models with Continuous and Binary Variables. *Working Paper* , http://www.mansci.uwaterloo.ca/about/webpage.php?id=2&username=dfuller.

29. Gabriel, S., & Leuthold, F. (2010). Solving Discretely-Constrained MPEC Problems with Applications in Electric Power Markets. *Energy Economics , 32*, 3-14.

30. Gabriel, S., Conejo, A., Ruiz, C., & Siddiqui, S. (2011a). Solving Discretely-Constrained, Mixed Linear Complementarity Problems with Applications in Energy. In Review.

31. Gabriel, S., Garcia-Bertrand, R., Sahakij, P., & Conejo, A. (2006). A practical approach to approximate bilinear functions in mathematical programming problems by using Schur's decomposition and SOS type 2 variables. *Journal of the Operational Research Society , 57*, 995–1004.

32. Gabriel, S., Kiet, S., & Zhuang, J. (2005b). A Large-Scale Complementarity Model of the North American Natural Gas Market. *Energy Economics , 27*, 799-818.

33. Gabriel, S., Kiet, S., & Zhuang, J. (2005a). A Mixed Complementarity-Based Equilibrium Model of Natural Gas Markets. *Operations Research , 53* (5), 799-818.

34. Gabriel, S., Rosendahl, K., Egging, R., Avetisyan, H., & Siddiqui, S. (2011c). Cartelization in Gas Markets: Studying the Potential for a 'Gas OPEC'. *Energy Economics* , doi:10.1016/j.eneco.2011.05.014.

35. Gabriel, S., Shim, Y., Conejo, A., de la Torre, S., & Garcia-Bertrand, R. (2010). A Benders Decomposition Method for Discretely-Constrained Mathematical Programs with Equilibrium Constraints. *Journal of the Operational Research Society , 61*, 1-16.

36. Gabriel, S., Siddiqui, S., Conejo, A., & Ruiz, C. (2011b). Solving Discretely-Constrained Nash-Cournot Games with Applications to Power Systems. In Review.

37. Galiana, F., Motto, A., & Bouffard, F. (2003). Reconciling Social Welfare, Agent Profits, and Consumer Payments in Electricity Pools. *IEEE Transactions on Power Systems , 18*, 452-459.

38. GAMS. (2010). General Algebraic Modeling System. *GAMS Version 22.9* .

39. Ganzerli, S., & Pantelides, C. (1999). Load and Resistance Convex Models for Optimum Design. *Structural Optimization , 17*, 259-268.

40. Gibbons, R. (1996). *Game Theory for Applied Economists.* Princeton, NJ: Princeton University Press.

41. Golombek, R., & Gjelsvik, E. (1995). Effects of Liberalizing the Natural Gas Markets in Western Europe. *Energy Journal , 16* (1), 85–111.

42. Groenwold, A., & Etman, L. (2010). On the Conditional Acceptance of Iterates in SAO Algorithms Based on Convex Separable Approximations. *Structural and Multidisciplinary Optimization , 42*, 165-178.

43. Gunawan, S., & Azarm, S. (2004). Non-gradient-based Parameter Sensitivity Estimation for Single Objective Robust Design Optimization. *Journal of Mechanical Design , 126* (3), 395-402.

44. Hock, W., & Schittkowski, K. (1980). *Test Examples for Nonlinear Programming Codes.* Springer-Verlag: New York, NY.

45. Hogan, W., & Ring, B. (2003). On Minimum-Uplift Pricing for Electricity Markets. *Working Paper* , John F. Kennedy School of Government, Harvard University.

46. Hu, J., Mitchell, J., Pang, J., Bennett, K., & Kunapuli, G. (2009). On the Global Solution of Linear Programs with Linear Complementarity Constraints. *Working Paper* , http://www.rpi.edu/~mitchj/papers/LPEC_global.pdf.

47. Hu, W., Li, M., Azarm, S., & Almansoori, A. (2011). Multi-Objective Robust Optimization Under Interval Uncertainty Using Online Approximation and Constraint Cuts. *Journal of Mechanical Design , 133* (6), 061002-1-9.

48. IEA. (2008). *World Energy Outlook 2008*. Retrieved 8 24, 2011, from International Energy Agency: http://www.worldenergyoutlook.org/2008.asp

49. Karlof, J., & Wang, W. (1996). Bilevel Programming Applied to the Flow Shop Scheduling Problem. *Computers and Operations Research , 23* (5), 443-451.

50. Labbé, M., Marcotte, P., & Savard, G. (1998). A Bilevel Model of Taxation and Its Application to Optimal Highway Pricing. *Management Science , 44* (12), 1608-1622.

51. Lagaros, N., & Papadrakakis, M. (2007). Robust Seismic Design Optimization of Steel Structures. *Structural and Multidisciplinary Optimization , 33*, 457-469.

52. Lee, S., Chen, W., & Kwak, B. (2009). Robust Design with Arbitrary Distributions Using Gauss-type Quadrature Formula. *Structural and Multidisciplinary Optimization , 39*, 227-243.

53. Li, M., Azarm, S., & Boyars, A. (2006). A New Deterministic Approach using Sensitivity Region Measures for Multi-Objective and Feasibility Robust Design Optimization. *Journal of Mechanical Design , 128* (4), 874-883.

54. Li, M., Gabriel, S., Shim, Y., & Azarm, S. (2011). Interval Uncertainty-Based Robust Optimization for Convex and Non-Convex Quadratic Programs with Applications in Network Infrastructure Planning. *Networks and Spatial Economics , 11*, 159-191.

55. Lu, X., Li, H., & Chen, C. (2010). Variable Sensitivity-Based Deterministic Robust Design for Nonlinear System. *Journal of Mechanical Design , 132*, 064502-1.

56. Luo, Z., Pang, J., & Ralph, D. (1996). *Mathematical Programs with Equilibrium Constraints.* Cambridge, U.K.: Cambridge University Press.

57. Magrab, E., Azarm, S., Balachandran, B., Duncan, J., Herold, K., & Walsh, G. (2004). *An Engineer's Guide to MATLAB.* New York, NY: Prentice Hall.

58. Marcotte, P., Savard, G., & Zhu, D. (2001). A Trust Region Algorithm for Nonlinear Bilevel Programming. *Operations Research Letters , 29*, 171-179.

59. MATLAB. (2008). MATLAB and Simulink for Technical Computing. Mathworks. *MATLAB Version 2008b.*

60. Montemanni, R. (2006). A Benders Decomposition Approach for the Robust Spanning Tree Problem with Interval Data. *Discrete Optimization , 174*, 1479-1490.

61. Moore, J., & Bard, J. (1990). The Mixed Integer Linear Bilevel Programming Problem. *Operations Research , 38*, 911-921.

62. Murty, K. (1983). *Linear Programming.* New York, NY: Wiley.

63. Ng, T., Sun, Y., & Fowler, J. (2010). Semiconductor Lot Allocation Using Robust Optimization. *European Journal of Operational Research , 205*, 557-570.

64. NPC. (2007, 7 18). *National Petroleum Council.* Retrieved 4 22, 2011, from National Petroleum Council: http://www.npc.org/Study_Topic_Papers/29-TTG-Unconventional-Gas.pdf

65. O'Neill, R., Sotkiewicz, P., Hobbs, B., Rothkopf, M., & Stewart, W. (2005). Efficient Market-Clearing Prices in Markets with Nonconvexities. *European Journal of Operational Research , 164* (1), 269-285.

66. PGC. (2010). *Potential Gas Committee Reports Increase in Magnitude of U.S. Natural Gas Resource Base.* Retrieved 4 22, 2011, from Potential Gas Committee: http://www.mines.edu/Potential-Gas-Committee-reports-unprecedented-increase-in-magnitude-of-U.S.-natural-gas-resource-base

67. Qiu, Z., & Wang, X. (2010). Structural Anti-Optimization with Interval Design Parameters. *Structural and Multidisciplinary Optimization , 41*, 397-406.

68. Ragsdell, K., & Phillips, D. (1976). Optimal Design of a Class of Welded Structures Using Geometric Programming. *Journal of Engineering for Industry , 98* (3), 1021-1025.

69. Rangavajhala, S., Mullur, A., & Messac, A. (2007). The Challenge of Equality Constraints in Robust Design Optimization: Examination and New Approach. *Structural and Multidisciplinary Optimization , 34* (5), 381-401.

70. Renegar, J. (1995). Incorporating Condition Measures into the Complexity Theory of Linear Programming. *SIAM J. Optim , 5* (3), 506-524.

71. Renegar, J. (1994). Some Perturbation Theory for Linear Programming. *Mathematical Programming, Series A , 65* (1), 73-91.

72. Royden, H. (1988). *Real Analysis.* New York, NY: Prentice Hall.

73. Saito, H., & Murota, K. (2007). Benders Decomposition Approach to Robust Mixed Integer Programming. *Pacific Journal of Optimization , 3* (1), 99-112.

74. Scaparra, M., & Church, R. (2008). A Bilevel Mixed-Integer Program for Critical Infrastructure Protection Planning. *Computers and Operations Research , 35* (6), 1905-1923.

75. Siddiqui, S., & Gabriel, S. (2011b). A New Method for Solving Mathematical Programs and Equilibrium Problems with Equilibrium Constraints Using Schur's Decomposition and SOS1 Variables with a Natural Gas Market Application. *Networks and Spatial Economics* , In Review.

76. Siddiqui, S., Azarm, S., & Gabriel, S. (2011a). A Modified Benders Decomposition Method for Efficient Robust Optimization under Interval Uncertainty. *Structural and Multidisciplinary Optimization , 44* (2), 259-275.

77. Siddiqui, S., Azarm, S., & Gabriel, S. (2011d). A Modified Normal Boundary Intersection Method for Efficient Generation of Pareto Frontier. *Manuscript* .

78. Siddiqui, S., Gabriel, S., & Azarm, S. (2011c). Solving Mixed-Integer Robust Optimization Problems with Interval Uncertainty Using Benders Decomposition. *Manuscript* .

79. Skagen, O. (2010, 1 18). Global Gas Reserves and Resources: Trends, Discontinuities and Uncertainties. *SSB/NAE Workshop* . Oslo, Norway.

80. Soyster, A. (1973). Convex Programming with Set-Inclusive Constraints and Applications to Inexact Linear Programming. *Operations Research , 21*, 1154-1157.

81. Steffensen, S., & Ulbrich, M. (2010). A New Relaxation Scheme for Mathematical Programs with Equilibrium Constraints. *SIAM Journal of Optimization , 20* (5), 2504-2539.

82. Steffy, L. (2009). *Shale or Sham.* Retrieved 7 12, 2011, from Houston Chronicle: http://www.chron.com/disp/story.mpl/business/steffy/6717948.html

83. Strand, J., Miller, S., & Siddiqui, S. (2011). Infrastructure Investments under Uncertainty with the Possibility of Retrofit: Theory and Simulations. *World Bank Policy Research Working Paper No. 5516* .

84. Uderzo, A. (2010). Exact penalty functions and calmness for mathematical programming under nonlinear perturbations. *Nonlinear Analysis , 73*, 1596-1609.

85. Velarde, J., & Laguna, M. (2004). A Benders-Based Heuristic for the Robust Capacitated International Sourcing Problem. *IIE Transactions , 36*, 1125-1133.

86. Wen, U., & Huang, A. (1996). A Simple Tabu Search Method to Solve the Mixed-Integer Linear Bilevel Programming Problem. *European Journal of Operational Research , 88*, 563-571.

87. XPRESS. (2003). Getting Started. *XPRESS* . Dash Optimization Ltd.

88. Youn, B., & Xi, Z. (2009). Reliability-Based Robust Design Optimization Using the Eigenvector Dimension Reduction (EDR) Method. *Structural and Multidisciplinary Optimization , 37*, 475-492.

89. Zhu, J., & Ting, K. (2001). Performance Distribution Analysis and Robust Design. *Journal of Mechanical Design* , *123* (1), 11-17.

90. Zou, T., & Mahadevan, S. (2006). A Direct Decoupling Approach for Efficient Reliability-Based Design Optimization. *Structural and Multidisciplinary Optimization* , *31*, 190-200.